CECOIA III, 3ème Conférence Internationale Economie et Intelligence Artificielle,
Tokyo, Japan, 31 août - 4 sept. 92, publication JASMIN/IFORS 92, pp. 415-418.

Complex Situations Understanding Support: a Multi-Agent Architecture

Marcel EGEA, Bernard ESPINASSE, Michel VIGUIER

GRASCE UA CNRS 935, Université Aix-Marseille III,
Centre Forbin, 23 Cours Gambetta,
13627 Aix-en-Provence, Cedex 1, France, Tel: (33) 42 96 14 96. Fax : (33) 42 23 39 28

ABSTRACT

Decision support, and particularly support for strategy elaboration, requires some understanding of complex and ill-structured situations, with incomplete information and subject to impredictable evolutions. Facing this complexity, the current Decision Support Systems are often insufficient for two main reasons. The first is that no conceptual frame exists that would permit to apprehend cognitive processes in the understanding of complex situations. The second reason is that the tools used in these systems (operational research, classic artificial intelligence . . .) do not fit the requirement of the simulation of such cognitive processes. The first part of this paper presents a conceptual framework to design understanding support of complex situations. This conceptual framework is based on different research on cognitive processes of understanding and more particularly on the link between planning and understanding in problem solving. Then we propose a modular software architecture that permits the application of this conceptual framework. These modules use distributed artificial intelligence techniques with reactive and cognitive specialised agents which simulate with co-operation cognitive processes. Finally, we present an example to illustrate the dynamic elaboration of a strategy based on the understanding of situations with a limited implementation of the software architecture proposed.

Key Words: DSS, understanding, planning, cognitive processes, strategy, distributed artificial intelligence, cognitive agent, reactive agent.

1 - INTRODUCTION

Decision Support Systems (DSS) do not have to focus only on problem solving, but they have to concern problem finding and situation understanding [Landry & all 85] [Espinasse 92]. The support to the strategies elaboration needs complex situations understanding, which are characterised by incomplete and unstructured informations. These informations are often in continuous evolution. In front of this complexity, current DSS are devoid of interest for two reasons. The first reason is that we do not have a conceptual framework sufficiently formalised that permits to apprehend cognitive processes of complex situations understanding The second reason is that tools currently used in actual DSS (Operational Research, classical artificial intelligence, . . .) are limited in the simulation of such cognitive processes.

2 - CONCEPTUAL FRAMEWORK

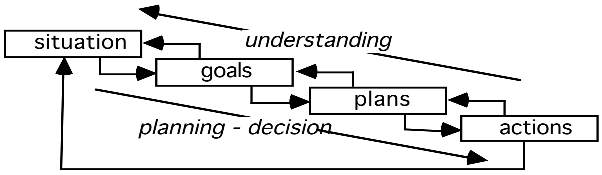
We consider the understanding as an activity of building of coherent representation composed of elements. Most of the times, this building is finalised and this finality corresponds to what the decider wishes to do with the representation he builds. The coherence of this representation is obtained by integration of all relevant elements with the finality in a same structure. [Hoc 84].

The understanding activities have been largely studied in the field of language understanding [Schank & Abelson 77]. These studies stress on the central place of planning in the understanding. The relevance of these studies surpass the only language understanding field. In history understanding, the reader owns procedural knowledges (what the history actors do) and has to infer declarative informations (the goals pursued and the action justifications). In problem solving, it is the reverse, the subject owns the goals and has to find actions to achieve it.

The study of understanding processes leads to develop two main types of complementary models; ascendant models and descendant models [Hoc 84]. Ascendant models focus on constructive mechanisms of ascendant nature, which infer a structure from a statement often verbal. An illustration of these models is the UNDERSTAND model [Hayes & Simon 74], whose object is to report the problem statements understanding for their solving. Descendant models focus on the schema evocations and their particularisations with the problem data. The schema concept is associated to the presence in memory of high level units, which can economically code information configurations and their structuration.

As previously seen, understanding can be assimilated to an elaboration of a representation of which the coherence is obtained by integration. This necessary integration leads us to emphasise the important link between understanding and planning. R.Wilenski [Wilenski 83] shows that from the planning point of view, understanding and problem solving use the same general mechanisms.

