Abstract
We present here an oral dialogue system designed for air-traffic controller training with an air-traffic simulator. The system has been developed with two main objectives in mind: we try to make it both habitable and robust. The dialogue manager is designed to be able to cope with actual operational conditions and to allow users flexibility and freedom in the choice of their formula-14 of the task and the system which is being used is dialogue according to the universe state of the task the weights make some messages allowed or forbidden. The dialogue manager LIMSI. The actual conditions of use and allow the user some flexibility in the choice of his expressions and his mode of pronunciation. On the one hand, we took into account the limitations of the recognition device available, and set up internal error detection and correction strategies which required the cooperation of all the knowledge levels (acoustic as well as semantic-pragmatic); in case of failure, an error recovery dialogue was also provided.

1 Introduction
This paper presents a system which handles an ongoing oral dialogue between a student controller and a simulation device available, and set up internal error detection and correction strategies which required the cooperation of all the knowledge levels (acoustic as well as semantic-pragmatic); in case of failure, an error recovery dialogue was also provided. In the paper, we describe the knowledge representation which renders the cooperation of different knowledge sources possible. We give details of the correction mechanisms and also describe an approach that makes use of pragmatic knowledge to predict a sub-language which dynamically limits the domain search for recognition and therefore improves its accuracy.

2 Knowledge representation
The model proposed here is based on frame theory. The concept introduced by Minsky for vision process modeling has been used in several systems with varying interpretations [6]. A frame is a data structure which describes each object or concept with a number of named slots which hold data values or properties as well as procedures which may be executed if certain conditions exist. In the text or message interpretation phase, the instantiation consists of verifying to what extent a text or a message can be identified with this formalized description.

Concepts used in the air-traffic control (ATC) language can be classified in several categories (concerning heading, level, speed ...). A frame is associated to each category. All knowledge required for message analysis and interpretation is present in the frame. In order to define and manipulate frames, we built a frame-based language core. An example showing the data organization in a frame corresponding to the "level" category is given in Fig 1.

For each plane present in the air sector for which the controller is responsible, the system possesses an image of the plane which contains all the information concerning it. The system keeps a trace of the preceding exchanges (in an internal form), in the dialogue history.

(*niveau
 (ISA  *instruction)
 (indatif LINK *ind)
 (action VAL (maintenez descendez ...))
 (sujet VAL (niveau))
 (parametre LINK niveau ))
 (constraint (param.valeur mul 5)
 (param.valeur <= 95)
 (param.valeur >= 600))

(niveau
 (digit1 VAL (0 1 2 3 4 5 6 7 8 9))
 (digit2 VAL (0 1 2 3 4 5 6 7 8 9))
 (digit3 VAL (0 5 ))
 (valueur FUNC calcul))

Fig. 1 Example of a frame ("level" category)

3 Message analysis
The methods used for utterance analysis in dialogue are frequently the same as the ones used in written natural language processing. They utilize a language model represented by ATN which yields an internal structure of the message and extracts the meaning with the help of interpretation rules or filtering mechanisms in order to associate this structure to a dialogue frame. This approach assumes that the message uttered strictly respects syntax rules used for written text analysis, which is rarely the case in an oral interaction.

Applications for which dialogue and language are completely task-oriented allow association of a form of message with a precise action. This type of dialogue generally
4 Error detection and correction

Error detection:

For speech understanding, the recognition process includes a disturbing parameter that is caused by recognition errors. The system must then be able to detect errors made either by the recognition system or by the speaker, and, therefore, extract a meaning from an utterance even if it is incomplete (partly recognized). It must also detect errors, be capable of detecting incoherences and ambiguities, and find a remedy to minimize the number of exchanges: it means that the system must avoid completely rejecting a message whenever it is possible.

Two principles are used in message coherence checking: the limitation of domain variability and message redundancy. For example, in the message "descend right level 230", the information brought by "descend" is already included in "level 230" as well as in the present level known by the system. These constraints are integrated in a declarative form to the frame when it is defined. The control validity module examines all the frames in the network dialogue which has been built up during the dialogue, in order to check the validity of each frame by using the syntactico-pragmatic constraint rules which are associated to each frame. If a rule is not verified, the corresponding slot is marked accordingly.

