



# Dialogs

Natural Language Interpretation  
for Robotics Applications

## USER GUIDE

v.0.5

---

LAAS-CNRS

---

# Dialogs: Grounding verbal interaction for robotics

Patrick Tsemengue, Mahdi Chouayakh, Séverin Lemaignan, Raquel Ros Espinoza

March 29, 2011



# Contents

<b>1</b>	<b>Module overview</b>	<b>5</b>
<b>2</b>	<b>Data Structure</b>	<b>6</b>
2.1	Description . . . . .	6
2.2	Examples . . . . .	7
<b>3</b>	<b>Natural Language Parsing</b>	<b>9</b>
3.1	Pre-processing . . . . .	9
3.1.1	Punctuation . . . . .	9
3.1.2	Changing Words . . . . .	9
3.1.3	Beginning Sentence . . . . .	10
3.1.4	Dualities . . . . .	11
3.1.5	"And" processing . . . . .	12
3.1.6	Merging . . . . .	13
3.1.7	Other processing . . . . .	14
3.1.8	Commas and subsentences . . . . .	14
3.1.9	Limits of the pre-processing that can be corrected . . . . .	15
3.2	Main Process . . . . .	15
3.2.1	Tenses . . . . .	15
3.2.2	Sentence type . . . . .	16
3.2.3	Subsentence . . . . .	17
3.2.4	Sentence analysis . . . . .	17
3.2.5	Yes or no question and other sentences . . . . .	18
3.3	Main Rules . . . . .	22
<b>4</b>	<b>Reference grounding</b>	<b>23</b>
4.1	Resolution . . . . .	23
4.1.1	Algorithms . . . . .	24
4.1.2	Adjectives ONLY approach . . . . .	26
4.1.3	Quantifier approach . . . . .	26
4.1.4	Demonstratives determiners approach . . . . .	27
4.1.5	Occurrence of the words 'OTHER' . . . . .	27
4.1.6	Handling Unidentified Anaphora error . . . . .	28
4.1.7	Handling Insufficient input error . . . . .	28
4.2	Anaphora Matching . . . . .	28
4.3	Discrimination . . . . .	30
4.3.1	Merge . . . . .	30
<b>5</b>	<b>Outputs of the Dialogs module</b>	<b>32</b>
5.1	Statement builder . . . . .	32
5.1.1	Simple sentences . . . . .	32
5.1.2	Relative sentence processing . . . . .	32
5.1.3	Verb processing . . . . .	33
5.1.4	Negative sentence approach . . . . .	34
5.1.5	verb tense approach . . . . .	34

5.1.6	Adverb modifying the verb - processing approach . . . . .	35
5.1.7	Adverb modifying the sentence - processing approach . . . . .	35
5.2	Question handler . . . . .	35
5.2.1	Wh-question . . . . .	35
5.2.2	Yes-No-Question . . . . .	36
5.3	Verbalization . . . . .	37
5.3.1	Sentence type of verbalization . . . . .	37
5.3.2	The transformation which are always done . . . . .	37
<b>6</b>	<b>Examples</b>	<b>38</b>
6.1	Debug output of a complete execution . . . . .	38
<b>7</b>	<b>User studies</b>	<b>40</b>
7.1	Relation between speakers . . . . .	40
7.1.1	First dialog with Mahdi . . . . .	40
7.1.2	Second dialog with Patrick . . . . .	40
7.1.3	Third dialog with Mahdi . . . . .	41
7.2	Spy Game . . . . .	42
7.3	Cleaning the table . . . . .	45
7.3.1	First user . . . . .	45
7.3.2	Second user . . . . .	46
7.3.3	Third user . . . . .	47
7.4	Context . . . . .	49
7.5	Other . . . . .	52

# Chapter 1

## Module overview

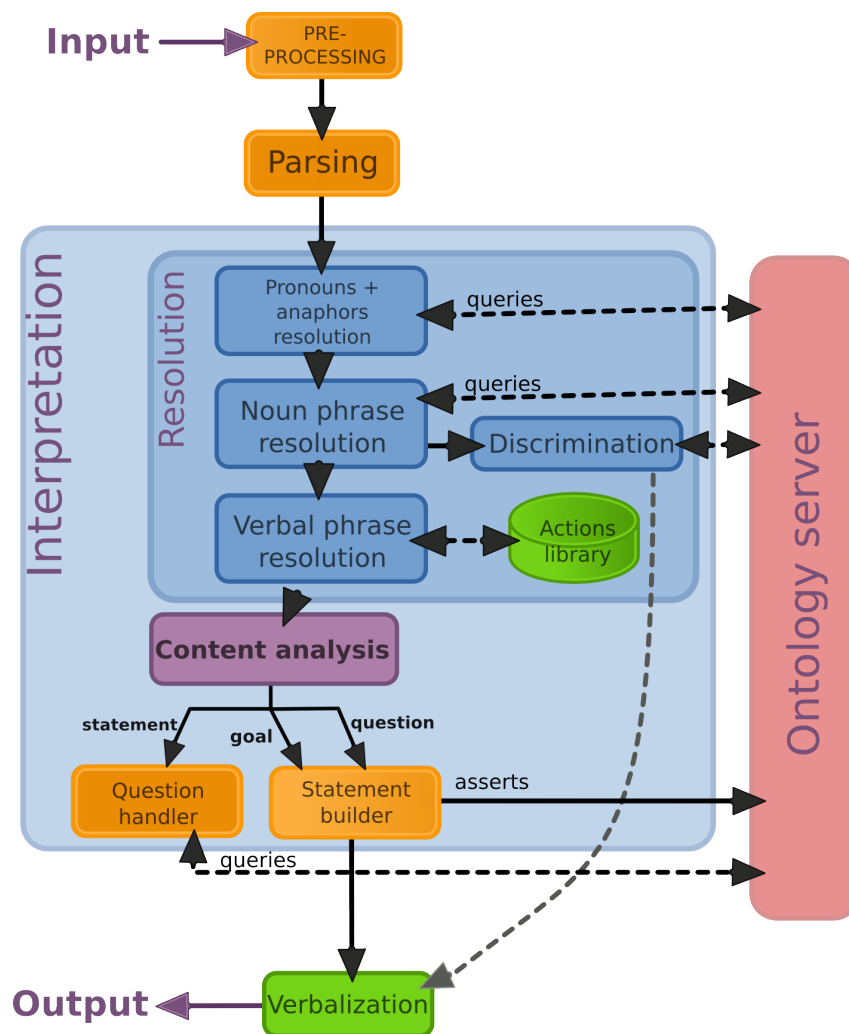


Figure 1.1: Dialogs overview

## Chapter 2

# Data Structure

### 2.1 Description

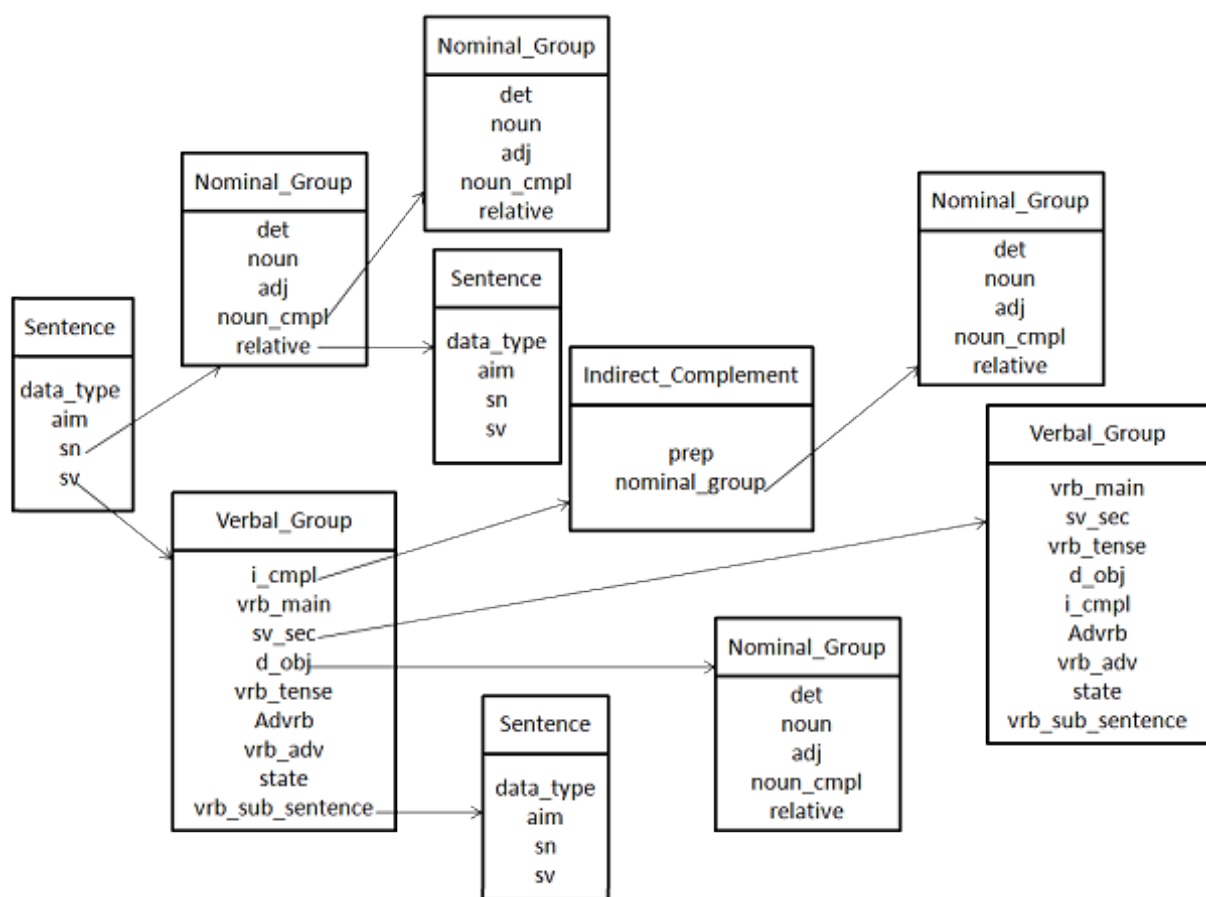


Figure 2.1: Data structures in Dialogs

The *Sentence data structure* consists of a tree of Nominal and verbal structures and *Sentences*. The use of a tree eases the information fetching. Inner components of a *Sentence* are arranged into list of objects. Therefore, the data that is transmitted through the Dialogs modules will be a list of several *Sentences*.

- A *Sentence* is made of:
  - **sn**: a nominal structure typed into a list of *Nominal\_group*

- sv: a verbal structure typed into a list of Verbal\_Group
- aim: used for retrieving the purpose of a question (Mandatory when processing a w\_question or a subsentence)
- data\_type: is the type of the sentence (w\_question, statement...)

- Nominal group class declaration

- “det”: determiner
- noun: a simple noun
- adj: a list of adjectives describing the noun
- noun\_cmpl: the noun complements attribute, which is a list of Nominal\_group
- relative: The relative subordinate clause, which is of a list of Sentences type

Additional Nominal Group attribute are:

- “id”: Holding the ontology identifier of its container nominal group set by default to none.
- \_resolved: This hold the value TRUE when the Nominal group has been resolved (Cf. Resolution)
- \_conjunction = 'AND' #could be 'AND' or 'OR' ...
- \_quantifier = 'ONE' #could be 'ONE' or 'SOME' ... (Cf. Quantifier)
- An indirect complement or adverbial is made of:
  - Indirect complement class declaration
  - gn: nominal group
  - prep: preposition
- Verbal group class declaration
  - vrb\_main: the main verb of a sentence
  - vrb\_sec: an accompanying verb of the main verb
  - vrb\_tense: the main verb tense
  - d\_obj: the direct object referred by the main verb
  - i\_cmpl: the indirect object referred by the main verb or an adverbial formed from a nominal group
  - vrb\_adv: an adverb describing the verb
  - advrb: an adverb used as an adverbial of the whole sentence
  - “comparator”: the nominal group, object of the comparison. Used for comparing nominal groups grammatically (E.g.: Bigger than)

## 2.2 Examples

Input:

The man, who talks now, has a new car

Parsing result:

```
Sentence(Sentence.statement, '',
  [Nominal_Group(['the'], ['man'], [], [], [Sentence(Sentence.relative, 'who',
    [],
    [Verbal_Group(['talk'], [], 'present simple',
      [],
      [],
      [], ['now'], Verbal_Group.affirmative, []))]]]),
  [Verbal_Group(['have'], [], 'present simple',
    [Nominal_Group(['a'], ['car'], [['new'], []], [], [])],
    [],
    [], [], Verbal_Group.affirmative, [])])
```





## Chapter 3

# Natural Language Parsing

We decided to store different data, with different functionalities in the sentence, in the same variable. The goal is to limit the data structure attribute number. For example, we have decided to create a variable that contains the direct object of the verb and another one contains indirect complements of the verbs - indirect objects or adverbials that are noun phrases as both follow preposition. Before starting the Parsing of the utterance, we attempt to give it a formalized form that eases the parsing. The following describe functions involved in the formalization process.

