17

"TO SAY AND TO DO" VIRTUAL ACTIONS IN THE STRUCTURE AND RECOGNITION OF DISCOURSE PLANS WITH REGARD TO PRACTICAL PLANS

Cristiano Castelfranchi+ and Rino Falcone+*

+National Research Council - Institute of Psychology
Finalized Project "Information Systems and Parallel Computation"
* University of Siena - Multi-Media Lab
CRIS@PSCS2.IRMKANT.RM.CNR.IT or FALCONE@VAXIAC.IAC.RM.CNR.IT

KEYWORDS: Plan Recognition, Computer Supported Cooperative Work (CSCW).

ABSTRACT Cooperative activity requires both communication and execution of practical actions. Cooperative Works cannot be supported by obliging users to explicitly describe what they are doing and without understanding the relationships between what they discuss about and what they are doing. Furthermore, users cannot be oblige to send explicit messages to understand each other. We claim that in CSCW the system's capability of understanding actions, in particular of recognising plans and intentions, will play a fundamental role. We focus on the capability of recognising the correspondence between the plans agreed upon by the agents (Virtual Plans) and the ones they perform. This problem is particularly significant for the Action Workflow Approach.

1. INTRODUCTION

Recently, critical judgements about the CSCW systems based on the analysis of conversations and Speech Act Theory have been growing more and more; in particular, attention has been focused on limits such as unnaturalness and coercion [Suc94, Bog91] made their appearance. These limits can be brought back to different reasons (for instance Suchman herself ascribes them to the inadequacy of Speech Act Theory). In our opinion, the inner artificiality of these systems should not be ascribed to its background theory, but to the fact that the user is obliged to explicit each time (and, as a consequence, to classify) every act the user himself intends to do. This obligation is very unnatural, and the same holds for the obligation to declare explicitly the accomplishment of the task and to erase the commitment from the "agenda". This obligation derives from the necessity to match declarations with activities.
In fact, cooperative activity requires both communication and execution of practical actions. But there is a special relation between conversations and actions: they should be coherent, they should match with each other. You cannot support cooperative work by obliging users to explicitly describe what they are doing and without understanding the relationships between what they discuss about and what they are doing. You cannot support cooperative work by obliging users to send explicit messages to understand each other.
To cooperate in natural conditions, humans use not only communication (sending messages), but also their implicit domain knowledge and the interpretation of the behaviour of the partners. They are able to understand each other without explicit communication.
How to cope with this problem of coercive explicitation in CSCW systems ? In our view, a quite hard way, that cannot be easily avoided, consists in the gradual introduction of comprehension capacities into the system.
In general, we can say that the HM interaction systems are necessarily inclined to force human interaction and cooperative work in the machines' own schemes. On one side, human beings are coordinated in their actions by means of two instruments which are equally fundamental: message sending and the observation of others' actions. On the other side, machines just interact by means of the adequate message sending.
Therefore the problem lies in of allowing the user of the HC system not to submit to this machine characteristic: We must find ways and forms to endow machines with the capability of receiving and understanding the information about what has been done in an interaction.
The development of multimedia technologies will contribute to the solution of this problem especially

K. Nordby et al. (eds.), *Human-Computer Interaction*

in the synchronous mode. Nevertheless the increase of the capability of the system not only to transfer but also to understand the actions which are taking place will be fundamental. The capability of recognizing plans, intentions and actions will play a fundamental role.
In this work we try to focus on a particular aspect of plan recognition (PR): the capability of recognizing the correspondence between the plans agreed upon by the agents and the ones they perform. We introduce the notion of Virtual Action and Virtual Plan, and give their representation in a Plan Recognition system called CHAPLIN (CHart Applied to PLan INference). We analyse the problem of matching virtual plans with executed plans. This aspect is particularly significant in the Action Workflow Approach [MWF92] to have a real agreement and a real closure of the cooperation loop.