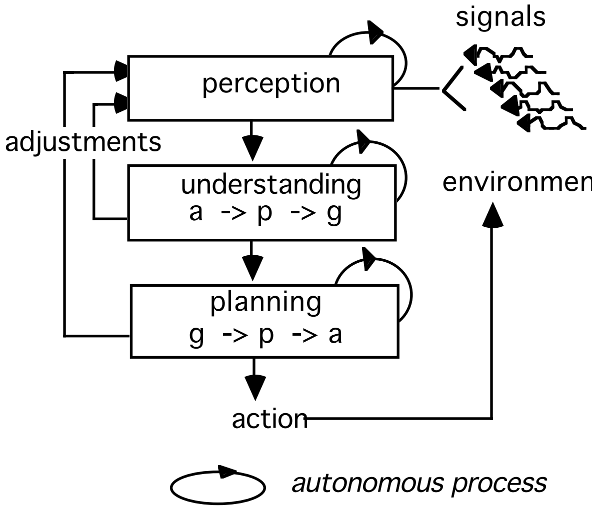
Wilenski's work seems us very interesting and useful in our research of conceptual framework to design support in complex situation understanding. The main contribution of Wilenski is to have shown that planning and understanding are two different aspects from a same process led in the reverse way. From a situation and a goal, the planning consists in establishing action plans, then executing actions, which will modify the situation. From the observation of a situation resulting of one or more actions, the understanding consists in infering what is the plan of which these actions are issued and to what goal this plan is linked. The following schema illustrates this two processes :



The cognitive processes of understanding and of planning use knowledge related to plans. Plans are linked to one or more goals. We distinguish two kind of plans; plans that permit to explain and plans that permit to decide actions. Plans which are related to particular goals can be found in memory through a specification of this goal. In planning, we can consider interaction between goals, evaluate alternative plans to abandon some goals and reinforce other goals.

In strategies elaboration, the decider has to understand (explain) his environment, for instance the behaviour of one or more actors. This consists in discovering their goals and plans, from observations of effects of actions done by these actors (signals). Then, on this understanding, the decider can define again his goals and determinate the action plan he has to execute.

The strategy elaboration is composed of three phases : the perception, the understanding and the planning. The perception phase consists in observing signals of the environment and to extract of them some pertinent semantic features in relation with a defined finality. The understanding phase consists in refining and/or modifying the goals chosen according to a sensory feed back on real effects of the actions, or on the events coming from the environment : action -> plan -> goal. The planning phase consists in choosing goals and in organising dynamically their priorities, to determinate by what actions and in what order one or more goals can be reached : goal -> plan -> action.The following schema illustrates this three phases:



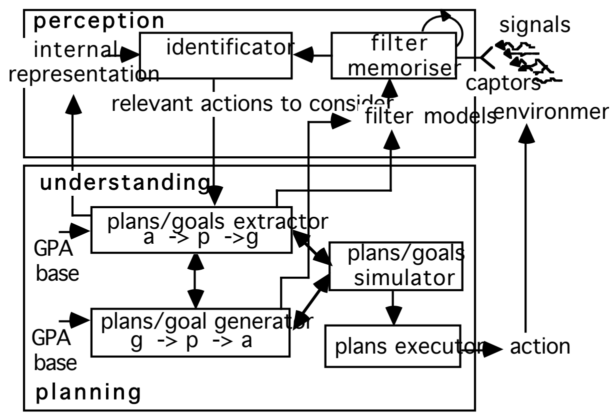
The cognitive processes of understanding and planning lead to adjustments of the environment perception process (adjustment loops). These adjustments are based on knowledge associated to the plans. Note that these three phases are relatively autonomous and executed in parallel.

3 - A MULTI-AGENTS ARCHITECTURE

We propose a modular software architecture, which allows the implementation of the conceptual framework previously presented. The realisation of the different modules of this architecture needs to use distributed artificial intelligence technics (DAI). These techniques are necessary because as the plans' structure is relatively complex, the schemas associated are also complex, compared to schemas largely used in problem solving with classical artificial intelligence techniques.