### 3.1 Pre-processing

#### 3.1.1 Punctuation

The end of the sentence phrase in an utterance is detected by the use of punctuation. At the end of the utterance, when the punctuation is missing, a full point is automatically added. Points can be attached or separated from the word that comes before. All commas occurring at the end of the pre-processing are deleted as they are of no use for the rest of the process. Duplicated punctuation is truncated.

Example: "the bottle is blue???" ? "the bottle is blue?"

#### 3.1.2 Changing Words

- Concatenate prepositions

Some prepositions are composed of several words; it is putting them together with "+" instead of space.

Utterance: "The bottle is next to the table in front of the kitchen."  
? \The bottle is next+to the table in+front+of the kitchen."

- Extension of contractions

This function makes some changes. It eliminates a word to be replaced by other one or modify it and add some others.

Utterance: "I wanna play with my guitar. I'd like to go to the cinema."  
?\I want to play with my guitar. I would like to go to the cinema."

- Is

In pre-processing, all "'s" linked to a personal pronoun or a few prepositions as "what" and "that" may not represent a relationship of possession but the verb state "is". So after this condition, change will be made or not.

Utterance: "It's on the table."  
?\It is on the table."

- Possession

In this module, we are processing "'s". If we have this form, certainly there is a relationship of possession that can be translated by the use of "of" ("s" of the verb "be" was processed before). We therefore must find the noun before and after "'s". Knowing that the noun after is not preceded by a determinant, it is necessary to add it. We have consequently chosen the determinant "the" because the object is known.

Utterance: "Jido's blue bottle is on the table".  
 ?\The blue bottle of Jido is on the table."

This processing can be done in cascade which allows us to have two or more ownership. We use for that the list of nominal groups which is in LIFO.

Utterance: "You shouldn't drive his poorest uncle's wife's big new car."  
 ?\You shouldn't drive the big new car of the wife of his poorest uncle."

If we have a plural noun, the "'s" is not maintained, in its place we have "s". Example: the boys' ball is blue. After this process, we look into the sentence to add the missing determinant after "of".

- Numbers

The numbers are determinants. For compound numbers, we created a function that determines them and concatenates them with "+". We have also to delete "and" between them.

Utterance: "I take twenty two bottles."  
 ?\I take twenty+two bottles."

- I/i

All "i" will be changed on "I"

- Month/day/pm/am

Months and days must be with capital letter, so we make change if it is indeed. For "pm" and "am", normally, they are linked to the digit before, so we separate them.

### 3.1.3 Beginning Sentence

- Upper case process

We can start sentences with capital letters or not. But if the word starts with a capital letter, we process it as a nominal group (proper name). So when it is an imperative sentence, the verb must be known before or it must be written with a lower case. Algorithmically, we know that is not an upper case, if we can determine a noun, a number, a specific start word phrase or a verb (known in advance).

```

1  --The algorithm:--
2      if capital_letter(sentence[0]) == 1:
3          lowercase(sentence[0])
4          if nominal_group(sentence, 0) != []:
5              return sentence
6          elif sentence[0] in beginning_sentence_list:
7              return sentence
8          elif sentence[0] in verb_list:
9              return sentence
10         else:
11             uppercase(sentence[0])
12         return sentence

```

Some processes require the presence of the lower case even if it is a proper name. These processes are during the capital letter process.

- Changing the beginning of sentence

Alternative process should be done with the process of capital, is changing the placement of some elements found at the beginning of a sentence. For that, we determine the nature of the first element. If it is an adverb or adverbial, we move it at the end of the sentence and we put, just before it, a comma (if the sentence ends with a subsentence or a relative, we must separate this element from the secondary sentence).

Utterance: "And now, can you reach the tape."  
 ?\Can you reach the tape; now."  
 Utterance: "And now, can you reach the tape which is blue."  
 ?\Can you reach the tape which is blue; now."

If it is an indirect complement or second verb, we try to find comma after it. If there is a comma, we move all the part before it. Else we move just the indirect complement.

Utterance: "and in a dialog there is an interaction between them."  
 ?\there is an interaction between them; in a dialog."  
 Utterance: "to have a dialog, we need more than 1 protagonist."  
 ?\we need more than 1 protagonist; to have a dialog."

- Interjection

We can have interjection with exclamation mark - "!" - but also with comma - "," - if it is linked to another sentence. So we have to know the position of the first comma - ",". If before it there is a preposition, the comma is for this secondary sentence. Otherwise, we know if it is an interjection with the first word. Finally, we have to find the nature of the element which follows it. If it is a nominal group, we have a list of nominal group which are linked with "and". After recovering all these nominal groups, if we still have a comma, it is necessary an interjection. Else we continue the other processes.

Example: "He Patrick, the bottle is on the table. Give it to me"  
 Example: "Jido, give me the bottle."  
 Example: "Jido, Patrick and you, give me the bottle"  
 Example: "Jido, Patrick and you will go to the cinema." ? we have not interjection

```

1  __The algorithm:__
2  for i in sentence
3      if i==',' and no_subsentence(sentence[:sentence.index(i)])
4          if start_with_interjection_word(sentence)
5              sentence[sentence.index(i)]=','
6              return [sentence[:sentence.index(i)+1], sentence[sentence.index(i)+1:]]
7          #We find the nominal structure from the beginning
8          nom_struc=find_nom_structure(sentence)
9          #After the num_struc we must have a verb else it is an interjection
10         if sentence[lent(nom_struc)]=='!'
11             sentence[lent(nom_struc)]=','
12             return [sentence[:lent(nom_struc)+1], sentence[lent(nom_struc)+1:]]

```

### 3.1.4 Dualities

- Duality between the adverbial subsentence and adverbial nominal group

There are a few preposition of subsentence which can be as both adjective or preposition of nominal group. During this process, we determine this difference. Indeed in a main sentence if we have a preposition and a nominal group that follows it, we determine the nature of the word that comes after the noun. In the case where it is an adverb which is related to the sentence (such as *tomorrow*) or another nominal group or another preposition... , this following element is another adverbial of the same sentence.

Utterance: "I do my homework before 10 minutes from now."  
 \I do my homework before 10 minutes from now."

Otherwise we have a verb or an adverb which preceded the verb (as *quickly*) it is a subsentence. In order of the following processes make the differentiation, we added ":" for preposition to mention that is a preposition of adverbial subsentence.

Utterance: "I do my homework before he arrives."  
 \I do my homework: before he arrives."

- Duality between "but" of nominal groups and "but" of subsentence

Usually, we could not differentiate "but" between two nominal groups or "but" that represents a preposition of an adverbial. So the user has to put ":" before "but" to process it as conjunction between two nominal groups. However we can limit the cases. Normally, the user must write:

Example1: \The bottle is not blue but it is red."  
Example2: "It is not the glass :but the bottle."  
Example3: "It is not a blue :but red glass."  
Example4: "give me the bottle now but you can keep the glass."

But when we find "but", if just after we have not a nominal group it must be the conjunction (a subsentence has a subject).

Example3: "It is not a blue but red glass."

Else, we will find the nominal group just before. If its position is not immediately before the preposition (as nominal\_group1 + but + nominal\_group2), it must be preposition of the subsentence.

Example4: "give me the bottle now but you can keep the glass."

In the other case (nominal\_group1 + but + nominal\_group2), we cannot differentiate between the two "but" (  
)

Example1: \The bottle is not blue but it is red."  
Example2: "It is not the glass but the bottle." ? "It is not the glass but the bottle."

### 3.1.5 "And" processing

- "and" between adjectives

In lists of adjectives, commas and "and" conjunctions are removed.

Utterance: "The bottle is blue, big and fanny."  
?\The bottle is blue big fanny."

- Adding "and" between nominal groups

There are two cases. During these processes, we stop them when a condition is not satisfied in a particular order.

The first case, we determine the presence of a comma between two nominal groups. With using a list, we store them and we store the other nominal groups which are also separated by comma. This process is until we find the conjunction "and". If not we stop the progression in this function.

Utterance: "give me the bottle, the glass, the pen and the paper."  
? Nominal group list separated by comma is [the bottle, the glass, the pen]

Normally, the following word is the conjunction "and". If not we stop the process, else we add the following nominal group in the list

Utterance: "give me the bottle, the glass, the pen and the paper."  
? Nominal group list separated by comma is [the bottle, the glass, the pen, the paper]

We make change with the list.

Utterance: "give me the bottle, the glass, the pen and the paper."  
? "give me the bottle and the glass and the pen and the paper."

The second case, if there is no comma between nominal groups and we find two which follow, we store them and we store the other nominal groups following in a list.

Utterance: "I give you the bottle the glass the pen and the paper."  
? Nominal group list separated by comma is [you, the bottle, the glass, the pen]

Normally, the following word is the conjunction "and". If not we stop the process, else we add the following nominal group in the list

Utterance: "I give you the bottle the glass the pen and the paper."

? Nominal group list separated by comma is [you, the bottle, the glass, the pen, the paper]

After, we need to verify an important property: All nominal groups should be similar. There are two forms: we can have a personal pronoun or a det+adj+noun. Example: [you, me] or [the man, the bottle]

It is possible to have a mixture of both forms, but this follows a rule equally important: We always have the complete form and when a personal pronoun form is detected all nouns following must be a personal pronoun (we cannot have det+adj+noun then personal pronoun then det+adj+noun). Example:[you, me, the man] is not possible ? [the man, you, me] is possible.

? Nominal group list separated by comma is [you, the bottle, the glass, the pen, the paper]

? the rule is not respected

If this rule is not respected, the program resumes the position of the second noun in the list, in our sentence "the bottle"

Utterance: "I give you the bottle the glass the pen and the paper."

? Nominal group list separated by comma is [the bottle, the glass, the pen, the paper]

? "give me the bottle and the glass and the pen and the paper."

- "and" between sentences

We can have many sentences separate by "and". The user must add comma, just before the conjunction. With "; and", we cut the sentence into two different sentences.

### 3.1.6 Merging

- Conjunction

This variable tells us the conjunction that precedes the noun. By default we have chosen to put "AND". So for the first nominal group, this variable has no effect (if it is "AND"). This variable can be also: "OR" and "BUT". The conjunction is linked to the nominal group which follow it When the conjunction is for adjectives or determinants, we have to duplicate the nominal group to extend the conjunction of the entire nouns.

Utterance1: "You'd like the blue bottle or the glass?"

"You'd like the blue bottle or the glass?"

Utterance2: "The green or blue bottle is on the table."

\The green bottle or the blue bottle is on the table."

Utterance3: "Don't give me two but three bottles."

\Don't give me two bottles but three bottles."

Therefore, we can have several nominal groups with different conjunctions. Example: I'll play a guitar or a piano and a violin.

- What+to

Utterance1: "tell me what to do" ? "tell me the thing that is to do."

- Merge sentence

Some time, user forgets the verb "be". In some case, we add a relative form to correct the mistake.

Utterance: "the bottle on the table"

?\the bottle which is on the table."

Some time, user has to put the comma because after pre-processing we can have a relative followed by verbal structure. So semicolon is compulsory.

Utterance: "the bottle on the table, is blue"  
?"the bottle which is on the table; is blue."

- Infinitive verbs

Some verbs need infinitive form without using "to". In this function we add the preposition to make the following process easier. That creates a new verbal structure.

Example of verb: "let"  
Let me take... ? Let me to take...

### 3.1.7 Other processing

- Other processing

In this function, we will make different processes. Starting with question with "which", this w\_question are about a nominal group which doesn't have a determinant. Because of the necessity to have a determinant to recover it, we add "the".

Utterance: "which bottle do you mean?"  
\which the bottle do you mean?"

For sentence with verb 'think' without preposition, there is normally a subsentence which follows it, that is why we add "that".

Utterance: "I think it is cool"  
\I think that it is cool."

Regarding the interpretation module, 'in front of' and 'in the front of' converge to the same result. We chose to make the transformation to that level to follow a single model.

Utterance: "it is in front of the man."  
\it is in the front of the man."

- Move some prepositions

Some prepositions follow the noun, it is compulsory to exchange the positions.

Utterance: "I played a guitar a year ago."  
\I played a guitar ago a year."