2. FROM SAYING TO DOING IN ACTION WORKFLOW

In the Action Workflow of Winograd and others [MWF92] they tried to catch the essential cell in the cooperative process of the main loop (Fig. 1): Proposal, Agreement, Performance, Satisfaction. This structure turns out to be rather hybrid, that is to say it tries to uniformly represent both communicative (mostly linguistic) and practical acts (that is execution acts: performances). Therefore, if on the one hand we have the great advantage to represent the working situation in one single cycle, on the other hand this uniformity introduces some non homogeneous elements that must be analyzed.
In particular, from the practical point of view, this uniformity of the working cycle is in charge of the user. It is true that the system keeps the relationships between commitments and original conversations in its memory, but there are two important limits to be considered:
a) The fact that the commitment is not understood by the system in the conversation without an explicit and official act by the user.
b) The system's incapability to catch the happened enclosure of the loop in the working activity (i.e. that the taken commitment has been satisfied by the performer) unless following a user's declaration. Thus the performance recognition is actually reduced to a performance declaration. Moreover, the satisfaction itself is based on the declared performance.
Now then, considering that the work in many organizations is increasingly carried out by the computer's support and that the support systems for cooperative work are more meaningful and useful if they are introduced into an information technology working environment, it seems reasonable to predict that the performance itself will take place by means of the computer or it is controlled by it.
We claim the support system must be able to recognize the completion of the operative cycle and in general to understand interactions. This implies that it is necessary to introduce "intelligence" in CSCW systems. In other words, AI and DAI techniques are needed. This idea was opposed by the forerunners of the CSCW discipline, especially by the conversational approach (Winograd and Flores)[Win87].
On the contrary, the problem discussed is especially relevant to the Action Workflow approach. The problem concerns the recognition of both actions and executive plans and their match with the above mentioned plans (requested, negotiated, promised and so on).

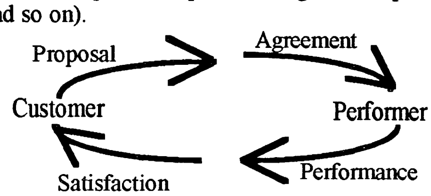


Figure 1

3. CONVERSATIONS AND PLANS

In Multi-Agent Plans (MAPs) recognition [FCa94] and, more in general, in the recognition of the interactions among agents it is very interesting the analysis of what we could call the problem of the relationship between "to say" and "to do", that is the relationship between the actions as subjects of communication either in or for the plan and their realization. We can say that this problem arises in two typical cases of interaction among agents:
1) Conversations about plans and the realization of the plans themselves:
it is the case where a discussion among the agents having the realization of practical plans as a subject (a typical case is negotiation) comes before the realization of the plans by one or more of the agents themselves.
In general, cooperative interactions are necessarily formed of conversations about the things to be done (who does what, and so on) ("conversation for action") and of the execution of the agreed plan [Win87]. To recognize this global exchange plan properly (the interaction cycle), we have to pace the problem concerning the recognition of the connection between "the actions we talk about" and "the actions done" in the plan itself.
This problem does not coincide with that of the relationship between the conversation structure or speech plan concerning a practical task and the task's structure (plan) [Gro74].
2) Conversations within the plans:
It is the case where some local conversation plans are included within the plan itself and they are required by the global execution of the plan (some of them will be described hereafter). It is clear that these local conversation plans could be like the ones described in 1); actually we mean in this case any

single communication actions whose result could be easily foreseen as it should be in a predefined plan.
In the analysis of the transformation from SAP (Single Agent Plan) into MAP [CFa94] we showed the necessity to introduce communication actions required by the correct cooperation and synchronization of the agents. The most part of such communications mentions other actions of the plan (such as in the case of the sentence "give me the bistoury" in the surgical-operation plan).
Thereafter we will just analyze the way we relate the performance of a directive or commissive action (actions asking the person who talks or who listens to do something) to the action itself. Nevertheless we could also consider assertive linguistic acts describing an action of the plan (ex: "Luigi ended up with protocoling").

