A message processing system with object-centered semantics

Jean-François DELANNOY

Groupe Représentation et Traitement des Connaissances
Centre National de la Recherche Scientifique
31, chemin Joseph Aiguier
13402 MARSEILLE CEDEX 9 - FRANCE
e-mail : GRTC @ FRMOP11.BITNET

Abstract

This paper presents a report processing method with object-centered semantics. The syntactic analysis is performed along classical generative principles, though with a deliberately simple output as a list of index-value doublets, which the semantic module processes using methods in an object-oriented framework. The final representation is made of two types of object-centered structures : first, case-like, event level dated structures corresponding to the input clauses; second, detailed representation of the current state of an agent of the reference world, plus records for the follow-up of a task over time. Uncertainty, imprecision and prevision are handled using specialized fields. This framework is applied to the processing of daily naval reports in English.

1. Introduction

The objective of the project is the processing of messages in English reporting the daily evolution of naval situation in the Mediterranean. The information extracted is exploited for situation monitoring, maintenance of a historical database, formulation of previsions, and detection of highlights and anomalies.

A report typically gives information on the activity of one ship during the past 24 hours; for example:

> "La Belle Poule is performing oceanographical measurement in the northern Mediterranean, while moving south at a speed of 5 knots. La Belle Poule was approximately 40 nautical miles south of Marseille at 10:00, April 14."

This example exhibits discursive information on the current activity and movement of the ship and a spatio-temporal plotting relative to a reference location.

2. Representation

2.1. The taxonomy of "permanent objects"

The permanent objects (Figure 1) are relatively perennial, non-event-based entities which make up the fixed knowledge background or *reference world*. They are :
- ships: instances of known ships;
- spatial items : instances of zones and places, plus geographical directions;
- action types, down to preterminal class level (action instances are created upon parsing, as detailed hereunder). Actions subclassify into activities and movements;

Unlike action types, which are not expected to be modified, new instances of ships and locations can be added interactively to the taxonomy.

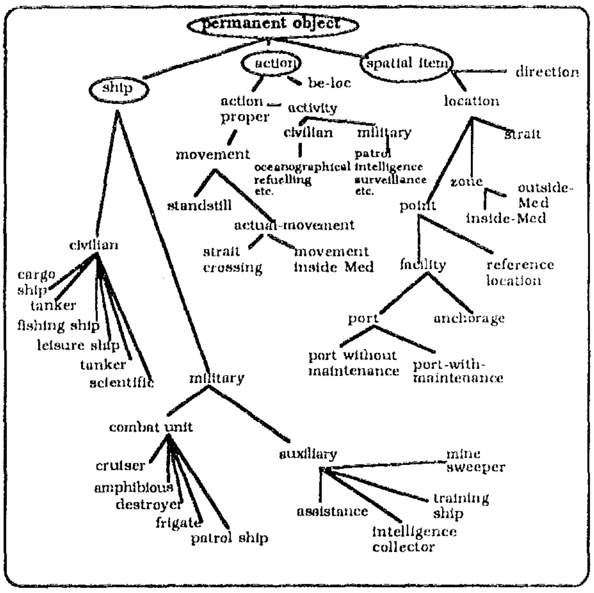

Figure 1 : The taxonomy of permanent objects

2.2. Ship frame

The ship frame indicates both the current state of a ship and indications on its activity.

The general structure of a ship frame is :
- status, a boolean (active/inactive)
- ongoing tasks
- completed tasks. This and the preceding field contain pointers to instances of the class "task", whose fields are : type (an activity), start date, intermediate (last recorded) date, end date and location.
- temporary information : current location, and, when available : goal, destination, geographical direction, companion ship, and speed;
- list of spatio-temporal plottings.

2.3. *Action frame*

Actions (event-level representation) are instances of the subclasses of the class "action". Modality values are attached :

- to the action itself :
  - temporal aspect : "previsional", "under-way", "completed".
  - degree of certainty : "observed", "certain" (meaning given as certain; this is the default value), "probable", "possible", ordered by decreasing certainty.
- to the action parameters :
  - certainty, as above
  - precision : "exact", "approximate"

2.4. *Inference procedures*

The inference procedures, implemented as methods and demons, perform the following tasks :

- check and complete new information
- manage correspondence between fields (e.g. between the "goal" and "ongoing tasks" fields), taking modality values into account
- look for better modalities (uncertainty, imprecision)
- launch previsions
- try to confirm active previsions
- signal salient and anomalous points.

Previsions can be explicit in the text (future action; goal or destination), or scenario-related.