- 2 Determinants

When we have two determinants with the same nominal group, we delete one of them

Example: \all the bananas are here" ? we delete "all"

### 3.1.8 Commas and subsentences

An adverbial is a sentence; it should be treated as sentences. The problem is to find its starting point and ending point. For starting, every adverbial have to start with a proposition. Concerning the end, we used a semicolon (Cf Subsentence). Therefore, all commas before this preposition must be deleted. The comma at the end is changed to semicolon if it is in the middle of the sentence. But if it is the end, the final point is enough.

### 3.1.9 Limits of the pre-processing that can be corrected

v In some functions like the one where we differentiate preposition of subsentence and preposition of indirect complement, we recover the nominal group with his noun complement. We can have with it another nominal group which is related to it by a conjunction. This case is not processed.

Utterance: "I do my homework before the man and he arrive."

After recovering "*the man*" we have "*and*" which is not an adverb or a nominal group or preposition. So it is a subsentence ? it is correct

Utterance: "he did his homework before you and me."

Same problem, but it is wrong because we have not a subsentence but a double indirect complements

v Some time, when we put "*and*" between sentences, we delete the subjects if it is the same with the one of the first sentence. So we must duplicate it.

Utterance: "I give you the pen and put the bottle on your left"

The second sentence will be considered as an imperative. So we have to add "*I*" to correct the mistake

? "I give you the pen and put the bottle on your left"

v We might observe some other limits that nonetheless, are not discussed in this report.

## 3.2 Main Process

This module calls a chain of external functions to apply the rules of grammar to the sentence. Once the information is stored, we will delete it from the sentence. For example, the determination of nominal group is through the determinant, the personal pronoun or uppercase.

I play guitar ? here; there is a mistake because 'guitar' is not preceded by a determinant  
I play a guitar ? OK

When we find the nominal group, we find also his adjectives, his noun complement and his relative. After the process, we delete all this information from the sentence.

### 3.2.1 Tenses

For tenses, we can process:

1. Present simple ? For singular "s"
2. Present perfect ? "has" or "have" + past participle
3. Present progressive ? "is" or "are" or "'m" + verb + "ing"
4. Present passive ? "is" or "are" or "'m" + past participle
5. Present conditional ? Using modal in conditional or "would"
6. Past simple ? Verb in the past
7. Past perfect ? "had" + past participle
8. Past progressive ? "was" or "were" + verb + "ing"
9. Past passive ? "was" pr "were" + past participle
10. Past conditional ? Present conditional with the present perfect
11. Future simple ? Using "will"
12. Passive conditional ? If we have "be" +past participle preceded by conditional



### 3.2.2 Sentence type

Once the pre-processing ends the first processing, sentences undergo according to the type of different modules that depend on the shape of the sentence another process. This classification is done using a list that contains the words with sentence can usually start.

```
Example1: "is the bottle of my brother in your right?"
? Using auxiliary, it is an yes or no question
Example2: "good afternoon"
? Using good or hello/hi, it is start dialog
Example3: "no."/"Sorry."
? It is disagreeing
Example3: "OK"
? It is agreeing
```

If we have an interrogative word, the sentence is a w\_question. The classification will therefore be the subject of the question.

Examples:

1. When must you take the bus ? date
2. Where is Broyen ? place
3. Where must Jido and you be from ? origin
4. Why should she go to Toulouse ? reason
5. Who could you talk to on the phone ? people
6. Whose blue bottle and red glass are these ? owner
7. To whom are you talking ? people

Which salesperson's competition won the award which we won in the last years ? choice

For questions constructed with "what", they can be processed differently. Indeed, for the issues that concern objects or conditions reported directly after "what" returns aims as examples:

1. What time is the news on TV? ? time
2. What size do you wear? ? size
3. ...

Otherwise we proceed as for other w\_questions. However, we have to meet certain conditions:

1. If verb = "happen" ? situation
2. If verb = "like" + tense is not conditional ? description
3. If verb = "go" + first i\_compl ends with "ing" ? explication
4. If verb = "think" ? opinion
5. Else for all other cases it is ? thing
6. It the case when we have "what kind" or "what type" ? classification + object of the question. The processing of the aim is not as for other w\_questions.

Utterance: "What type of people doesn't read this magazine? What kind of music must he listen to everyday? What kind of sport is your favourite?"

?We will have aim: classification+people, classification+music and classification+sport. The analysis is applied on \don't read this magazine?", \must he listen to everyday?" and \is your favourite?"

For questions constructed with "how", they can have a different processing. Indeed, concerning objects or conditions reported directly after "how", we have a similar processing with "what questions".

Examples:

1. How old are you? old
2. How long is your uncle's store opened tonight? long
3. How far is it from the hotel to the restaurant? far
4. How soon can you be here? soon
5. How often does Jido go skiing? often
6. How much or How many? quantity
7. How about going? invitation
8. ...

Otherwise we proceed as for other w\_questions. However, we have to meet certain conditions:

1. If verb = "like" ? opinion
2. Else for all other cases it is ? manner

### 3.2.3 Subsentence

An adverbial is a sentence; it should be treated as sentences. The problem is to find its starting point and end point. Normally a user must define the adverbial by 2 commas. The first which is normally before the preposition will be removed (so the user can omit it). The second is at the end, is not essential if it is the end of the sentence. The program will automatically add the semicolon that marks the end of the subsentence at the end of the sentence or in place of the comma.

Utterance: "the man, who talks, has a new car. I play the guitar that I bought yesterday."  
?\the man who talks; has a new car. I play the guitar that I bought yesterday;."

In the case when we have many adverbials in the sentence, it is necessary to put as many commas as there are bounding adverbials.

Utterance: "The bottle that I bought from the store which is in the shopping centre, , is yours."  
?\The bottle that I bought from the store which is in the shopping centre; ; is yours."  
Utterance: "I give the bottle that I bought from the store which is in the shopping centre"  
?\I give the bottle that I bought from the store which is in the shopping centre;;."

At the end, we can analyze sentences such as:

Utterance: "don't quickly give me the bottle which is on the table, and the glass which I cleaned yesterday, at my left."

### 3.2.4 Sentence analysis

If we have some data type as "disagree" or "start", we can have in the same sentence other phrases that must be processed. So if, after the first phrase, it is not the end of the sentence and we detect ";", we continue the processing of the second part. Else we consider that the main information is the recovered data type and we stop the analysis. Utterance: "Sorry, can you give me the bottle." ? We process the second part Utterance: "Sorry, Jido." ? We don't process second part When we have an "!", it can be interjection, exclamation or imperative. The exclamation is with an interrogative word. If the sentence starts with an interrogative word which can be a preposition of subsentence, we have to see the following element. If it is a nominal group, it is a subsentence because w\_question need to have a subject inversion. Utterance: "when did he come?" Utterance: "when he comes, I will give him the bottle." The processing of all sentences is based on analysis of yes or no questions and of other sentences, so we have to discuss it.

### 3.2.5 Yes or no question and other sentences

There are no many differences between the two functions, so we will explain them at the same time.

- Subject and nominal groups

The determination of the subject is the same; we just have to take account of the inversion of subject in questions. When we recover a nominal group with its properties, we delete it.

- Plural

When there is plural, usually there is no determinant, so we cannot recover the nominal group. That is why we have to add "a" before it. In the case of an indirect complement, there is a preposition, so we need to see the plural after it.

- Adjective quantifiers

In a sentence, there are adjective quantifiers. That is why we changed the adjective list. The new form is ['the adjective', ['list', 'of', 'quantifier']]

Example: the very and too big man ? ['big', ['very', 'too']]

- Relative

The relative removes the redundancy through the preposition. If it replaces the subject, there is no problem. Otherwise it is necessary to recreate the redundancy to keep track of information sent.

Utterance: "I take that bottle that I drink in."  
?\I take that bottle that I drink in that bottle."

- Determinants and quantification

Normally a noun is detected by its determinant. If we have a plural at the beginning of a sentence or after a verb, we can detect it and add the determinant "a". After that, we need to find the quantifier which corresponds to it. This part of process is to make the following processes easier. These are the rules of quantifications:

1. a/an ? SOME
2. a/an+plural ? ALL (we delete the determinant because we add it in process plural)
3. the+pluriel ? ALL
4. vide+pluriel ? ALL
5. no ? NONE
6. no+pluriel ? ANY
7. any ? ANY
8. those/these ? SOME
9. there+are ? SOME
10. every ? ALL
11. anything ? ALL
12. something ? SOME
13. some ? SOME
14. else ? ONE

- Digit

We have to convert numbers to digits. For that, we perform these rules:

1. End with 'teen' == +10
2. End with 'ty' == \*10
3. Hundred == \*100
4. Thousand == \*1000
5. Million == \*1000000

- Modal

To find it, we used a list of all the modal forms (simple or conditional). The processing (for tense or other) will change if there is a modal in the sentence. The modal is used in its current form in the sentence with the main verb in vrb\_main.

Example: "You shouldn't drive his poorest uncle's wife's big new car. Should I give you the bottle? Shall I go?"

?We will have vrb\_main=['should+drive'], vrb\_main=['should+give'] and vrb\_main=['shall+go'].

- Adverb liked to the verb

We may process a particular case. In fact, the adverb may be in the end of sentences. It is therefore imperative to recover it.

Utterance: "Take the bottle carefully."

?vrb\_adv=['carefully']

- Negation

It takes a position in the sentence, so we have to consider it. Usually, it follows an auxiliary "do". After the determination of the verb and its properties, we delete it to start the process of the end of the sentence.

- Duality between conjunctive subsentence and the nominal group

"that" can be used as a determinant for a noun and as a preposition for an adverbial. The differentiation was made; because "that" of the subsentence is necessarily followed by a nominal group (the subject) and the preposition in main sentence must be preceded by the verb or a personal pronoun as "me"

Utterance: "Learn that I want you to give me the blue bottle." ? After "that" we have nominal group

Utterance: "Give me that bottle." ? After "that" we have no nominal group ("bottle" is just a noun)

Utterance: "I tell you that I want you to give me the blue bottle that is blue." ? it is a subsentence

- Infinitive verb

During the process, it is possible to have several verbs present in the infinitive form. It is also possible that they belong to different subsentences. Because we determine it through "to", we added ":" in the case when "to" must not be processed at this stage of the analysis.

Example: "tell me the thing that is to do."

In this example "to do" must be processed with the relative and not with the main sentence

Example: "I want to tell him the thing that is to do"

At the beginning of the process: "I want :to tell him the thing that is :to do". Basically the first "to" is in the main sentence: "I want to tell him the thing that is :to do". When we start the process of the second verb and we find the relative, we take off ":" to have "the thing that is to do."

- Duality between adverbial and relative

In the case where we have subsentences with prepositions "which" and "where", we must differentiate the relative from the adverbial. For that, we must discover the nature of the item that comes just before. If it is a nominal group in the complete form (not pronoun but det+adj+noun), it is a relative which define this nominal group. So we escape and continue the process. But if it is not a nominal group, we perform the adverbial process.

- Duality between indirect complement and the noun complement

Some adverbials represent an additional noun complement for the nominal group. This depends on the verb and the place occupied in the sentence.

Example: "give me the bottle on the table."

The bottle is on the table and not the action is on the table so we must have "give me the bottle which is on the table."

- Compare

This form is visible through the presence of "than". We have not had time to deepen the reasoning on this grammatical form.

- Duality between the direct and the indirect complement

In processing, we differentiate a direct object with the absence of preposition. Otherwise, the nominal group will be inserted in the i\_cmpl.

Example: I give it to you  
\it" direct object  
\to you" indirect object

It is possible to have two groups with no preposition and are not bound by "and". In this case, the first noun that is d\_obj will be moved in i\_cmpl and the second one takes its place. In fact in every sentence there a single direct object

Example: give me the bottle  
\the bottle" direct object  
\me" indirect object

Some verbs don't need a direct complement. So we have to move contain of d\_obj and put it in i\_cmpl.

Example: \tell me the thing that you do"  
\the thing that you do" is not an agent of the verb, so it must be in i\_cmpl.

- Agent complement

In a passive sentence form, we have an inversion between the subject and direct object. We have an additional officer who is preceded by "by", we shall be considered as an adverbial and so we will put it at the i\_cmpl.

Utterance: "the code is written by me."

- State verb

In the case of a state verb, we note the absence of a direct object as opposed to adverbial. We have therefore chosen to use the variable d\_obj to convey information related to adjectives.