4. VIRTUAL ACTION IN CHAPLIN

In the attempt to extend CHAPLIN (a chart-based plan recognition system) [FCa93,Vil90] from SAPs to MAPs we met the Virtual Actions' problem.
The formal description of a plan's body is represented in CHAPLIN by the general rule:
$P \text{-->} A_1 \ldots A_N$.
If we consider the case of a single agent (X) and other various parameters ($y_1 \ldots y_L$) of the plan's actions, we will have:
$P(X;\sigma_0)\text{-->}A_1(X;\sigma_1)\ldots A_N(X;\sigma_N)$
where: $\sigma_i \subseteq \{y_1 \ldots y_L\}$ with $0 \leq i \leq N$;
Multiagent plans are plans where the protagonists of the component actions are more than just one ($X_1,\ldots,X_M$) then we could write:
$P(X_1,\ldots,X_M;\sigma_0)\text{-->}A_1(\xi_1;\sigma_1)\ldots A_N(\xi_N;\sigma_N)$
where: $\xi_j \subseteq \{X_1,\ldots,X_M\}$ with $1 \leq j \leq N$ and
$\sigma_i \subseteq \{y_1 \ldots y_L\}$ with $0 \leq i \leq N$, and with
$U\xi_j = \{X_1,\ldots,X_M\}$.
From the formal point of view the representation of actions having other actions as arguments is expressed by ($y_1 \ldots y_L$) being actions in their turn.
We call "virtual actions" those actions of a plan mentioned as arguments of other actions of the plan. More definitely we define **virtual actions** those mentions, arguments of actions (in general of communicative actions), referred and allowing to mention other actions of the plan, to be executed.
In PR, and in particular in CHAPLIN, there are constraints on the arguments of the actions that are about to be recognized; these constraints have to be extended also to the possibility that the actions' parameters be actions in their turn.
The problem here is that of being able to deal with simple or complex actions inserted as arguments of other actions, with a structure reminding nesting in natural language. The critical point arises in the fact that in the recognition of a plan including both the communicative act of quoting an action and the realization of the action itself, it is necessary to be able to recognize both of them and to verify if the match is correct.
From a formal point of view, thanks to the introduction of virtual actions, a plan can satisfy this constraint rewriting:
$P(X_1,\ldots,X_M;\sigma_0;A_{vir})$ -->
$A_1(\xi_1;\sigma_1)\ldots A_N(\xi_N;\sigma_N)$ EXEC(A_{vir})
where: A_{vir} is contained in at least one of σ_i with $1 \leq i \leq N$ and it can represent either a simple or a complex, either a practical or linguistic action and where EXEC expresses the fact that an action forming the plan is the agent counterpart of A_{vir} itself in the plan.
Let us imagine the "sell-buy" plan (Figure 2) including the buyer's asking to the seller for giving him a certain object and therefore the actual action of giving the object (realized by the seller). Alongside with the classical plans Virtual Actions play an important role in the case of interaction protocols; we describe 4 simple main protocols (MAPs) in the dialogue where virtual actions are represented: Request-Plans, Offer-Plans, Permit-Plans and Order-Plans [CWo93].
In Fig. 3, Act(A_x, $\sigma_1,\ldots,\sigma_n$) can be both an elementary and a complex action; moreover, the virtual actions can be instantiated with regard to all of the arguments, with just a part of them, or they cannot be instanced at all.

Sell-Buy (Buyer Seller Give (Seller Buyer Thing-to-sell)) ->

Ask (Buyer Seller Give (Seller Buyer Thing-to-sell))
Give (Seller Buyer Thing-to-sell)

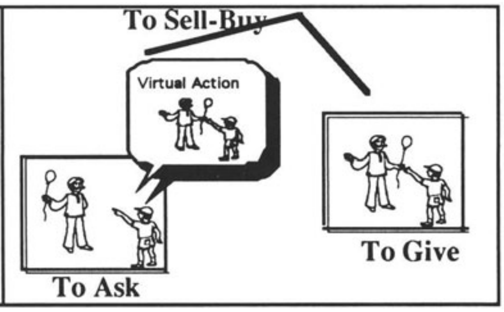


Figure 2

1) Request-Act-Loop(A_1,A_2, Act(A_2,σ_1,..,σ_n)) --> Ask(A_1, A_2, Act(A_2, σ_1,..,σ_n)) Commit(A_2, A_1, Act(A_2, σ_1,..,σ_n)) Exec(Act(A_2, σ_1,..,σ_n))	2) Offer-Act-Loop(A_1, A_2, Act(A_1, σ_1,..,σ_n)) --> Offer(A_1, A_2, Act(A_1, σ_1,..,σ_n)) Accept(A_2, A_1, Act(A_1, σ_1,..,σ_n)) Exec(Act(A_1, s_1,..,s_n))
3) Permit-Act-Loop(A_1, A_2, Act(A_2, σ_1,..,σ_n)) --> Ask(A_2, A_1, Act(A_2, σ_1,..,σ_n)) Permit(A_1, A_2, Act(A_2, σ_1,..,σ_n)) Exec(Act(A_2,σ_1,..,σ_n))	4) Order-Act-Loop(A_1, A_2, Act(A_2, σ_1,..,σ_n)) --> Order(A_1, A_2, Act(A_2, σ_1,..,σ_n)) Exec(Act(A_2,σ_1,..,σ_n))

Figure 3

That is to say, it is possible there is either a Request-Act-Loop(A_1, A_2, give(A_2, A_1, paper)) or a Request-Act-Loop(A_1, A_2, give(A_x, A_1, paper)) where A_x indicates that the subject of the action "give" is not instantiated in a specific agent.

Even though CHAPLIN deals with the problem of virtual actions just in the light of PR, this is a general problem about plan theory.