Error correction

Tests were carried out on 4 speakers pronouncing a corpus of 50 sentences twice. They showed that the most frequent error was a confusion error (occurring more than half of the time - 56%) Tests allowed us to define a word confusion matrix containing, for each word, a list that was ordered according to the confusion frequency and that contained all of the words that might be confused with this word.

The system searches for a correctly instantiated slot in the erroneous frame. Using a correct word found in the message and corresponding to one of the frame slots, it tries to correct the error by checking the local syntax and the confusion matrix. The non-compatible word is replaced by a word with which it can be confused and which belongs to the list of authorized predecessors and successors of the correct word (considered as an anchor point). The instantiation is executed again on the whole message, which allows taking a consequential effect due to the correction into account.

Utterance "tournées gauche cap 2 3 0°" ("turn left heading 2 3 0°")

Recognized message "tournées gauche 4 2 3 0°" ("turn left 4 2 3 0°")

After correction the message becomes "Tournées gauche cap 2 3 0°"

**after control**

\[
g_{100} = \{ \text{ISA, *instruction} \}
\]

**after correction**

\[
g_{100} = \{ \text{ISA, *instruction} \}
\]

Fig 2. Example of correction

5 dialogue

Dialogue module

The dialogue module (Fig. 4) is the central module of the system: it has to coordinate the modules involved and, in particular, the contextual interpretation of the messages coming from the task (simulator) or the speaker. During the dialogue, the system collects knowledge which is progressively integrated in the network either by creating, by removing, by correcting or modifying elements. The structure the system has at its disposal is a network of frames, called dialogue network, which represents the current state of the dialogue; it is composed of all the information gathered by the system from the beginning of the dialogue; whenever an action is ready to be executed, which means that all the required parameters are present, it is sent to the addressee (simulator or user). The system processes all the frames in the same way whether they come from the speaker or from the simulator.

The frames provided by the analyzer (Analyzer1 for the user's messages and Analyzer2 for the task messages) are merged in the dialogue network. The different modules are then executed in the following order:

- error detection and correction,
- message generation to the simulator and the user,
- updating of the dialogue network and of the knowledge bases,
- prediction generation for the message that will follow.

Each module accesses the dialogue network to retrieve information and provide other information. When the system starts up, the dialogue network is empty.

Merging the message representation into the network constitutes an essential dialogue function; this function is responsible for checking the correspondence of the current message against the preceding ones. For each frame representing the current message, a corresponding frame is searched for in the dialogue network. If the search succeeds, the frame is merged with the corresponding one. Information already acquired is completed or updated with secondary frame information.

Message generation

Messages belong to 4 types:

1) answer to a speaker's question,
2) confirmation message after an action that was requested by the speaker has been executed by the system.
3) dialogue communication channel handling messages (repetition, contact, holding on, acknowledgement)
4) sub-typed questions: either a request for precision about the speaker's preceding message (in order to better understand this punctual message); or, after a speaker's request (a system question for complementary information to execute the requested action) which leads to a dialogue aiming at a precise goal and allowing the speaker to provide the necessary information. Questions are asked in such a way that the speaker is lead to give terse and precise answers, which renders recognition easier [8]. To improve dialogue naturalness, several formulations are associated to one message allowing the system to choose one of them in a random way.
The system examines the network and stacks the messages to be generated. The state of the frame determines which messages are to be generated. If an error has been detected and marked accordingly in the frame, the system asks a question in order to obtain all missing information from the speaker, and the corresponding frame state awaits the speaker’s response. If the frame is fulfilled the system generates a message specified in the description frame which can either be the answer to a question, an acknowledgment or repetition.