Utterance: "The bottle is on the table. The bottle is blue. The bottle is Blue."  
 ? For the first sentence, it is an adverbial  
 ? In the second, we have an adjective. We create, on the level of d\_obj, a Nominal\_Group contained that adjective  
 ? In the third sentence, we have a proper name (any word in uppercase is considered a proper name) and so we use the d\_obj

- End of the sentence

During the process, we delete the word that we determinate the nature. At the end, if we still have some words, we try to create with them other nominal groups. There is some words that we need to delete during this process

Utterance: \I play guitar"  
 At the end we have : \guitar" / We create a nominal group with it / We find a direct complement.

- Change subsentence to relative

For some adverbials, the interpretation can be easier if we change them into relative.

Example: adverbial with \what" ? we will have relative with \the thing" as nominal group  
 Example: adverbial with \where" ? we will have relative with \the location" as nominal group

- Imperative

If the sentence doesn't start with a nominal group, it is an imperative one. So for plural it is compulsory to add the determinant to find the nominal group. Also when the sentence starts with an adjective, we add "the".

Example for plurals: \bananas are fruits."  
 Example for adjective: \blue bottle is beautiful."

But if the sentence starts directly with a noun (not plural), the program cannot parse it correctly.

Example which parsing is impossible: \bottle is an object."

- Forcing question

A sentence may appear in a declarative but ends with a question mark. In this case, we force the query and retrieve a yes or no question instead of statement.

Utterance: "Mahdi is going to the Laas?"

- Demonstrative pronoun

In English, we have some demonstrative pronouns; they can be used as pronoun or as determinants. We have therefore reduced their scope of use. Indeed, a demonstrative pronoun cannot be followed only by the verb of state "be".

Example1: this is a bottle  
 Example2: there is a bottle on the table  
 Example3: this bottle is blue  
 Example4: this goes to the cinema  
 ? the fourth example generates a mistake

For a question, it requires a subject inversion. To find if it is a demonstrative pronoun and not a determinant, we need to know if the item that follows is neither a verb in the present progressive or passive present, nor a word related to the verb.

### 3.3 Main Rules

- Capital letters

If the sentence is an imperative one, we have to put the verb with a lower case. Verb with upper case is allowed if we know it (in the list of thematic roles). All proper name must be with capital letter otherwise we don't parse it as a nominal group.

- Punctuation

If there are many sentences in the utterance, the user must put punctuations to separate them. For subsentence, the user must put a comma at its end.

- Adjective/noun

During the parsing, we don't use a dictionary. So every word has just one function. If the word is an adjective, it cannot be noun. For example wine is a noun (the wine), but we can use it as an adjective (the wine glass). So during the parsing, wine is all time a noun and "the wine glass" cannot be parsed.

- Determiner

Nominal groups are found with the determinant. In some case, the parser adds a determinant automatically (Cf Main process)

## Chapter 4

# Reference grounding

### 4.1 Resolution

The purpose of this module is to produce a resolved sentence that is to accurately identify each element involved in a sentence, by affecting them an existing and unique reference in the ontology. Let's suppose we are processing the natural language input *the red cube is on the blue table*. Before committing this information in the ontology, we first try to uniquely identify *the red cube* (1) and *the blue table* (2). To do so, we build a set of matching RDF<sup>1</sup>/OWL statements in order to query the ontology. Assuming that there is only one *red cube* in the ontology, we should retrieve its unique identifier, which is what we are looking for.

Let's consider the example below, ?concept is the identifier to retrieve and to affect to the nominal group being processed. However, if there is more than one red cube, then we need to retrieve the accurate identifier by the process of discrimination (Cf. Discrimination).

E.g.: "the red cube" Generated statements:

```
[ ?concept rdf:type Cube, ?concept hasColor red]
```

The resolution of reference to the speaker, recipient or anaphora is different. We look through the nominal group elements and if we find personal pronouns such as "I" or "me", the nominal group is affected with the current speaker identifier. If we find a personal pronoun such as "you", the nominal group is identified with the recipient identifier; finally when anaphora such as 'it' or 'one' occur, we attempt to retrieve the matching object. (Cf. Anaphora resolution)

Action verbs are also resolved, however differently from the nominal group resolution as, we do not build a set of RDF statements to query the ontology, but instead, we look up their matching synonyms (thematic roles) in the shared files in order to retrieve their reference in the ontology. In the sentence "*I take the bottle*", the action verb "*take*" is referenced by "*Get*". (Cf: thematic\_roles)

---

<sup>1</sup>Resource Description Framework



### 4.1.1 Algorithms

---

**Algorithm 4.1.1:** GENERATEDESCRIPTION(*group*)

---

```

procedure GENERATEDESCRIPTION(group)
  noun ← GETNOUN(group)
  if ONTOLOGY.LOOKUP(noun) ∈ (Instances)
    then { id ← ONTOLOGY.LOOKUP(noun)
           return ( $\mathcal{D} + \{ * \text{ sameAs } < id > \}$ ) }
    else  $\mathcal{D} = \mathcal{D} + \{ * \text{ type } < noun > \}$ 
    (1)

  det ← GETDETERMINANT(group)
  if det ∈ (possessives)
    then  $\mathcal{D} = \mathcal{D} + \{ * \text{ isRelatedTo } < possessor > \}$ 
  if det ∈ (demonstratives)
    then { if ONTOLOGY.CHECK( $\{ < currentSpeaker > \text{ focusesOn } * \}$ )
           then  $\mathcal{D} = \mathcal{D} + \{ < currentSpeaker > \text{ focusesOn } * \}$ 
           else  $\mathcal{D} = \mathcal{D} + \text{ANAPHORICMATCHING}()$  }
    (2)

  adjs ← GETADJECTIVES(group)
  for each adj ∈ adjs
    do { if adj == < other >
         then { id ← HISTORY.GETMATCHINGGROUP(group)
                   $\mathcal{D} = \mathcal{D} + \{ * \text{ differentFrom } < id > \}$ 
                  return (D) }
         else  $\mathcal{D} = \mathcal{D} + \{ * \text{ hasFeature } < adj > \}$  }
    (3)
    (4)

  nounComplements ← GETNOUNCOMPLEMENTS(group)
  for each nouncmpl ∈ nounComplements
    do  $\mathcal{D} = \mathcal{D} + \text{GENERATEDESCRIPTION}(\text{nouncmpl})$ 

  relativeClauses ← GETSUBORDINATERELATIVECLAUSES(group)
  for each relative ∈ relativeClauses
    do {  $\mathcal{G} \leftarrow \text{GETNOMINALGROUPS}(\text{relative})$ 
         for each g ∈  $\mathcal{G}$ 
         do  $\mathcal{D} = \mathcal{D} + \text{GENERATEDESCRIPTION}(g)$  }

  return ( $\mathcal{D}$ )

```

---

The method called at (3) consists in looking through sentences that have been stored in the conversation history, then extracting their nominal group in order to retrieve the identifier of the most recently mentioned concept that holds the same characteristics as the nominal group that is being processed (algorithm 4.1.2).

Anaphora matching (2) is described in chapter 4.2.

---

**Algorithm 4.1.2:** HISTORY.GETMATCHINGGROUP(*group*)

---

```

procedure HISTORY.GETMATCHINGGROUP(group)
  comment: Extract Nominal group from sentences stored in the history
   $\mathcal{H} \leftarrow \text{HISTORY.GETALLNOMINALGoup}()$ 
  comment: Generate description of the nominal group that is being processed.
  comment: The adjective "other" is to be removed before calling this routine
   $\mathcal{G} \leftarrow \text{GENERATEDESCRIPTION}(\text{group})$ 
   $\text{candidates} \leftarrow \mathcal{H} \cap \mathcal{G}$ 
  if  $|\text{candidates}| = 0$ 
  then { output (Couldn't find another object with the same characteristics!)
        exit
  else if  $|\text{candidates}| = 1$ 
  then  $\text{id} \leftarrow \text{candidates}[0]$ 
  else  $\text{id} \leftarrow \text{DISCRIMINATION}(\text{candidates})$ 
  return (id)

```

---



---

**Algorithm 4.1.3:** RESOLUTION(*sentence*, *currentSpeaker*)

---

```

 $\mathcal{G} \leftarrow \text{PARSENOMINALGROUPS}(\text{sentence})$ 
for each  $g \in \mathcal{G}$ 
  {
     $\mathcal{D} \leftarrow \text{GENERATEDESCRIPTION}(g)$  (1)
     $\text{candidates} \leftarrow \text{ONTOLOGY.FIND}(\mathcal{D})$  (2)
    if  $|\text{candidates}| = 0$ 
    then { output (Couldn't resolve the group!)
          exit
    else if  $|\text{candidates}| = 1$ 
    then  $\text{id} \leftarrow \text{candidates}[0]$  (3)
    do {
      else {
        if  $\text{ONTOLOGY.CHECKEQUIVALENT}(\text{candidates})$ 
        then  $\text{id} \leftarrow \text{candidates}[0]$ 
        else  $\text{id} \leftarrow \text{DISCRIMINATION}(\text{candidates})$  (4)
      }
       $\text{REPLACE}(g, \text{id}, \text{sentence})$ 
    }
  }

```

---

Based the partial statements generated at (algorithm 4.1.3, (1)), a query is sent to the ontology server to retrieve possible instances that match the description (algorithm 4.1.3, (2)).

The discrimination routine called at (4) is described separately.

Two remarks must be made that doesn't appear in alg. 4.1.3:

1. If a sentence starts with *Learn that...*, failures during discrimination are interpreted as new concepts, and instead of marking the nominal as not resolved, and new identifier is created.
2. For questions like *Which color is the bottle?*, the discrimination algorithm can not use the feature *color* to identify to bottle. The resolution algorithm pass this kind of constraints as a parameter of the DISCRIMINATION ROUTINES.

### 4.1.2 Adjectives ONLY approach

Let's consider the following example:

```
"the yellow banana is good"
```

and assume there exists a unique yellow banana in the ontology referenced as 'Y\_BANANA'. It is fairly possible to resolve the identifier of "the yellow banana", but how about the single information "good" ?

'adjectives\_only' is an approach implemented in order to assume as resolved, nominal groups holding information only in the adjective attribute. In doing so, we can easily derive the statements:

```
[Y_BANANA hasFeature good]
```

### 4.1.3 Quantifier approach

We have implemented some quantifiers that may be used to assume that a nominal group is resolved.

Let's consider the example "Danny is a human". Resolving "Danny" should succeed as long as there is a concept labelled "Danny". However, the nominal group "a human" cannot be resolved. Although the sentence is to commit new information, how can we assign to this particular nominal group a unique identifier? What if there are numbers of humans, which one are we talking about?

Therefore, we assume as resolved all nominal groups with the quantifier "SOME" and "ALL". In doing so, we can allow the creation of statement towards the ontology.

```
"Statement created"  
[DANNY rdf:type Human]
```

Regardless the semantic interpretation of a sentence, when both the subject and object hold indefinite quantifiers - either 'ALL' or 'SOME' - we create the predicate `"" rdfs:subClassOf ""`. Doing this allows us to create class grounding. When both hold the definite quantifier `"" ONE ""`, we create the predicate `"" owl:sameAs ""`. For other case, we use the object property `"" rdf:type ""`:

```
"Bananas      are      fruits"  
ALL           +      ALL           => rdfs:subClassOf  
[Banana      rdfs:subClassOf      Fruit]  
  
"A banana     is      a fruit"  
SOME          +      SOME          => rdfs:subClassOf  
[Banana      rdfs:subClassOf      Fruit]  
  
"the green banana is      a fruit"  
ONE           +      SOME          => rdf:type  
[green_banana rdf:type      Fruit]  
  
"the green banana is      the banana in the table"  
ONE           +      ONE           => owl:sameAs  
[green_banana owl:sameAs      green_banana]
```

cf: Parsing for quantifier list

- Problem:

The quantifier approach is useful only if we process sentence with the state verbs (e.g: to be).

E.g: Bananas grow on Trees

This would produce

```
[* performedBy Banana,  
 * rdf:type Grow,  
 * involves Tree],
```

This would transform "Banana" into an instance, and severely break a future test referencing "Banana" as a Class.

#### 4.1.4 Demonstratives determiners approach

Let's suppose we want to say something similar to this: " *This is on the shelf*" or "*this is green*". Processing demonstrative determiners such as "this" assumes, there exists in the ontology a statement such as '[ACHILLE focusesOn A\_CUBE]', where the "ACHILLE" is to mention the current speaker and 'A\_CUBE' the reference of the concept that is being pointed by the current speaker. Therefore, any occurrence of the determiner "this" affects to its parent nominal group the identifier 'A\_CUBE'.

##### No focus?

What if there is nothing pointed by the current speaker? The resolution of the parent nominal group will involve either anaphora matching or discrimination.

Let's consider the example below on "Take this one!" or "Take this!". Would you get that the Human means to take "the green cup" or "the red bottle"? Resolving the nominal group that hold "this" implies the use of anaphora matching that is to replace the "this one" with a possible object that has been stated earlier in the conversation.