The research in DAI mainly concerns the design of multi-agents systems [Ferber 89] and more precisely the co-ordination between autonomous and intelligent agents : how can they co-ordinate their knowledge, their goals, their competencies and plans to act and solving problems sometimes multiple and independent (MAPS: Multi-Agent Problem Solving System [Garbay 89] . . .). Agents are called cognitive where they are able to realise complex operations (every agent is assimilable to an expert system more or less sophisticated). Agents called reactive are less intelligent, but are often more numerous. Through reaction mechanisms to the events, they are able to permit the emergence of behaviours associated to the goal reached. A set of such agents can be organised for example in neural network (connexionist models) to solve tasks of perception as pattern recognition or learning (sub-cognition).

The previous conceptual framework leads us to propose a modular multi-agent architecture [Ferber & Ghallab 88], which permits to simulate the perception-understanding-planning activities. This architecture is composed of three modules composed of one or many agents (cognitive or reactive) specialised for a specific task. The following schema illustrates this architecture :



The PERCEPTOR module is composed of two agent sets, mainly of reactive nature : FILTOR-MEMORIZING agents and IDENTIFICATOR agents. FILTOR-MEMORIZING agents carry out the interface with the environment (to catch signals from the environment, filter and memorise them). These agents own a base of filter models and they constitute a base of signals' chronics. From this base and with an internal representation model, the IDENTIFICATOR agents are able to discover relevant actions to consider.

The PLANNING module is composed of three agent sets mainly of cognitive nature : the GENERATOR agents, the SIMULATOR-VALUER agents and the EXECUTOR agents. The GENERATOR agents define the sets of action plans to carry out to reach the defined goals. These sets of plans are memorised in a GPA base (Goals-Plans-Actions). When action plans are selected and generated, the SIMULATOR_VALUER agents simulate and value them. Finally, the EXECUTOR agents have to supervise the execution of actions' sequences associated to selected plans.

The UNDERSTANDING module is composed of two agent sets mainly of cognitive nature : the EXTRACTOR agents and the SIMULATOR-VALUER agents. From the actions discovered by the IDENTIFICATOR agents, the EXTRACTOR agents do the understanding process as follows :

- taking into account the actions,
- if these actions can be related to a set of plans known in the GPA base, these actions can be explained and the reached goals, which are associated can be discovered through plans,
- other, possible explanations to these actions can be elaborated through the generation by the GENERATOR agents of hypothetic plans from the GPA base,

The simulator-valuer agents, common to the planning module, permit the valuation of these hypothetic plans.

The three modules have a relative autonomy and are in intense co-operative interaction. From the discovery of plans and goals of the environment actors, the understanding module permits to determine if and how these plans and goals interfere with plans and goals of the decider. The planning module has to react (reactive planning) to adapt the behaviour to the environment (goals conflicts) and if necessary, it has to be able to modify, to change goals and plans. The recognition of plans and goals of the environment actors by the understanding module leads the perception module to adjust the filter models used to observe the environment signals.

4 - AN EXEMPLE

This example is a simplification of a general model. Our system is constituted of a finite set of actors a,b,c,... Each actor a has to realise a finite family of goals O(a). Each goal can be hierarchically decomposed in subgoals.

To each goal $\theta \in O(a)$, is associated a finite set of plans $P(a,\theta)$, a plan $p \in P(a,\theta)$ being a finite list of actions $p = (\alpha_1,..., \alpha_n)$. Each action uses a finite set of resources $R(\alpha)$. Each actor knows his environment (the other actors) throught the use of resources required for the realisation of actions.

The assumptions on the building of our model are :

A1) Resource's rarity assumption : Each resource r is single and cannot be used simultaneously by several actions (risk of conflicts between actors)

A2) Non injectivity assumption : Two different actions α and β may need the same resource's family : $\exists\alpha,\exists \beta / \alpha \neq \beta$ et $R(\alpha)=R(\beta)$.

A3) Previous knowledge assumption : Each actor has his knowledge base on goals, plans and actions (GPA base), noted C(a).

The different actors use a blackboard T to get informations on their environment. The blackboard T shows, at every time t, the state T(r) of a resource r (free or used by an actor).

To each actor is associated the previous multi-agents architecture.

Perception phase

To make the perception task, the filter-memorising agent of actor a, updates a historic H(a) (lists of resources used by the other actors) by acceding to the blackboard. The identificator agent exploits H(a), but H(a) is not sufficient to completely identify an action executed by an other actor (assumption A2).

Understanding phase

The understanding of actions, the goal's identification of competitive actors increase the knowledge in the GPA base. The extractor agent will use, for example, the idea of "proximity goal". The hypothetical plans generated by this agent would be valued by the simulator-valuer agent.

