CARAMEL : A flexible model for interaction between the cognitive processes underlying natural language understanding.

Gérard Sabah.

LIMSI, B.P. 133
91403 ORSAY CEDEX - FRANCE
Tel : (1) 69 85 80 03 - Fax : (1) 69 85 80 88
E-mail : SABAH@FRLIM51 (EARN)

Abstract :

In this paper we present a general natural language processing system called CARAMEL (in French : *Compréhension Automatique de Récits, Apprentissage et Modélisation des Échanges Langagiers*). Over the last few years our group has developed many programs to deal with different aspects of natural language processing. This paper describes a general architecture that integrates them in a flexible way, and provides a control strategy capable of adapting itself to the requirements of a particular task. The model is composed of three fundamental elements :
- a structured memory containing permanent knowledge and working structures of the system
- a set of processes, dedicated to the execution of the various cognitive tasks
- a supervisor, whose function is to trigger, to run coherently and to synchronize the processes.

The system contains a kind of blackboard, which is enhanced with a control mechanism driven by meta-rules. This architecture is fully implemented. We are currently developing the meta-rules necessary to use the model for various tasks.

1. Justification of the approach

Research in AI is based on the assumption that intelligence can be thought of as the **transformation of information** (Newell & Simon (63), Fodor (68)) in order to solve problems. Viewed in this light, NL processing consists in **changing representations** from a **formal language** to another.

The languages used in CARAMEL are based on conceptual graphs (Sowa 84). This general formalism has been extended in order to allow sophisticated inference mechanisms. Its use facilitates the communication between the processes.

Given the diversity of knowledge sources, we are confronted with the problem of conflict resolution and resource management. The current trend in AI is to express knowledge in a declarative form, but the independence of the knowledge sources causes artificial ambiguity. In order to be efficient, NL systems may need to integrate all the knowledge sources[1], but organizing the knowledge with this objective in mind is a rather difficult enterprise. The essential problem is the inadequacy of the control mechanisms in classical systems, no matter whether their architecture is serial, hierarchical or heterarchical. Erman et Lesser (80) suggest that a solution is to base complex systems on a *blackboard architecture*. This allows for distributed control, for autonomous and asynchronous use of the knowledge sources. The implementation of increasingly elaborate systems leads AI to *multi-experts systems*, that allow for complex communications and for flexible use of meta-knowledge (Haton 89). Nevertheless, one important problem remains : once a process is started no external control can intervene. Hence, control remains *intermittent*.

Our architecture integrates these ideas and extends the blackboard model by including a continuous control system, allowing a sophisticated management of the processes and of their interruption. This is made possible by a parallel implementation.

The architecture of CARAMEL

Our model takes into account the fact that the various processes cannot be pre-ordered (their order depends on the global task and the particular data to be processed) and that their management has to be based on a **planning** process. In CARAMEL the sequence of processes to be applied in a given situation in order to realize a given task is planned by a **supervisor**. Another important role of this supervisor is to update the working memory : as it is the only one to have a global view, it is entitled to decide what representations should be kept and be updated in a global working memory. Finally, it builds a general representation of the reasoning process ; thus it can explain why a given process has been triggered.

1. For an application of this principle in a parser, see (Sabah & Rady 83).

The system may be used in such different applications as : user-friendly interfaces, on-line help in text processors, summaries of texts, intelligent tutoring systems, etc.

2. The CARAMEL model

2.1 The memory

As we have seen, natural language processing can be conceived of as a change of representation. To handle this task, we makes use of three kinds of memory :

- a *short term memory*, which receives the results of the perceptual processes.
- a *working memory*, which contains all the structures (eventually provisional) built by the different processes. The text is represented at the various levels differently according to the point of view. Even though the working memory functions like a blackboard, it is under the control of the master process.
- a *long term memory* which contains all the knowledge of the system. This memory contains for example information about morphology, words, grammar, syntax, semantics and pragmatics, and it is a stable representation of the current state of the world. All this knowledge is permanent, declarative and expressed in terms of conceptual graphs (Sowa 84), as this formalism facilitates the communication between processes[2].