E.g:  
 [Human] - What are the objects on the table?  
 [Robot] - The green cup and the red bottle  
 [Human] - Okay, I'll get the green cup. You, take this one! //Here, the Human is not pointing any object

However, if instead of "take this one" , the *Human* says: "Take this bottle". Resolving the nominal group involves discrimination.

```

1  # MODULES
2  # - Oro: this offers services of the ontology server
3  # - AnaphoraMatcher: this offers methods to resolve anaphoric words
4  # - Discrimination: this offers methods to discriminate a nouns, using theirs statements
5  descriptions
6  # ROUTINES
7  # - get_description(): this provides statements with the description of the nominal group that
8  is called in its parameter
9  # - get_noun_history(): this provides a list of all recent nominal groups that have been
10 involved during a conversation
11
12 def Resolve_demonstrative_determiner(noun):
13
14     # Retrieving in the ontology the concept that is pointed by the current speaker
15     # if there exists one, then the resolution of 'this' or 'that' is done
16     ontology_candidates = Oro.find('?concept', [current_speaker + ' focusesOn ?concept'])
17     if ontology_candidates:
18         id = ontology_candidates[0]
19
20     # if there is no concept pointed by the current speaker
21     # then attempting to resolve it with anaphora matching or discrimination
22     else:
23         if is_anaphora(noun):
24             id = AnaphoraMatcher.match(noun, get_noun_history())
25
26         else:
27             # Retrieving in the ontology all the concepts of the same description as the one
28             that is to be resolved.
29             # Then discriminating
30             id = Discrimination.clarify(get_description(noun))
31
32     return id

```

#### 4.1.5 Occurrence of the words 'OTHER'

Let's consider the dialogue below with an occurrence of the word 'other'. Processing "give me the other one" or "Give me the other bottle" consists in determining a possible bottle that has been stated earlier in the conversation, and looking up through the ontology bottles different from that one.

E.g:

[Human] - What is on the blue table?  
[Robots] - The green bottle and the blue bottle  
[Human] - Give me the green bottle.  
[Human] - Now, give me the other one

```
1     def Resolve_other(noun):
2         # Retrieving in the ontology all the concepts of the same description as the one that is to
3 be resolved.
4         # Then attempting to identify it with the intersection of ontology's concept candidates and
5 history
6         candidates = Oro.find('?concept', get_description(noun))
7
8         # Intersection
9         candidates = [c for c in get_noun_history() if c in ontology_candidates]
10
11         # If there exists some candidates from the preceding intersection ,
12 # the one that is to be retrieved is the first one in the list ,
13 # as the intersection has ordered the list elements according to the nominal group history
14         if candidates:
15             id = Discrimination.clarify(get_description(noun) + [ '?concept owl:differentFrom ' +
16 candidates[0] ])
17
18         return id
```

#### 4.1.6 Handling Unidentified Anaphora error

- Intercepting the Unidentified Anaphora error
- if there exist a current object possibly after human confirmation, filling in the nominal group with an anaphoric word with it
- Going to discrimination

Cf Anaphora matching

#### 4.1.7 Handling Insufficient input error

- Retrieving the nominal group with insufficient information
- Replacing it with the merged nominal group
- Going to discrimination

Cf: Discrimination

### 4.2 Anaphora Matching

Anaphora processing consists in replacing the anaphoric word that is to resolve (Cf. Resolution) with a concept that has been mentioned earlier in the conversation. To do so, we collect the conversation history of the sentences that have been successfully processed in the Dialogs module. Two cases are to be specified that is, the case when the anaphoric word refers to an agent, which consists in replacing personal pronouns – Him, Her ... - with an agent's identifier; and the case when it refers to an object's identifier. Here, we discuss the latter case.

Let's consider the example below:

[Human] - Jido, the bottle is on the table next to me. I put the bottle on the shelf next to you.  
[Robots] - Okay!  
[Human] - Now, give it to me. (Occurrence of the anaphora 'it' that is to be identified).

The first step consists in gathering candidate nominal groups, among sentences that occurred before the one holding the anaphora. The extraction is made in the reverse order of history's sentences that is to say; the first sentence that has been said would be the last one in the extraction process. However, the order of nominal groups of the same sentence is preserved. Thus, the list that is produced is the following:

From the last sentence - [I, the bottle, the shelf, you]

From the last but one sentence - [Jido, the bottle, the table, me]

Candidate list

[I, the bottle, the shelf, you, Jido, the bottle, the table, me]

The second step consists in producing a list with no redundant nominal group identifier:

[I, the bottle, the shelf, you, the table]

Then, as we deal with objects only, we produce a list with no reference to agents. The list of candidate nominal groups is now created.

[The bottle, the table, the shelf]

As the Dialogs module does not deal with conversation context and due to robot's self-comprehension limits, what follows is a questions-and-answers loop between the robot and the human in order to identify the correct nominal to match the anaphora:

[Robot] { Do you mean the bottle? (Robot asking for confirmation { Do you mean that I give you the bottle?})

[Human] { No ... (Possibly another answer such as 'No, I mean the cup')

[Robot] { Do you mean the table?

The human answer presents three possible aspects.

- If it is an agreement from the robot's question the questions-and-answers loop breaks. Thus, the anaphora is identified.

[Human] { Yes,

- If it is a disagreement and the human answer possibly holds the nominal group to match the anaphora. The Merge process is then called to build a new sentence that will carry on through the Dialogs module (Cf. Merge).

[Human] { No, I mean the glass.

- If it is a disagreement and the human answer does not hold enough information to match the anaphora, the questions-and-answers loop carries on with the next element of the list and the robot's question for human confirmation.

[Human] – No.

```
1  --The algorithm:--
2  if anaphora(sentence) == 1:
3      nom_gr_list = find_nominal_groups_list(memory)
4      first_replacement(nom_gr_list[0], current_nom_gr)
5      nom_gr_list = nom_gr_list[1:]
6      loop:
7          print (question_to_know_if_it_is_the_nominal_group)
8          waiting_user_utterance (utterance)
9          sentence_analysis = parsing (utterance)
10         anaphora_tuple = process_anaphora(sentence_analysis)
11         if anaphora_tuple[1] == True :
12             exit
13         elif anaphora_tuple[0] == None :
14             first replacement with the nom_gr_list[0]
```

## 4.3 Discrimination

The discrimination routine consists in retrieving a nominal group identifier from the ontology given its description. When several concepts happen to hold similar but not equivalent descriptions, it is offered to the user the mean to give more information.

```
1  def discrimination(noun):
2      # Retrieving in the ontology the concept that matches the given description
3      candidates = Oro.find('?'concept', get_description(noun))
4
5      # If nothing found, ask for more information to the User
6      # The status of the discrimination is assigned to "FAILURE"
7      # E.g: The Corrida? What do you mean?
8      if not ontology.candidates:
9          raise UnisifficientInputError({What.do.you.mean(noun),
10                                         status:"FAILURE" })
11
12     # If anything found
13     # if it is a single object, the discrimination has succeeded.
14     # if there are two objects, check their equivalence
15     # if there are several more, ask for more information
16     # E.g: Which one? the red or the blue bottle?
17     # The Status of the discrimination is assigned to "SUCCESS"
18     else
19         if len(candidates) == 1:
20             id = candidates[0]
21
22         elif len(candidates)== 2 and Oro.check(candidates[0] + " owl:sameAS " + candidates[1])
23             id = candidates[0]
24
25         else:
26             raise UnisifficientInputError({Which.one.do.you.mean(candidates),
27                                             status:"SUCCESS" })
```

### 4.3.1 Merge

The Merge routine consists in appending information to the nominal group that is being processed and occurs after a failure in the discrimination process (Cf.Discrimination). It takes into consideration the insufficient input error status – either “FAILURE” or “SUCCESS” – (Cf.Discrimination)

- In case of “SUCCESS”

The discrimination has retrieved several concepts matching the description of nominal group that is being processed. Further information is needed to accurately identify this last one. Therefore, the user is expected to enter a new utterance. E.g.: [Human] – Jido, give me the bottle. [Robot] – Which bottle do you mean? The red or the yellow one? [Human] – The red one.

Here, the unresolved nominal group -‘the bottle’ - is merged with more information - ‘the red bottle’.

- In case of “FAILURE”

The discrimination has not retrieved any concept matching the nominal group description. Either, the concept does not exist or the entered utterance is mistaken. The human is expected to enter new information that is to replace the current one.

E.g.:

```
[Human] { Jido, give me the carpet.
[Robot] { The carpet? What do you mean? Give me more information...
[Human] { Sorry, I mean the red cup
```

Here, ‘the carpet’ is turned into ‘the red cup’

Regarding these two preceding aspects, we state the following rules: Let A be the nominal group that is being processed and because of which the insufficient input error has been raised.

Let B be the utterance holding the nominal group with further information.

```
Case ‘SUCCES’ {
  []+[]=[],
  []+B=B,
  A+B=AB,
  A+[]=A
}
```

```

Case 'FAILURE' {
  []+[]=[],
  []+B=B,
  A+B=B,
  A+[]=A
}

```

When applying those rules, the state of the utterance B is also to be taken into consideration.

- In case of an affirmative sentence, the merging process goes through each nominal group and output the merged one, once the whole sentence has been visited. Attributes holding relative or conjunctive subordinate clauses are processed recursively.

The nominal group that is to be merged will be computed according to the rules mentioned below.

Notice: The user may enter a common noun phrase with no determiner. It is then parsed as an imperative sentence, with the noun phrase taken as a verb. When this case is detected, we first turn the sentence type into a "statement", where the noun phrase would be seen as the subject. Eg:

```

[Robot]: - Do you mean the bottle or the glass
[Human]: - bottle

```

After parsing, *bottle* is a verb in an imperative sentence. After the change, "bottle" becomes the subject in a Statement sentence.

If the noun phrase starts with a capital letter:

```

[Robot]: - Do you mean the bottle or the glass
[Human]: - Bottle

```

Here "Bottle" is seen as a proper noun. The merging process effects only on statement or subsentence type.

```

[Robot]: - Do you mean the bottle?
[Human]: - No.

```

Here we have no merge because the data type of the sentence is "disagree"

- In case of a negative sentence, the insufficient input error status holds the value 'FAILURE' – if not, it is changed so. If the utterance B hold both a verbal structure, nominal groups that are contained in it are processed only if their conjunctive attribute is assigned the value "BUT".



## Chapter 5

# Outputs of the Dialogs module

### 5.1 Statement builder

This module aims to build RDF<sup>1</sup>/OWL statements corresponding to the sentence that is being processed. Let's notice that, statements are created either for the resolution of sentences or for querying or committing the ontology. Here, we describe the latter case, and explain what feature of the sentence can be fully processed.

#### 5.1.1 Simple sentences

Let's assume there exists a unique agent labelled 'Danny' in the ontology with the reference 'DANNY'; also, let's assume there is a only one blue car with the reference 'blue\_car'. Therefore we create the following statements:

E.g:

"Danny drives the blue car"

```
['EVENT rdf:type Drive',  
'EVENT performedBy DANNY',  
'EVENT involves blue_car']
```

where EVENT is to mention a static situation reference that is to be generated.

- Inconsistency:

What happens if we say "the blue car is red"; that is to generate [blue\_car hasColor red] ? What if we say "this cylinder is green" whereas the current speaker is pointing a cube; that is to generate such statements [CUBE rdf:type Cylinder]? Attempting to commit those statements would also lead the ontology to an inconsistent state.

Leading the ontology to an inconsistency state occurs every time we try to commit a new information that cannot be inferred accurately with the existing ones. In this project, we use the methods "safeAdd" and "safeAddForAgent" to overcome this problem. (Cf. Oro-server).

#### 5.1.2 Relative sentence processing

There are two ways of processing the relative, which are derived by the subject of the sentence, as it may also be the subject of the relative:

E.g: "the man that is talking is my boss"

Or it may be an object of the relative. In this case, the subject of the sentence is duplicated. (Cf. Parsing)

---

<sup>1</sup>Resource Description Framework

E.g: "the man that you see, is my boss" is turned into "the man that you see + the man, is my boss".

Processing this case involves checking if the subject equals any of the object in the relative sentence, in order to keep track of the subject's reference.

### 5.1.3 Verb processing

In most cases, processing a verb consists into generating a reference for a situation, that is performed by an agent and involves an object, regarding a circumstance that could represent a temporal or spatial aspect, or even the fact of an agent receiving something.