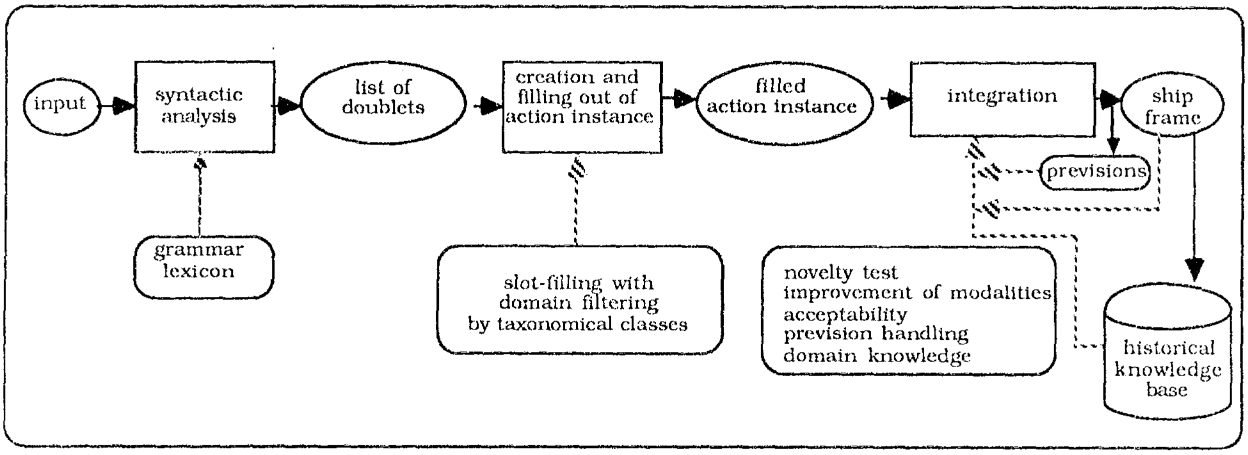

Figure 2: The processing flow

3. Syntactic analysis

3.1. *The analyzer*

Syntactic analysis is performed by an augmented DCG grammar. The output is a list of doublets <p,v>, where p is either a preposition or a syntactic category, and v the lexical-semantic translation of the item(s). The task of the analyzer is not domain-neutral and not purely grammatical :

- the "p" element of the doublets is in fact filtered and sometimes transformed : prepositions which translate identically end up the same (but inversely, ambiguities caused by plurivocal prepositions are left for the semantic processor to solve, mainly by the use of domain filtering);
- a lot of lexicon entries are complex nouns and verbal phrases

A clause is represented in the output under the form :

$<np, x\text{-}np,>.<vp, x\text{-}vp>.<x\text{-}prep_1, x_1>. \ldots .$
$<x\text{-}prep_n, x_n>.nil$

3.2. *What is expected from syntactic knowledge?*

There have been deeply contrasted positions on the role of syntax. It can be thought of as a full-fledged first stage, as an auxiliary which is sufficiently informative even when a complete syntactic structure per se (e.g. an x-barred tree) is not built (conceptual analysis : [Schank & Riesbeck 81]), as a co-process in close cooperation with semantics (since SHRDLU), as the first in an ordered sequence of increasingly costly means (as suggested by [Rau & Jacobs 87], who then list slot-filling involving filtering, heuristics when choices are to be made, and general domain knowledge).

Grossly, syntax in itself carries information on :
- constituent ordering and constituent relationship
- flexion and syntactic function (if the grammar is relational).

But more actively, syntactic analysis backed by lexical semantics, even if less sophisticated than in LFG or Mel'cuk's model for example, can play an active part in sorting items out and ascribing them an adequate translation, as for prepositions with an identical meaning. In fact, besides applying well-formedness rules, the analyzer and lexicon can do some rearrangements so as to have the job all prepared for case attribution to function nicely.

4. Semantic processing

The system first searches a clause for an action in the verb doublet or, if the verb has translated as "*empty*" (for verbs like "perform", conduct"), in the following noun phrase. The field valuation mechanism of the object environment checks that the action value does belong to the declared domain. An instance of the action type is created, and the system fills its fields with the values it finds in the second element of the doublets; the condition on the first place (preposition) is expressed as a parameter. Domain checking is again performed. A case-like structure is obtained.