6 Prediction

Predictions essentially consist of using understanding upper levels of knowledge (syntactic, semantic and pragmatic) in order to circumscribe search space for words and improve recognition performance. The predictive characteristic of dialogue system is inversely proportional to the dialogue language flexibility and richness. Several types of predictions can be distinguished:

1) syntactic predictions: they can be used together with syntax grammars and consist either of constraints generated during analysis after a word is recognized [7] or of semantic grammars. The latter are the most frequently used in the recognition system. Grammar rules are similar to syntax grammar rules, but words are associated to semantic classes. Only sentences which respect both syntactic and semantic rules are declared valid.

2) semantic predictions: the use of correspondence among two successive utterances according to the task and the user’s models allows the system to considerably limit the recognition search space, especially for a structured task-oriented dialogue.

Speech recognition systems make frequent use of syntactic and semantic predictions but more rarely use pragmatic ones. Such an approach, to be efficient, requires complete modeling of the task universe as well as of the potential users. We propose a method aiming at predictively utilizing this knowledge with direct effect on syntax rules.

Predictions may be of 2 types: 1) prediction or exclusion of a grammar subset, 2) reinforcement or diminution of frequency probabilities of one subset, without cancelling probabilities of other message types.

Definition of pragmatic predictions

Predictions are not only dependent on the last message but more generally on the dialogue history and on the current context. They are linked to the knowledge the system has of the task universe at one instant. They can be associated to: 1) context and dialogue history, 2) the speaker, 3) a first recognition result and 4) preliminary information on the speech signal.

There are some examples of predictions associated with the controller/pilot dialogue:

1 punctual exchange: when the system asks a question to the student-controller, the universe of possibilities is limited and the system can predict the response category.

2 dialogue history: before the controller gives the pilot an instruction concerning a message category, he generally asks the present value of the parameters concerned. Therefore, if a controller’s message is a question concerning a category, the controller’s following message has a high probability of being a reply to the same category.

3) only the planes present in the air sector for which the controller is responsible, can be taken into account.

Predictions are described by rules in which all the knowledge sources co-operate. Rules are activated before each recognition process. Each rule is defined in the following way:

\[ \text{No rule } < \text{condition}> < \text{prediction} > \]

\[ < \text{condition}> : = ... \]

\[ < \text{prediction}> : = (C, P_i) \ldots (C, P_n) \]

In the occurrence probability of a concept \( C \) it must be augmented by \( P_i \). The weight \( P_i \) expresses degree of precision of the prediction and its value is proportional to the prediction force. It can also be negative, either to forbid or to reduced a concept occurrence, or to cancel the effect of a preceding prediction, the activation conditions of which are no more valid: a negative weight of the same value is applied, in that case.

Dialogue and syntax handling

Syntax is the only means that a dialogue has to constrain recognition; therefore, if the predictions must be translated into actions in the syntax which the recognition system uses. Dialogue requires an evolutive syntax according to the current context. We use a probabilistic syntax for recognition and the system modifies the rule probabilities according to context. The language is described by a binary grammar which consists of indicating, for each word (or group of words), the word (or group of words) which can follow it. A binary grammar can be generated from a tree grammar or from a corpus. The advantage of such a grammar is that the number of rules is considerably reduced (110) with regard to a tree grammar (465). The training of the rule probabilities is also rendered easier.

Prediction propagation in the syntax network

As we have already noted, all knowledge is represented in the form of a unified structure composed of frames linked by hierarchical or semantic connections. When the system starts to run, arcs and starting nodes are associated to each concept. This correspondence built by the system is based on words. Each concept is depicted by a set of words used to specify it. For example, the concept "heading" can be described by the messages "take heading 20°" or "turn right heading 30°". The list of the words which correspond to the concept "heading" is therefore: "maintain, continue, take, turn, left, right, heading, degrees ...". Knowing the words associated to a concept, the system is able to infer the arcs of the automaton corresponding to the concept. This is rendered possible due to the use of a binary syntax representation which affects only one arc to one word.