Planning phase

The planning is carry out by the generator agent. This agent can use several strategies. By example, if an actor a tries to achieve a goal o with a plan p, and if the next action α of this plan must use a list of resources L, then three strategies can be considered :

Strategy 1 : Take all the available resources L(i) required for this action α . If there is a conflict on a resource, this agent waits for some times δ, then requests again the resource and if it is not available, chooses a new strategy .

Strategy 2 : Seek an other plan $p' \neq p$ for the same current goal o, as the next action α' of plan p' uses the minimum of resources.

Strategy 3 : Seek an other current goal o', trying to "understand" what are the goals of co-operative actors. So, the actor a defines a set of actors :

- the actor a builds the set $\Omega(a)$ of goals, requiring a minimum of missing resources,
- the subset $\Omega1(a)$ of $\Omega(a)$ includes those goals requiring a minimum of remaining actions for their plans,
- the actor builds the subset $\Omega2(a)$ of goals disturbed by a minimum number of competitive actors $A=(a_{j1},...,a_{jq})$,
- the actor a determines, for each actor $b \in A$, the possible goals $\Gamma(a,b)$ known by a and that b seems to follow up, according to the historic H(b). We suppose that, in this example, the goal's family O is provided with a distance d. The historic H(b) produces subgoals of possible goals. For the actor a, the most probable subgoals of actor b are those, withe a minimum distance to C(a),
- the actor a will select a goal of $\Omega2(a)$ the less "disturbed" by the competitive actors A, i.e.,, that it exists a proximity V of o, for the metric d on the goals that verify : $\forall\, b \in A \Rightarrow \Gamma(a,b) \cap V = \emptyset$. If such a goal does not exist, the actor b will use one of the previous strategies.

This example is implemented in C++ under Unix. Each actor is defined as an object class.

5 - CONCLUSION

The support to the strategies elaboration requires the inderstanding of complex situations characterised by incomplete and unstructured informations often in continuous evolution. Facing this complexity, the current Decision Support Systems are insufficient . To design new DSS that support the decider in complex situations understanding, we have proposed a conceptual framework, based on different research on cognitive processes of understanding. The particularity of this framework is to largely use the link between planning and understanding in problem solving. The software architecture that we have proposed to implement this framework, has to be validated. This architecture is too limited in the simulation of cognitive processes related to situation understanding and particulary in strategy elaboration. Each module defined in it has to be refined through new examples more and more complex in the dynamic elaboration of a strategy based on the understanding of situations.

6 - REFERENCES

ESPINASSE, B. (1992), A Constructivist Model for Decision Support : COGITA Project, a Formulation Assistant, Decision Support Systems - IFORS-SPC1, 26-29 march, Bruges, to appear in *DSS: the International Journal*, 1992.

FERBER, J. (1989), Objets et agents : une étude des structures de représentation et de communication en intelligence artificielle, *Thèse d'Etat*, Université Paris VI, 1989.

FERBER, J., GHALLAB, M. (1988), Problématique des univers multi-agents intelligents, *Actes des journées nationales du PRC-GRECO Intelligence Artificielle*, Teknea Ed., 1988.

SCHANK, R.C., ABELSON, R. (1977), *Scripts, Plans, Goals and Understanding,* Hillsdale, N.J., Erlbaum, 1977.

GARBAY, C., PESTY, S. (1989), MAPS : un système multi-agents pour la résolution de problèmes", *in 7°congrés Reconnaissance des Formes et Intelligence Artificielle*, 29 nov. 89, Paris, AFCET Publication, 1989.

HAYES, J.R., SIMON, H.A. (1974), Understanding Written Problem Instructions, in L.W.Greeg (ed) - *Knowledge and Cognition*, Potomac, Ma. Erlbaum, 1974, pp 167-200.

HOC, J.R. (1987), *Psychologie cognitive de la planification,* Presses Universitaires de Grenoble, 1987.

LANDRY,M., PASCOT, D., BRIOLAT, D. (1985). Can DSS Envolve without Changing Our View of the Concept of Problem, *North Holland, Decision Support Systems 1* , pp.25-36.

WILENSKY R. (1983), *Planning and Understanding : A computational approach to human reasoning,* Addison & Wesley 1983.