When the action instance has completed the valuation of its fields, it pours itself into an existing active task of the same ship if available, or else into a new task it creates. Temporary data (current direction, destination, speed, company and goal) are replaced without testing if the new values are different. If the new action is an activity rather than a movement, it will either merge into the last recorded task if it is the same or a compatible one, or will generate a new task. If it is declared to be completed, it is (re-)written into the completed-tasks field after its status flag has been set to "inactive". If it is incompatible with an existing ongoing task, that task is closed

For all of the above, a new modality for an already valuated field will be checked against the existing modality : a "better" modality (e.g. certain vs probable) supersedes the previous one, whereas a worse one is anomalous and can be signalled. The processing of the above example thus results in the creaction/updating of the following frames :

Output lists :

(<np,La-Belle-Poule>.<vp,empty>.<oceanographical measurement>.<in,northern-Mediterranean>).
(<vp,movement>.<noprep,south>.<speed,5>).nil

<np,La-Belle-Poule>.<vp,be-loc>.<adv,approximate>.
<plotting,<40,south,Marseille,10.00,04,14>>.nil

Remarks :
- ellipsis of the subject noun phrase is recognized.
- plottings have their own structure and treatment;

Action instances :

oceanographical-measurement-13

agent	La Belle Poule
location	northern Mediterranean

movement-56

agent	La Belle Poule
direction	south
speed	5

plotting-87

ship	La-Belle-Poule
ref-loc	Marseille
distance	40 (approximate)
date	041410

Remark : instances of movements are erased after use.

Updated ship frame :

La Belle Poule

ongoing tasks	task7.nil
completed tasks	nil
zone	northern Mediterranean
direction	south
destination	Tunis (possible)
speed	5 knots
companion ship	none
plotting list	

<Marseille,11,south,approx,041215>.
<Marseille,29,south,approx,041312>.
<Marseille,40,south,approx,041410> .nil

The task is described as :

task7

type	oceanographical measurement
start date	041312
intermediate date	041410
end date	
location	northern Mediterranean

5. Related work

The principled application of structured object representation to semantic processing had its operational landmarks in Bobrow and Winograd's KRL and the systems developed by the Yale AI group (e.g. [Schank & Riesbeck 81]). [Hirst 87] proposes an overall application of the object paradigm, including to syntax, rather in the spirit of [Small & Rieger 82]'s and [Finck 89]'s word experts. [Fargues, Catach, Dugourd 86] use logic grammars, but with a semantic representation based on networks (Sowa's conceptual graphs) rather than frames.

6. Implementation

The system is implemented in Objlog ([Dugerdil 89]), a frame language based on Prolog II and featuring multiple inheritance with points of view, selective inheritance for value-sharing in relationships other than taxonomical, and dynamic facets. The grammar itself is written in Prolog II. A menu-and-mouse interface has been developed for the interrogation module.

7. Conclusion

I have tried to show that objects are a convenient and efficient way to implement semantic representation as well as analysis in a reasonably small domain. Syntax based on a moderately strict set of well-formedness conditions and some initiative in renaming is an appropriate partner for such a semantic analyzer.

Many thanks to Françoise Picard at GRTC for collaboration and advice. This research is backed by a CIFRE convention with SYSECA-Temps Réel and the Association Nationale pour la Recherche Technique (ANRT).

References

Bobrow D., Winograd T. (1977) : "An Overview of KRL, A Knowledge Representation Language", *Cognitive Science*, vol 1, n° 1, 3-46

Delannoy J.F., Picard F. (1989) : "Conception et implémentation d'une base de connaissances utilisée dans le cadre d'un système de compréhension automatique de messages en anglais", Rapport interne n° 377, Groupe Représentation et Traitement des Connaissances , Marseille

Dugerdil Ph. (1989) : *Contribution à l'étude de la représentation des connaissances fondée sur les objets*, Thèse de l'Université d'Aix-Marseille II

Fargues J., Catach L., Dugourd A. (1986) : "Conceptual Graphs for semantic and knowledge processing", *IBM Journal of Research and Development*, vol 30 no 1, January 1986, 70-79

Finck. D. (1989) : "Description d'un analyseur systémique : ANASYS", note du Centre de Recherche en Informatique de Nancy, CRIN 89-R-101

Hirst G. (1987) : *Semantic Interpretation and the Resolution of Ambiguity*, Cambridge University Press

Small S., Rieger C. (1982) : "Parsing and Comprehending with Word Experts (A Theory and Its Realization)", in Lehnert W.G., Ringle M.H. (eds.) : *Strategies for Natural Language Processing*, Lawrence Erlbaum Associates, 89-145

Rau L.F., Jacobs P.S. (1987) : "Integrating top-down and bottom-up strategies in a text processing system", *Proceedings of the 2nd ACL Conference on Applied Natural Language Processing*, Austin, Texas, 1987 (published 1988), 129-135

Schank R.C., Riesbeck C. (eds.) (1981) : *Inside Computer Understanding*, Lawrence Erlbaum Associates