A concept prediction is propagated in a finite-state automaton on the corresponding arc with a change of the probabilities which are associated to the arcs. The probability that phrase \( W \) occurs is:

\[ p(w) = \prod_{j=1}^{j=n} p_j \sum_{w \in L(G)} p(w) = 1 \]

If the weight to be added to arc 1 probability is \( dp \), the new probabilities: \( p_{11} = p_1 + dp, ... , p_{1n} = p_n - dx \) with \( dx = dp/2 \).

Use of probability by the recognition system

The recognition system used is AMADEUS designed at LIMSI and commercialized by the BEGYS company DAVOX. It is a speaker-dependent, continuous speech recognition system, using DTW algorithm. It can process from 200 to 500 words of vocabulary in real time, due to the use of a DTW specialized chip also designed at LIMSI. The system allows recognition context (syntax and templates) to be quickly changed, which means that when it is necessary for the application, it is possible to considerably augment the authorized vocabulary size. AMADEUS uses a regular grammar depicted in the form of a table, comprising backward and forward arcs. Probabilities can be affected to forward arcs. The system was adapted in order to use these transition probabilities in the recognition process, which is essentially based on DTW[9].

The recognition process consists of finding the sequence of words which minimizes the cumulated distance \( D \) and maximizes the occurrence probability \( P \). This is equal to minimizing: \( (D - n \times \log(P)) \) [2]. For each word transition, the retained distance is therefore:
\[ \sum_w (d - n \log(p_i)) \]

- \( n \): updating coefficient,
- \( d \): distance obtained by DTW,
- \( p_i \): transition probability.

### 7 Results

The results reported in Table 1 have been obtained with 4 speaker (1 woman, 3 men), each pronouncing 200 sentences which were generated in random order but have a correct semantic meaning according to two different scripts. Grammar characteristics as well as recognition rates are given in Table 1. Recognition uses weights equal to 1 or 0 on the arcs (authorized or forbidden arcs). Tests using probabilities ranging from 0 or 1 are under progress, together with tests including syntax-erroneous sentences. The use of a binary syntax slightly decreases performance, compared to tree syntax. But this decrease is counterbalanced by the use of a dynamic binary syntax. The use of probabilities ranging from 0 to 1 should improve performance even further.

Semantic recognition shown in table 1 means correct understanding of the message, despite recognition errors.

<table>
<thead>
<tr>
<th>Rule number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree: 465</td>
</tr>
<tr>
<td>Binary: 110</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Vocabulary size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree: 110</td>
</tr>
<tr>
<td>Binary: 110</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Language size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree: 835.10^16</td>
</tr>
<tr>
<td>Binary: 174.10^14</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dynamic factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree: 9.6</td>
</tr>
<tr>
<td>Binary: 20.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sentence average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree: 13.2</td>
</tr>
<tr>
<td>Binary: 13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Acoustic recognition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree: 94.4%</td>
</tr>
<tr>
<td>Binary: 91.4%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Semantic recognition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree: 98.0%</td>
</tr>
<tr>
<td>Binary: 97.0%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Confusion Matrix for correction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree: 98.4%</td>
</tr>
<tr>
<td>Binary: 98.4%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dynamic syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree: 99.5%</td>
</tr>
<tr>
<td>Binary: 99.5%</td>
</tr>
</tbody>
</table>

![Table 1: Recognition performance](image)

### 8 Conclusion

An oral dialogue system must not be only able to understand messages but also to detect errors due to the speaker or to speech recognition, in order to correct them in an internal way and thus minimize the number of exchanges with the speaker. The system must therefore make all task-universe knowledge cooperate in the recognition process by providing predictions which will dynamically restrict the recognition search space. As opposed to natural language which is difficult to automatically handle, the language used in a specialist dialogue in the framework of a precise task requires less complex structures for message understanding, because semantic-pragmatic concepts are clearly defined, precise and non-ambiguous. However, despite the task simplicity, observation shows that people tend to be very variable in their formulations. The system presented here tries to take that observation into account and allows the user greater flexibility together with robustness.

### References