2.2 Processes

Let us briefly mention some of the processes used in CARAMEL : a deterministic parser (Rady 83, Sabah et Rady 83, Francopoulo 88) based on case grammar (Fillmore 68) and systemic grammar (Halliday 73), an ellipsis (Sauvage 88) and an anaphora-solver, an error handler (Fournier 88), story interpretation processes (Sabah 78), (Berthelin 80), (Grau 84), a planner used in dialogue handling (Vilnat 86) [the work on dialogue draws on work from linguistics (Roulet 86) and philosophy (Searle 69, Grice 75)], and a sentence generator (Zock 90).

All these processes are triggered by a master process, the **supervisor**. Before we explain its functioning, let us see how the processes are represented in the system.

2. At present the implemented data corresponds to a French lexicon of 15 000 entries (about 350 000 conjugated forms), a semantic net of a thousand concepts, a grammar (350 rules) allowing the analysis of complex sentences with prepositional and relative phrases (in French), and pragmatic knowledge about the world (at present, the system knows only a few frames in order to test the validity of the processes).

Representation of the processes

In order to find the most adequate process in a given situation, the supervisor uses a representation of each process specifying its inputs and outputs. For example :

```
(Sentence analysis (INPUT   (Type      list)
                            (Elements  words))
                   (OUTPUT  (Type      cpt. graphs)
                            (Element   syntactic rpr)
                            (Element   semantic rpr)))
```

CARAMEL distinguishes between the following two types of process :

- **Elementary processes** (such as ellipse or anaphorae resolution), whose task is to complete existing representations. In case of problem, they will call the supervisor.
- **Compound processes** (parsing, story interpretation, generation...), which are composed of elementary or compound processes. They build new representations. In their turn, they can be considered sub-supervisors (based on an internal planning process, they decide what actions have to be performed and they know how to represent what has been done).

3. The supervisor

The supervisor triggers the adequate processes in order to build the necessary representations, it handles the various problems that can appear, and it represents the actions that have to be performed. Moreover, the supervisor builds a dynamic representation of the sequence of processes activated to solve the global task. This representation is built in the working memory and allows the system to give explanations about the strategy used. As each complex process acts as a supervisor with regards to its subprocesses, the system has a recursive structure.

The supervisor analyses the representations stored in the working memory and deduces the processes that may be triggered. It also takes into account the needs of the active processes and the global task : it has to handle interruptions coming from them. Thus, the system integrates bottom-up and top-down control.

The basis of this reasoning is a **planning** process. If, for example, the task of the system is to understand a story, it knows *a priori* that it has to built a global representation of it. First, a static planner builds the sequence of complex processes that can build this type of representation (*parser + story understanding*). The same type of reasoning produces sub-plans for the complex processes involved. For example, this means in the case of story understanding : *action, character and pragmatic interpretations, followed by a synthesis*)

This first phase is independent of the data to be processed, and, as some aspects are not yet well defined, several processes may be optional (pronoun resolution, error correction...). In a second phase, a dynamic planner allows the system to adapt the static plan to the specific data present in the working memories. Whenever a problem arises, the dynamic planning process takes care of it, by determining (based on the inputs and outputs of the available processes) the kind of process capable to solve it. This mechanism is similar to the "hierarchical planning" proposed by Stefik (81), or Wilkins (84). We use a similar kind of *meta-planning* : if there are several solutions possible, they are ordered by the supervisor such as to increase efficiency.

The existence of these two planning phases allows that a process selected in the static planning phase can break down. When this occurs, a help request is sent to the supervisor which triggers a dynamic planning phase. When the processes selected in this second phase solve the problem, the original process is resumed[3].

4. Conclusion

In this paper, we have presented a general architecture capable of adapting its control strategy to the requirements of different tasks. It permits a modular implementation, it makes the modification of existing processes and the integration of new capacities in the system easier.

Another characteristic of our system (not developed here) is that it is capable to explain its actions. This has proven quite helpful when putting the final touch to the rule-base.