There is no strong representation of MAP (both in generation and in recognition) without the capability of representing such a connection between the mentioning of an action and the occurrency of the action itself. For instance, if agent A says to agent B: "give me the protocol number", and the agent B gives A a protocol number, we cannot be able to recognize this plan without understanding that the action of B is the same action mentioned by A. In a similar way it is not possible to generate a MA plan by transforming a SA plan, just introducing a communicative action like "give me the protocol number" into it, without considering the relationship between the communicative action and the practical action of the plan "to give the protocol number" itself.

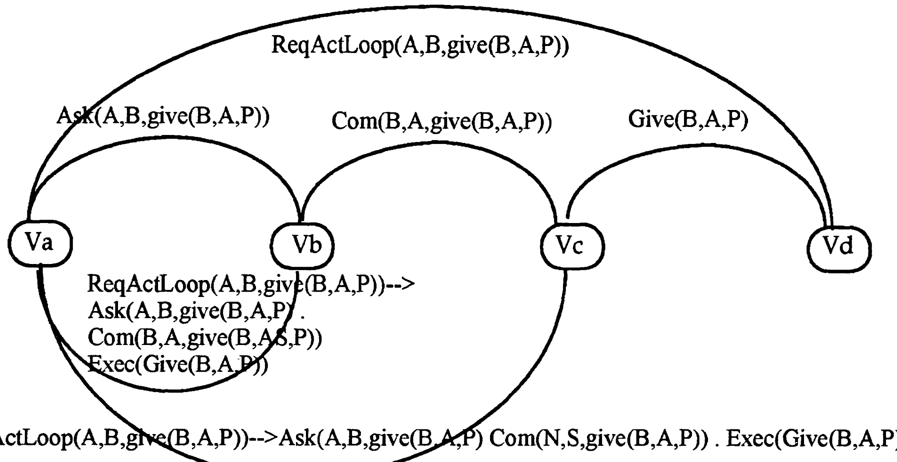
In Fig. 4 you can see how we can represent in the chart the recognition of a plan including the request of an action and its realization. In the example, for the sake of clarity, we avoided declaring the constraints to be set on the arguments and on the *virtual actions.*

The input provided to CHAPLIN, allowing it to recognize plans, consists of a description of the states of the world and of the actions in a normalized form corresponding to phrasal structures such as: (Give Seller Boy Ballonet) (Stay-Up Boy Street), (Blow Wind), etc.. CHAPLIN applied to SAP did not foresee nested structures such as (Ask Boy Seller (Give Seller Boy Ballonet)) that were simplified, whenever it was possible, into first level structures (ex: (Ask Boy Seller Ballonet)).

However, operating on multiagent plans, the problem about clearly dealing with nested actions turns out to be unavoidable.

The input of CHAPLIN for MAPs admits nested structure: the system must be able to recognize a virtual action (either a simple or a complex one) in the nested structure.

A practical development of the system embedded into a CSCW system would require the capability of recognizing the practical actions by actual operations made by the user on the computer (for example: to assign a protocol number, to send a document, to spell a text, and so on) rather than by propositional descriptions of them (such as in CHAPLIN).



ReqActLoop(A,B,give(B,A,P))-->Ask(A,B,give(B,A,P) Com(N,S,give(B,A,P)) . Exec(Give(B,A,P))

given the rule:

ReqActLoop(x,y,Avir(y,x,z))-->Ask(x,y,B,Avir(y,x,z) Com(y,x,Avir(y,x,z)) Exec(Avir(y,x,z))

where: Com = Commit, A= Agent1, B = Agent2, P = Paper Avir = Give

Figure 4

At that point the completion of the recognition of the plan by the PR system implies the recognition of the accomplishment of the commitment, and it could imply on the one hand, the erasing the task of the performer in the agenda and, on the other hand, the automatic sending of the message concerning the happened performance. In other words, this could represent a typical task for a Software Agent supporting cooperation [Acm94].

5. RECOGNITION OF VIRTUAL PLANS

By "Virtual Plan" we mean a set of virtual actions (within the same discourse) corresponding to the actions forming a plan.

In the field of recognition there is a very remarkable and complex problem: the one of the recognition of the plan on virtual layer. In fact we assert that a proper comprehension of a multiagent interaction requires that the system recognizes when the agents argue or negotiate about a series of actions which are parts of a single plan. For instance, taking hint from the plan analyzed by [Kau87]: if A says to B "make pesto and make spaghetti and boil " or if A says to B "make pesto" then he says to him "make-spaghetti" and then he says to him "boil", the system should be able to recognize that A asked B the plan "make-spaghetti-pesto" and its abstraction "make-pasta-dish". This according to the fact that the three actions *Make-pesto*, *Boil* and *Make-spaghetti* are the parts forming the plan *Make-spaghetti-pesto* (part-of) that, in its turn, is an instance of the plan *Make-pasta-dish* (abstraction).