E.g: "I drive my car in Toulouse"

Generated statement:

```
[EVENT performedBy MYSELF, EVENT rdf:type Drive, EVENT involves MY_CAR, EVENT isIn TOULOUSE]
where EVENT, MYSELF, MY_CAR and TOULOUSE are unique references of respectively the situation that
is to describe, the current speaker, the car of the current speaker and the city of Toulouse.
```

Here , only the reference EVENT is generated whereas the others are obtained form resolution.

#### State verbs (E.g: to be, to become)

In the case of state verbs, we do not describe a situation, but the subject of the sentence. This will consist in either class grounding (cf Quantifier approach) or features description.

#### Thematic roles

Cf /share/dialogs/thematic\_roles

#### Goal verb

Goal verbs are used to express the speaker's desire.

E.g: "I want to get the blue cube"

Generated statements:

```
[MYSELF desires EVENT,
...
]
```

#### Action verbs with passive behaviours (E.g: to see, to hear, to reach, to know)

```
[MYSELF sees EVENT,
...
]
```

**To know** For the case of 'to know', we first create a set of statements such as '[MYSELF knows CONCEPT, ...]', then we update or query the ontology on the 'CONCEPT' for an agent that is supposed to be MYSELF

```
Ontology.lookupForAgent(MYSELF, CONCEPT)
```

### 5.1.4 Negative sentence approach

This approach consists in committing a negative assertion, without leading the ontology into an inconsistent state. Three forms of negations can be pointed out. The negation of the type (E.g: "rdf:type" or "rdfs:subclassOf" ), the negation of the property (E.g: "hasFeature") and the negation of actions. The latter may bring numbers of ambiguity . Let's consider all the negation with the following examples:

#### Negation of the type

"Jido is not a Human".

Here, we create the following statements:

```
[JIDO rdf:type ComplementOfHuman], JIDO is the reference of "jido" in the ontology.
```

Also, we add these statements to ensure the consistency:

```
[ComplementOfHuman owl:complement of Human]
```

and

[ComplementOfHuman rdfs:subclassOf ComplementClasses], where the latter is to help us retrieve easily all the Complement types added in the ontology.

#### Negation of the predicate (object\_property)

"Jido is not gray".

Here, we create the following statements:

```
[JIDO hasColor xxx,  
xxx owl:differentFrom gray]
```

where xxx is a generated reference to a colour different from gray.

#### Negation of actions

"Jido does not drive the blue car'.

This sentence holds numbers of ambiguities. It may be interpreted as "Jido drives a car, but not the blue one", or

"Jido somehow acts on the blue car, but does not drive it" , or so on.

Here we do not take into consideration any of those ambiguities. We create the statements as if we were dealing with an affirmative sentence.

```
[?xxx rdf:type Drive,  
?xxx performedBy JIDO,  
...  
]
```

Then, we remove these statements after identification of ?xxx in the ontology.

### 5.1.5 verb tense approach

This approach aims to specify whether an action occurs in the past or in the present. Therefore, we create the object property 'eventOccursIn' and bind it with the flag PAST or FUTUR. The present tenses are assumed

as a default case; there is nothing to do.

E.g: Danny 'went' to Toulouse.

We create the statements:

```
[ACTION rdf:type Go,  
ACTION performedBy DANNY  
...  
ACTION eventOccursIn PAST]
```

### 5.1.6 Adverb modifying the verb - processing approach

The purpose of this approach is state the way the action is undertaken, which may be slowly, carefully, quickly, or so on. Assuming that this type information is taken in charge by the supervision module, we create such statements:

E.g: Danny moves slowly.

```
[ACTION rdf:type Move,  
ACTION performedBy DANNY  
...  
ACTION actionSupervisionMode SLOW]
```

### 5.1.7 Adverb modifying the sentence - processing approach

Cf /dialogs/timescale\_manager.py

## 5.2 Question handler

The purpose of this module is to query the ontology regarding the question's aim and type.

### 5.2.1 Wh-question

Processing a Wh-Question consists in, at the first step, building statements based on the sentence's structure. The second step determines the *query\_on\_field* parameter, then orientates the statement set according to the question's aim. Finally, the resulting statements are used to query the ontology.

Whichever its data type, a sentence is always held in the same data structure - (Cf: Data structure). Therefore, we can call the Statement builder module with more or less modifications in the first step Let's consider the following example: "who gave you the red cup?"

Parsed Sentence:

```
W-Question:  
AIM: People  
subject: //empty//  
direct object: "the red cup"  
indirect complement: "you"
```

Generated Statements from the statement builder module depending on the sentence structure:

```
[?EVENT rdf:type Give,  
?EVENT actsOnObject RED_CUP  
?EVENT receivedBy MYSELF]
```

Regarding the *Parsed Sentence* structure, the example is to answer the subject, we then need to extend the statement set with

```
[?EVENT performedBy ?CONCEPT]
```

where "*?CONCEPT*" is the identifier of the subject, which the answer that should be provided by the ontology.

The second step consisting in determining the part of the Sentence structure that is to be answered and how to extend statements has been implemented by the use of a map that takes into consideration the *aim* attribute, the verb and the *query\_on\_filed* parameter.

Depending on the verb, whether it is of thematicRole type, an action verb or a state verb, an object property is specified based on, if the answer is to complete the direct object or indirect complement, if it is to determine the reception of object by an agent, an action occurring at spatial or temporal concept, etc. . .

However, let's state a couple of remarks regarding questions starting with "WHAT" or "WHICH".

- There is as many value for the *aim* attribute as concept in the world, making it impossible to generate a complete map.

For instance in "What color do you prefer?" and "what buidling is next to the Eiffel Tower?", the *aim* attribute respectively holds the value "color" and "building".

- When the *aim* attribute refers to concept's features such as "color", "size", . . . the extension of the statements would consists in generating a token similar to `[RED_CUP hasFeature ?CONCEPT]`. For the case of non-feature such as "Building", "artifact", "object", . . . we generate a concept descriptor such `[?CONCEPT rdf:type Building]`

The Question Handler module is however not complete. It is possible to handle question starting with "What", "Which" and "Who". However, several types such as those started with "Whom", "Whose", "Why", "When", "How" are not implemented yet.

### Query\_on\_field

This field is to determine the part of the sentence structure that is to modify with the query answer Three values are presented in Question Handler that are *QUERY\_ON\_DIRECT\_OBJ*, *QUERY\_ON\_INDIRECT\_OBJ* and the default value *None*. For instance, if we have to process "Where is the cube?", the question can be turned into "The cube is on xxx " where, the answer 'xxx' is to fill the indirect complement of the sentence. Therefore, *query\_on\_field* is assigned the value *QUERY\_ON\_INDIRECT\_OBJ*

## 5.2.2 Yes-No-Question

Processing the Yes-No-questions solely consists in building the corresponding statements to the data structure of the Sentence; then checking their existence in the ontology.

E.g: "is Jido Taking the cube?"

Statements to be checked:

```
[EVENT performedBY JIDO,  
  EVENT rdf:type Get,  
  EVENT actsOnObject CUBE]
```

The ontology may return TRUE or FALSE. If True, we return to the user the fact that "Jido is taking the cube" exist and can be confirmed. If false, we do not return to the user that the fact is false, as it may be True, but not existing in the Robot's ontology model.

- Case of Can-You-do-something?

Let's suppose the user says "Jido, can you give me this cube?". After parsing this sentence, it appears to be a Yes-No-Question. It could means for us to either verify Jido's ability of giving a certain cube or generate the user's desire of receiving it. The later case is the one that has been implemented when we deal with such Yes-No-Questions - those holding the modal verb *can* and a default action verb Verb processing.

## 5.3 Verbalization

The verbalization is to read information sent over the *Sentence* structure and assemble them following a precise order given by the grammar rules to obtain a correct sentence as output.

### 5.3.1 Sentence type of verbalization

Thanks to `data_type`, we can find the global form of the sentence (Cf Sentence type). Except for questions with aim as “people” are always built with “who”. For “how mush” and “haw many”, we will have “how mush”. “what kind” and “what type”, we will have “what kind”.

### 5.3.2 The transformation which are always done

For all these cases, we can limit the processing. We have just to add some conditions.

Some contractions are always done.

- What is ? What's
- He is ? He's
- I would ? I'd
- There are ? There're
- ...

Possession case is all time processed

- the bottle of Jido ? Jido's bottle
- the bottle of me ? me's bottle

There are many other processes which used to have a proper and correct sentence (like the inversion of the preposition...)

## Chapter 6

# Examples

### 6.1 Debug output of a complete execution

The following example shows a complete execution of an imperative sentence.

E.g: "put the red cup next to the orange bottle"

```
1  -----[      NL INPUT      ]-----
2
3  - put the red cup next to the orange bottle
4
5
6  -----[      PARSING      ]-----
7
8  | >> IMPERATIVE
9  | verbal grp:
10 |     put (present simple)
11 |     direct objects:
12 |         the red cup
13 |
14 |     indirect objects:
15 |     next+to...
16 |         the orange bottle
17 |
18 |
19 |     >not resolved<
20 |
21 |
22
23 -----[ RESOLVING SENTENCE ]-----
24
25 Resolving references and anaphors...
26 cup is an existing concept in RAQUEL's model.
27 bottle is an existing concept in RAQUEL's model.
28
29 Resolving noun phrases
30 "the red cup"
31 cup is being processed as a common noun in RAQUEL's model.
32
33 Looking for this concept in RAQUEL's model:
34 [?concept rdf:type Cup, ?concept hasColor red]
35
36 Found these possible concepts ID: ['red_cup']
37 Hurra! Found "red_cup"
38
39 "the orange bottle"
40 bottle is being processed as a common noun in RAQUEL's model.
41
42 Looking for this concept in RAQUEL's model:
43 [?concept rdf:type Bottle, ?concept hasColor orange]
44
45 Found these possible concepts ID: ['orange_bottle']
46 Hurra! Found "orange_bottle"
47
48 Resolving verbal groups
49 "put"
50 Keeping "put"
51
52 [ Sentence after resolution ]
53 | >> IMPERATIVE
54 | verbal grp:
55 |     put (present simple)
```

```

56 |         direct objects:
57 |             red_cup
58 |             >resolved<
59 |         indirect objects:
60 |         next+to...
61 |             orange_bottle
62 |             >resolved<
63 |
64 |     >resolved<
65 |
66 |-----[  CONTENT ANALYSIS  ]-----
67 |
68 | Processing the content of an imperative sentence
69 | Generated statements:
70 |     >> ZQ7r8 rdf:type Put
71 |     >> ZQ7r8 performedBy myself
72 |     >> RAQUEL desires ZQ7r8
73 |     >> ZQ7r8 actsOnObject red_cup
74 |     >> ZQ7r8 isNextTo orange_bottle
75 |
76 | Statements added to the ontology.
77 |
78 | Nothing to verbalize.
79 | Sentence saved in history.
80 |
81 | [ NL sentence "put the red cup next to the orange bottle" processed! ]

```



# Chapter 7

## User studies

This chapter presents several longer dialogues between a human and the robot. The user was sitting in front of a table with several objects on it, and was asked to interact through a chat interface (Jabber) with the robot.

**These studies have been conducted in Wizard of Oz:** answers were manually sent by one of the DIALOGS developer, not the robot itself.

Experimental results were *a posteriori* processed by the parser (and the parser alone) to check which inputs would be parsed into a correct sentence structure.

### 7.1 Relation between speakers

#### 7.1.1 First dialog with Mahdi

Speaker	Utterance	Input correctly parsed?
user	hello Jido	ok
Jido	hello	
user	what are you doing	ok
Jido	I am doing nothing	
user	ok can you tell me what you see on the table	x
	<i>[Replaced by: "ok, can you tell me what you see on the table"]</i>	
Jido	I see the blue pen that is next to the black object	
user	what is that object?	ok
Jido	that object? what do you mean?	
user	the black object	ok
Jido	the black object is the black object that is on the green table	
user	ok, this object is a mouse	ok
Jido	do you mean the black object that is on the green table?	
user	yes	ok
Jido	ok	
user	now, I will write a dialog in 5 minutes, so can you give me the pen	ok
Jido	yes	

#### 7.1.2 Second dialog with Patrick

Speaker	Utterance	Input correctly parsed?
user	hi	ok

Speaker	Utterance	Input correctly parsed?
Jido	hello .	
user	what is your name?	ok
Jido	my name is Jido .	
user	ok ok. and you are a robot of the Laas?	ok
Jido	yes. I am a robot of the laas .	
user	do you know Mahdi. he is my colleague	ok
Jido	yes I know Mahdi.	
Jido	Do you mean Mahdi?	
user	yes	ok
Jido	ok	
user	did you see him last 5 minutes	ok
Jido	no .	
Jido	I did not see him .	
user	who is the last person who talk to you?	ok
Jido	the last person I talked to is you .	
user	and before me?	ok
Jido	I don't know .	
user	Learn that this person is mahdi	x
	<i>[Replaced by: "learn that this person is Mahdi"]</i>	
Jido	ok	
user	did he start his dialog?	ok
Jido	yes he did .	
user	ok	ok
user	when he finishes, tell him that I want to see him	ok
Jido	ok.	
user	thanks goodbye	x
	<i>[Replaced by: "thanks, goodbye"]</i>	
Jido	Goodbye.	