This capacity has interesting consequences on the learning process : as the system remembers the actions it has performed in a given situation, it is able - in the near future - to find analogies between the memorized situations and the actual one. In consequence it can bypass the supervisor when reasoning. CARAMEL is thus a flexible system, capable to manage a great number of different tasks, with an adaptative architecture and an optimal use of processes.

3. Parallelism offers another solution to adapt the static plan to the data. Its implementation, currently under development, will permit a **continuous control** on the processes : the supervisor will be able to examine the representations that are being constructed and, possibly interrupt a given process in order to wait for the result of another one or to give it some advice.

Bibliography

Berthelin Jean-Baptiste 1980 Story characters constructed by program, Coling, Tokyo.

Boguraev Branimir et Karen Spark-Jones 1987 A note on a study of cases, Computational linguistics, 13 1-2, 65-68.

Davis Randall, Bruce Buchanan et Edward Shorliffe 1977 Production rules as a representation for a knowledge-based consultation program, Artifical intelligence, 8 (1), pp. 15-45.

Erman L.D., V.R. Lesser 1980 The HEARSAY-II speech understanding system : a tutorial, in Trends in speech recognition, Lea, Englewood Cliffs Prentice Hall 361-381.

Fillmore Charles 1968 The case for case, in Universals in linguistic theory, Bach & Harms, Chicago, Holt, Rinehart and Winston, pp. 1-90.

Fodor Jerry 1968 The appeal to tacit knowledge in psychological explanations, *Journal of philosophy*, 20, p.632.

Fournier Jean-Pierre 1988 Traitement des erreurs dans la communication Homme-Machine en LN, Actes GRECO-PRC communication Homme-Machine, Paris.

Francopoulo Gil 1988 *Analyse du français avec apprentissage inductif de la syntaxe*, thèse univ. P&M Curie.

Grau Brigitte 1984 Stalking coherence in the topical jungle, 5th ICGCS, Tokyo.

Grice H.P. 1975 Logic and conversation, in *Syntax and semantics 3 : speech acts*, Coles & Morgan, Academic press, New York, pp. 41-58.

Halliday Michael 1973 Explorations in the functions of language , Arnold, Londres.

Haton Jean-Paul 1989 *Panorama des systèmes multi-agents*, Haton (ed), Architectures avancées pour l'IA.

Minsky Marvin 1974 *A framework for representing knowledge*, Memo 306, MIT, Cambridge Mass.

Newell Allen et Herbert Simon 1963 GPS, a program that simulates human thought. In *Computers and thought*, Feigenbaum & Feldman, McGraw Hill, NY, 279-293.

Rady Mohamed 1983 *L'ambiguïté du langage naturel est-elle la source du non-déterminisme des procédures de traitement ?*, Thèse d'état, Univ. P&M Curie.

Roulet 1986 Stratégies interactives et interprétatives en français contemporain, Cahier de Linguistique française n°7, Université de Genève.

Sabah Gérard 1978 Contribution à la compréhension effective d'un récit, Thèse d'état, Univ. P & M Curie.

Sabah Gérard, Mohamed Rady 1983 A deterministic syntactic-semantic parser, 8° IJCAI, Karlsruhe.

Sauvage Caroline 1988 Gestion des ellipses dans un système de compréhension du français, note LIMSI 88-7.

Searle John 1969 Speech acts, Cambridge.

Sowa John 1984 *Conceptual structures : information processing in mind and machine*, Addison Wesley, Reading, MA.

Stefik M. 1981 Planning with constraints, *Artificial Intelligence*, 16, pp. 111-170.

Vilnat Anne 1986 Relevant responses in man-machine conversation, Structure of multimodal dialogues, Venaco.

Wilkins David 1984 Domain-independant planning : representation and plan generation, *A.I.*, 22, 269-301.

Zock Michael 1990 "See What I Mean ?" Interactive sentence generation as a way of visualizing the meaning form relationship, WCCE-90, 5th World Conference on Computers in Education, Sidney.