### 7.1.3 Third dialog with Mahdi

Speaker	Utterance	Input correctly parsed?
user	good afternoon Jido	ok
Jido	hello .	
user	I think that Patrick want to see me? did you see him?	ok
Jido	Ok. No. I did not see Patrick	
user	ah ok.	ok
user	this is your pen	ok
Jido	ok .	
user	you can take it	ok
Jido	ok .	
Jido	do you want to see Mahdi ?	
user	no I'am mahdi	x
	<i>[Replaced by: "no, I'm Mahdi"]</i>	
user	do you know what i have written?	ok
Jido	no. I don't know what you have written. what is it?	
user	I finish the dialog and I check many problems	ok
Jido	ok. what are problems?	
user	the anaphora is not good	ok
Jido	the anaphora? what do you mean ?	

Speaker	Utterance	Input correctly parsed?
user	It is a rule of interpretation. it allows a relation between the dialog and the current sentence	ok
Jido	do you mean the anaphora?	
user	yes	ok
Jido	ok.	
Jido	what is sentence ? and dialog?	
Jido	and interpretation ?	
user	sentence is a set of a word	ok
Jido	ok .	
user	to have a dialog, we need more than 1 protagonist. and in a dialog, there is an interaction between them	ok
user	the interpretation is to find a defenition or a rule for something	ok
Jido	ok . I am sorry . I cannot process this entry due to its difficult structure for my dialog modules.	
Jido	I have learnd that user wants to see mahdi.	
user	who is user?	ok
Jido	user is user	
user	ok I think that I know who is he	x
	<i>[Replaced by: "ok, I think that I know who is he "]</i>	
user	he is Patrick	ok
Jido	ok .	
user	so I want to go	ok
Jido	where do you want to go?	
user	I will go to see Patruck	ok
Jido	who is Patruck ?	
user	sorry Patrick	x
	<i>[Replaced by: "sorry, Patrick"]</i>	
Jido	Do you mean you will go to see Patrick?	
user	yes	ok
Jido	ok .	
user	ok see you	x
	<i>[Replaced by: "ok, see you"]</i>	
Jido	bye . Hope we talk soon .	

## 7.2 Spy Game

The *Spy Game* is usually played with two children: one think of an object without telling it, and the other one must guess which one by asking yes/no questions.

Speaker	Utterance	Input correctly parsed?
user	hello jido	ok
Jido	hello	
user	what do you want to do?	ok
Jido	I don't know if I want or not	
Jido	and you?	
user	maybe play to a game	ok
Jido	ok. What is maybe?	
user	can, sorry	x
	<i>[Replaced by: "sorry, can"]</i>	
Jido	ok.	

Speaker	Utterance	Input correctly parsed?
user	to you propose which game	x
	<i>[Replaced by: "do you propose which game"]</i>	
user	are you ready?	ok
Jido	Yes.	
user	do you want to choose the game?	ok
Jido	I don't know if I want or not.	
Jido	and you?	
user	yes, I want	ok
Jido	ok.	
user	explain me the rules	ok
Jido	ok.	
Jido	rules of what?	
user	explain me the rules of the spy game	ok
user	please	ok
Jido	Spygame is a logic game involving two players. It is played with a set of objects that holds several similar features. The object of the game is to guess the opponent's choice of object, while pointing questioning about the object features. The winner is the player who manage to guess the opponent's choic of object in the smallest number of trials	
user	i see, i am ready to start	x
	<i>[Replaced by: "i see. i am ready to start"]</i>	
Jido	ok	
Jido	Choose and object to start	
user	done	ok
Jido	is it a bottle or a cup?	
user	my object is a bottle	ok
Jido	is it green?	
user	no, sorry	ok
Jido	is it blue?	
user	try again	ok
Jido	do you mean? another object?	
user	it is not blue	ok
Jido	ok	
Jido	is it orange?	
user	no	ok
Jido	it is the gray bottle	
user	yes	ok
Jido	I achieved it in \$ trials	
Jido	and you?	
user	can you give me it?	ok
Jido	ok.	
user	thanks	ok
Jido	you are welcome	
user	can you give me another one?	ok
Jido	yes. which one?	
user	does not matter	x
	<i>[Replaced by: "it does not matter"]</i>	
Jido	matter? what do you mean?	
user	anyone	
Jido	do you mean the blue or the orange or the green bottle?	
user	the orange one	ok

Speaker	Utterance	Input correctly parsed?
Jido	ok.	
user	thanks	ok
Jido	you are welcome.	
user	do you want to play again?	ok
Jido	the spygame process is still running. I picked an object. Will you guess it?	
user	yes	ok
Jido	I don't know if I want or not.	
Jido	and you?	
user	i want	ok
Jido	ok.	
user	is the object orange?	ok
Jido	no.	
Jido	try again	
user	is it blue?	ok
Jido	yes	
user	is it the blue bottle?	ok
Jido	no.	
Jido	try again	
user	i think it is the blue cup	ok
Jido	yes congratulation!	
user	thanks	ok
Jido	you achieved the game in 4 trails	
user	can you put it into the ravish	ok
Jido	the ravish? what do you mean?	
user	can you put it into the trash bin	x
	<i>[Replaced by: "can you put it into the trashbin"]</i>	
Jido	do you mean the trashbin?	
user	yes	ok
Jido	ok	
user	can you recover the cup?	ok
Jido	recover? do you mean take?	
user	take it again, please	ok
Jido	do you mean the blue cup?	
user	yes	ok
Jido	ok	
user	can you throw it away?	ok
Jido	Do you mean the blue cup?	
user	yes	ok
Jido	ok	
user	did you enjoyed the game?	ok
Jido	enjoy? do you mean throw away?	
user	no	ok
Jido	enjoy? do you mean take?	
user	do yow know me?	x
	<i>[Replaced by: "do you know me?"]</i>	
Jido	enjoy? do you mean know?	
user	no, forget it	ok
Jido	ok	
user	nice to meet you	ok
Jido	the spygame process has been activated. Do you want to play?	
user	no, thanks	ok
Jido	ok	
user	i must go	ok

Speaker	Utterance	Input correctly parsed?
Jido	ok	
user	where am i	ok
Jido	I don't know	
user	i am in the green room	ok
Jido	ok	
user	where am i?	ok
Jido	you are in the green room	
Jido	what is room?	
Jido	I gathered this information: You must go. is it confirmed?	
user	yes	ok
user	bye bye	ok
Jido	ok.	
Jido	bye	

## 7.3 Cleaning the table

### 7.3.1 First user

Speaker	Utterance	Input correctly parsed?
user	Hey Jido!	ok
Jido	hello	
user	Let's clean the table	x
	<i>[Replaced by: "let's clean the table"]</i>	
Jido	ok	
user	Pick up the papers, and throw them to trashbin	x
	<i>[Replaced by: "Pick up the papers, and throw them to the trashbin"]</i>	
Jido	ok	
user	I will arrange the bottles	ok
Jido	ok	
user	Give me the lotr_tape	ok
Jido	ok	
user	Now put the cups to the blue box	ok
Jido	which blue box? the big or the small one?	
user	the small one	ok
Jido	ok	
user	wait	ok
user	take the pens	ok
user	give me those cups	ok
Jido	which pen? the red, the balc or the blue one?	
user	all the pens	ok
Jido	those cups? what do you mean?	
user	forget the cups	ok
user	put the pens to the red box	ok
Jido	ok	
Jido	which pen?	
user	Where is the red cup?	ok
Jido	on the table	
user	Can you take it?	ok
Jido	no. I cannot reach it	
user	OK, I will give it to you	ok
Jido	ok	

Speaker	Utterance	Input correctly parsed?
user	thow it to the red box	ok
Jido	ok	
user	good boy	ok
Jido	ok	
user	Bye Jido	ok
Jido	bye	

### 7.3.2 Second user

Speaker	Utterance	Input correctly parsed?
user	where is the trashbin	ok
Jido	the trashbin is at the left	
user	at the left of what?	ok
Jido	the left of what ?what do you mean?	
user	I mean at the left of me or you	
	<i>[Parser failed]</i>	
Jido	do you mean at the left of me? the trashbin?	
user	yes	ok
Jido	yes. the trashbin is at my left	
user	ok	ok
user	thanks	ok
Jido	You are welcome	
user	could I reach the trashbin	ok
Jido	I don't know if you could or not.	
user	could you please put cassette jidoi in the trashbin	x
	<i>[Replaced by: "could you please put cassette_jidoi in the trashbin"]</i>	
Jido	casette jidoi? what do you mean?	
user	JIDO-E Tape	x
	<i>[Replaced by: "JIDO-E_Tape"]</i>	
Jido	ok	
Jido	I cannot reach it	
user	ok, I will do it for you	ok
Jido	ok.	
user	could you put red paper in red box	ok
Jido	I cannot reach it	
user	i will give it to you	ok
Jido	ok	
user	could you give me the orange bottle	ok
Jido	yes	
user	great	ok
Jido	Alright	
user	can you write a letter for me	ok
Jido	I don't know if I can or not	
user	no problem i will do it myself. could you pass me the pen	x
	<i>[Replaced by: "no problem. i will do it by myself. could you pass me the pen"]</i>	
Jido	ok. which pen?	
user	the black one	ok
Jido	ok.	
user	also pass me some papers, please	ok
Jido	the process Cleaning the table has been activated. Do you want to do it?	

Speaker	Utterance	Input correctly parsed?
user	no, give me some white papers	ok
Jido	ok.	
Jido	which papers?	
user	white one	ok
Jido	ok	
user	do you remember which pen you gave me. I will need the blue one now	ok
Jido	Remember? is it like give?	
user	yes	ok
Jido	I gave you the orange box and the white paper	
user	you did not give me the orange box but the orange bottle	ok
Jido	ok	
user	you also forgot that you gave a black pen	ok
Jido	forget? is it like give?	
user	ah! I need a drink. could you please fetch me orange cup	ok
Jido	ok.	
user	please, I am leaving. can you arrange table for me, put the left over orange bottle in the fridge and rest in trashbin	
	<i>[Parser failed]</i>	
Jido	ok. what are over, fridge and rest?	
user	orange bottle in the fridge, and wash the orange cup, and put it in the kitchen	ok
Jido	ok	
user	also don't forget to clean the table	ok
Jido	ok	
user	put blue bottles in the blue bx	ok
Jido	bx? what do you mean?	
user	blue box	ok
Jido	do you mean the big or the small one?	
user	on your left	ok
Jido	ok.	
Jido	I cannot reach it	
user	ok, put it in the blue box on my left	ok
Jido	do you mean the red or the blue one?	
user	the blue bottles in the blue box on our right	ok
Jido	ok.	
user	dump the green bottle on our right	ok
Jido	dump? is it like take?	
user	throw the green bottle in the blue box on our right	ok
Jido	ok	

### 7.3.3 Third user

Speaker	Utterance	Input correctly parsed?
user	Hello robot.	ok
Jido	hello	
user	Let's clean this table	x
	<i>[Replaced by: "let's clean this table"]</i>	



Speaker	Utterance	Input correctly parsed?
Jido	ok	
user	I start.	ok
Jido	ok	
user	Have you seen what I have done ?	ok
Jido	yes	
user	Good.	ok
Jido	good.	
user	Now, I will continue.	ok
Jido	ok.	
user	Can you tell me what I have done ?	ok
Jido	you put the blue bottle in the smal blue box	
user	Ans what else ?	x
	<i>[Replaced by: "And what else ?"]</i>	
Jido	the orange bottle is in the big blue box	
user	OK. Now, Cazn you give me the videotape ?	x
	<i>[Replaced by: "OK. Now, can you give me the videotape ?"]</i>	
Jido	ok. which videotape?	
user	The one that's near you	ok
Jido	ok.	
user	thanks	ok
Jido	you are welcome	
user	Now, please throw the paper sheets.	x
	<i>[Replaced by: "Now, please, throw the paper_sheets."]</i>	
Jido	papaer? what do you mean?	
user	sorry.. I repeat. Throw the paper_sheets please.	x
	<i>[Replaced by: "sorry.. I repeat. throw the paper_sheets please."]</i>	
Jido	ok	
Jido	I cannot reach the red paper	
user	Good	ok
Jido	good	
user	Now, throw the red glass in the small blue trash-bin	ok
Jido	the small blue trashbin? what do you mean?	
user	I mean the small blue box	ok
Jido	ok	
user	very good. Now look.	ok
Jido	Very good	
user	Throw the videotape.	x
	<i>[Replaced by: "throw the videotape."]</i>	
Jido	Which one?	
user	How many videotapes do you see ?	ok
Jido	I see one videotape	
user	Then throw it.	ok
Jido	ok	
user	Can you reach the bottles ?	ok
Jido	yes	
user	Throw one of them.	x
	<i>[Replaced by: "throw one of them."]</i>	
Jido	ok	
user	Good.	ok
Jido	good	
user	Now, it's my turn.	ok
user	I will throw one bottle.	ok
Jido	ok	
Jido	ok	

Speaker	Utterance	Input correctly parsed?
user	Now can you do anything to help me clean the table ?	ok
Jido	yes	
user	What can you do ?	ok
Jido	the orange bottle is on the table	
user	So ?	
<i>[Parser failed]</i>		
Jido	So? what do you mean?	
user	I have thrown the last videotape.	ok
Jido	ok	
user	Is there something on the table ?	ok
Jido	yes	
user	Please, be more expolicit.	ok
Jido	expolicit? what do you mean?	
user	I repeat. Please, be more explicit.	ok
Jido	explicit? what do you mean?	
user	OK. f/Forget it.	x
<i>[Replaced by: "OK. Forget it."]</i>		
Jido	ok	
user	Please, throw the last bottle.	ok
Jido	ok	
user	Are we done ?	ok
Jido	no	
Jido	the red paper is on the table	
user	You are right.	
<i>[Parser failed]</i>		
Jido	I am right	
user	I will give it to you.	ok
Jido	ok	
user	Now, What you do ?	x
<i>[Replaced by: "Now, What do you do ?"]</i>		
Jido	I don't know.	
Jido	and you?	
user	Can you clean the table ?	ok
Jido	yes	
user	Then do it.	ok
Jido	ok	
user	Perfect	ok
user	Good bye.	ok
Jido	Bye	

## 7.4 Context

This dialogue is very difficult for parsing and for interpretation so we don't try tests with it

Speaker	Utterance	Input correctly parsed?
user	hello	
Jido	hello	
user	how are you ?	
Jido	I am jido	
user	what's your family name ?	
Jido	My family name is Jido	

Speaker	Utterance	Input correctly parsed?
user	ok, so what's your firstname ?	
Jido	my firstname is Jido	
user	so you are Jido Jido right ?	
Jido	yes. I am Jido	
user	what are you doing ?	
Jido	I am seeing the Lord of Robots videotape, the Jido.e videotape, the blue bottle, the orange bottle, the green bottle, the gray bottle, the green table, the trashbin, the red box, the big blue box, the small blue box, the blue wineglass, the orange cup, the red cup, the red pen, the blue pen, the black pen, the red paper, the blue paper, the green paper, some papers and you. I am talking to user. I am listening to user. I am not playing the spygame.	
user	do you have friends ?	
Jido	no.	
user	That must be sad..	
Jido	that? do you mean friends?	
user	No, to be alone is sad	
Jido	ok	
Jido	yes.	
user	great	
Jido	great.	
user	What do you know about him ?	
Jido	Mahdi is Mahdi	
Jido	Mahdi is small	
user	what else ?	
Jido	I don't know	
Jido	and you?	
user	Is he nice ?	
Jido	I don't know if he is or not	
Jido	and you?	
user	So you do not really know him, do you ?	
Jido	no.	
user	But you told me earlier that you knew him !	
Jido	yes.	
user	ok ok	
Jido	ok ok	
user	How old are you ?	
Jido	I don't know	
Jido	and you?	
user	doesn't matter	
user	Where do you live ?	
Jido	live? is it like go?	
user	not really. Where are you ?	
Jido	I am at the LAAS	
user	Where is the LAAS ?	
Jido	at Toulouse	
user	Where is Toulouse ?	
Jido	at France	
user	ok, and where is France ?	
Jido	I don't know	
Jido	qnd you?	
user	France is in Europe	

Speaker	Utterance	Input correctly parsed?
Jido	ok	
user	Where are you working in LAAS ? In which room ?	
Jido	I don't know if I work or not	
Jido	and you?	
user	Don't know either	
user	Do you smoke ?	
Jido	ok	
Jido	I don't know if I smoke or not	
Jido	and you?	
user	I don't smoke	
Jido	ok	
user	I think you don't smoke	
Jido	ok	
user	Have you ever been happy ?	
Jido	yes.	
user	When ??	
Jido	on 2010	
user	and before ?	
Jido	before? what do you mean?	
user	before 2010, have you been happy ?	
Jido	yes.	
user	when ?	
Jido	on 2009-06-1-10:23:25	
user	why ?	
Jido	why? what do you mean?	
user	what did you do this day ?	
Jido	I was seeing the Lord of Robots videotape, the Jido.e videotape, the blue bottle, the orange bottle, the green bottle, the gray bottle, the green table, the trashbin, the red box, the big blue box, the small blue box, the blue wineglass, the orange cup, the red cup, the red pen, the blue pen, the black pen, the red paper, the blue paper, the green paper, some papers and you. I am talking to user. I am listening to user. I am not playing the spygame.	
user	Do you know that you're a robot ?	
Jido	yes.	
user	How do you live whith that ?	
Jido	live? is it like go?	
user	Do you meet humans ?	
Jido	yes.	
user	Who is the last human that you've met ?	
Jido	I met user	
user	and before ?	
Jido	I met Mahdi	
user	How do you know he is human ?	
Jido	He? do you mean Mahdi?	
user	yes	
Jido	He is a human and an agent and small	
user	how do you know that ?	
Jido	that? do you mean human?	
user	Have you ever travelled ?	
Jido	travel? is like go?	

Speaker	Utterance	Input correctly parsed?
user	Have you ever been in Morocco ?	
Jido	no	
user	You should	
Jido	should? what do you mean?	
user	It is great !	
Jido	great	
user	You should go to Morocco	
Jido	ok	
user	Give me a beer please	
Jido	a beer? I don't know what you mean	
user	It's a kind of drink for adults	
user	Do you want one ?	
Jido	ok. what are drink and adult?	
Jido	I don't know if I want or not	
user	An adult is a human who has more than 18 years	
Jido	ok	
user	Do you prefer girls or boys ?	
Jido	I don't know if I prefer or not	
user	you prefer girls	
Jido	ok	
user	Do you prefer girls or boys ?	
Jido	I prefer some girls	
user	No, you prefer boys	
Jido	No. I prefer some girls	
user	Are you sure ?	
Jido	yes	
user	Learn that you prefer boys	
Jido	ok	
user	Do you prefer boys or girls ?	
Jido	I prefer some girls or some boys	
user	Learn that you prefer girls	
Jido	ok	
user	Do you prefer girls or boys ?	
Jido	I prefer some girls and some boys	
user	well	
Jido	well	
user	What do you plan to do this we ?	
Jido	I don't know if I plan or not	
user	I have to go. Nice to meet you	
user	See you later	
Jido	Nice to meet you	
Jido	see you later	

## 7.5 Other

Speaker	Utterance	Input correctly parsed?
user	hi	ok
Jido	hello	
user	how are you jido?	x
[Replaced by: "how are you Jido?"]		
Jido	I am Jido	
user	your batteries are well charged?	ok

Speaker	Utterance	Input correctly parsed?
Jido	ok	
user	can you put the red paper in the red box	ok
Jido	I cannot reach it	
user	which paper can you reach?	ok
Jido	the green paper	
user	so put it on the blue box	ok
Jido	which blue box?	
user	the big one	ok
Jido	ok	
user	can you see some bottles on the table?	ok
Jido	yes	
user	which color are they?	ok
Jido	the blue bottle, the orange bottle, the green bottle, the gray bottle, the green table	
user	what is the position of the green bottle	ok
Jido	position? what do you mean?	
user	which cup is the most longer?	x
<i>[Replaced by: "which cup is the longest?"]</i>		
Jido	I don't know	
Jido	and you?	
user	can you see the white paper?	ok
Jido	yes	
user	there is another paper on it?	
Jido	the process cleaning the table has been activated. Will you do it?	
Jido	do you mean another paper on the white paper?	
user	yes	ok
Jido	ok	
user	can you see the blue paper?	ok
Jido	yes	
user	is it on your left?	ok
Jido	yes	
user	the red box is on your left or on you right?	ok
Jido	I don't know	
Jido	and you?	
user	the red box is on your left	ok
Jido	ok	
user	where is the red box?	ok
Jido	the red box is at my left	
user	no, the red box is on your right	ok
Jido	no. the red box is at my left	
user	do you have a tape for me?	ok
Jido	no	
user	where can I find tapes?	ok
Jido	i don't know	
Jido	and you	
user	I don't know	ok
Jido	ok	
user	do you know something about movies?	ok
user	who have the tapes?	ok
Jido	Patrick has the tapes	
user	which tape he has?	ok
Jido	LOR and Jido-e	
user	what is it about LOR?	ok

Speaker	Utterance	Input correctly parsed?
Jido	The Lord of the Robots film trilogy consists of three fantasy-adventure films: The Fellowship of the Robot, The Two Towers and The Return of the Humanoids. The trilogy is based on the three-volume book The Lord of the Robots by S. Lemaignan. Set in the fictional world of Middle-earth, the three films follow the hobbit Jido as he and a Fellowship embark on a quest to destroy the One Robot, and thus ensure the destruction of its maker, the Dark Lord Python. The Fellowship becomes divided and Jido continues the quest together with his loyal companion Hrp2 and the treacherous Gollum. Meanwhile, the wizard Mahdi and Aragorn, heir in exile to the throne of Gondor, unite and rally the Free Peoples of Middle-earth, who are ultimately victorious in the War of the Robot.	
user	who is the main actor?	ok
Jido	Jido	
user	you are an actor?	ok
Jido	gollum, jido ,the Dark Lord Python	
user	did you play in another film?	ok
Jido	yes	
user	which one?	ok
Jido	Jido-e	
user	is this movie great?	ok
Jido	I don't know if it great or not	
user	and what do you know?	ok
Jido	I know the Lord of Robots videotape, the Jido.e videotape, the blue bottle, the orange bottle, the green bottle, the gray bottle, the green table, the trashbin, the red box, the big blue box, the small blue box, the blue wineglass, the orange cup, the red cup, the red pen, the blue pen, the black pen, the red paper, the blue paper, the green paper, some papers and you.	
user	I know your name	ok
Jido	ok	
user	what is my name?	ok
Jido	user	
user	and the last name of P/Patrick	ok
Jido	I don't know	
Jido	and you?	
user	P/patoch	x
Jido	ok	
user	what is the last name of patrick?	x
	[Replaced by: "what is the last name of Patrick?"]	
Jido	I don't know	
Jido	and you?	
user	the surname of patrick is patoch	x
	[Replaced by: "the surname of Patrick is Patoch"]	
Jido	ok	
user	who is patoch?	x
	[Replaced by: "who is Patoch?"]	
Jido	patoch is patoch and patrick	

Speaker	Utterance	Input correctly parsed?
user	where is brian?	x
	<i>[Replaced by: "where is Brian?"]</i>	
Jido	I don't know	
Jido	and you?	
user	brian is in the kitchen	x
	<i>[Replaced by: "Brian is in the kitchen"]</i>	
Jido	ok	
user	where is patoch?	x
	<i>[Replaced by: "where is Patoch?"]</i>	
Jido	I don't know	
Jido	and you?	
user	patrick is in the kitchen	x
	<i>[Replaced by: "Patrick is in the kitchen"]</i>	
Jido	ok	
user	where is patoch?	x
	<i>[Replaced by: "where is Patoch?"]</i>	
Jido	patoch is in the kitchen	
user	who is in the kitchen?	ok
Jido	Patoch, Brian and Patrick	
user	are Patoch and Patrick the same person?	ok
Jido	I don't know	
Jido	and you?	
user	I have to go Jido	x
	<i>[Replaced by: "I have to go. Jido"]</i>	
Jido	ok	
Jido	the process cleaning the table has been activated. Will you do it?	
user	what will you do this afternoon?	x
	<i>[Replaced by: "what will you do in this afternoon?"]</i>	
Jido	I don't know	
Jido	and you?	
user	I don't know	ok
Jido	ok	
user	this afternoon you will see Jeremy	x
	<i>[Replaced by: "in this afternoon you will see Jeremy"]</i>	
Jido	ok	
user	bye jido	x
	<i>[Replaced by: "bye Jido"]</i>	
Jido	bye	