This means that the system must use its plan library by applying it to the virtual actions to recognize merely virtual plans, that is to say plans we talk about. And then it must use the same *scheletal plan* to recognize from the exec actions the same actually executed plan. We also assume (see the following paragraph) that without the recognition of virtual plans it would not be possible to compare them with the actually recognized executive plans.

It is necessary that a series of constraints control this recognition process of virtual plans in the communication acts. A general constraint posed by the Chart-based approach, in force also in this case, is that the order by which the virtual actions are mentioned in these communication acts must correspond to the order by which the actions appear in the rewriting rules of plan in the grammar. Currently, a more specific constraint is also in force: virtual actions forming a virtual plan must be argument of a same meta-action. In other words, we postulate that:

a) if x executes a meta-action on single actions forming a plan, he executes that meta-action on the plan itself; vice versa b) if x executes a meta-action on a complex action it is just as if he executed the same meta-action on each component action (if only one decomposition of that plan exists). That is: Ask(A B Make-spaghetti-pesto) corresponds to Ask(A B Make-pesto) and Ask(A B Make-spaghetti) and Ask(A B Boil).

Not only we would be able to recognize virtual plans formed by virtual actions being arguments of meta-actions but we should recognize the same plan if it is argument of different meta-actions (this permits, for example, to reach the agreement in the negotiation between the customer and the performer); consider this case:

Request[A, B, make-spaghetti-pesto] Commit[B, A, {make-pesto make-spaghetti boil}].

As one can see, the introduction of the notion of virtual actions and virtual plans creates a double level of PR and of matching: On the one hand, the recognition of virtual plans and their match with each other, which also implies the recognition of conversational plans (that we called metaactions); on the other hand the recognition of the executive plans and their match with the virtual ones.

6. THE PROBLEM OF MATCH BETWEEN VIRTUAL ACTIONS OR PLANS AND EXEC ACTIONS OR PLANS

In the correspondence between virtual actions and plans and executed actions and plans, we can establish the following possibilities:

When there is a *simple virtual action* this has to meet correspondence in an elementary exec action (Interaction Protocol in Fig.5); when a *complex virtual action* is mentioned (for instance make-spaghetti-pesto) there are two possibilities: - either the input describing the real actions directly mentions the execution (allowed by CHAPLIN) of that complex action (Interaction Protocol2), or - the complex virtual action can be recognized as executed, just when a corresponding plan is recognized by elementary exec actions (Make-pesto, Make-spaghetti, Boil) (Interaction Protocol3); when in the communicative act there is a *virtual plan* (that is to say a series of simple or complex virtual actions forming a plan) then the recognition of its execution happens in the following way: - either the input describing the real actions directly mentions the execution of a complex action corresponding to the plan (Interaction Protocol4), or - the virtual plan can be matched thanks to the recognition of the plan composed by the exec actions that correspond, one to one, to the virtual actions (Interaction Protocol5).

BIBLIOGRAPHY

[Bog91] Bogen D., Linguistic Formsand Social Obligations: a critique of the doctrine of literal expression in Searle. *Journal for the Theory of Social Behavior*, vol.21, 1,31-62, 1991

[Acm94] ACM., Special Issue on Intelligent Agents, *Communications of the ACM* 37, 7, July, 1994

[CFa94] Castelfranchi, C., Falcone, R., Towards of Single-Agent into Multi-Agent Plan Transformation; *PRJCAI '94*, Pechino, 1994.

[CWo93] Ghang, M.K. and Woo, C., SANP: A Communication Level Protocol for Negotiations. In E. Werner & Y. Demazeau, *Decentralized AI 3,* North Holland, Amsterdam, 1993

[FCa93] Falcone, R., Castelfranchi, C., CHAPLIN: A Chart based Plan Recognizer, Proceedings of the *Thirteenth International Conference of Avignon*, Avignon, France, 24-28 May, 1993.

[FCa94] Falcone, R., Castelfranchi, C. (1994), Plan Recognition: from single-agent to multi-agent plans. Proceedings of MAAMAW'94 Odense, 3-5 August, 1994.

[Gro74] B. Grosz, The Structure of task oriented dialogs. In *IEEE Symposium on Speech Recognition: Contributed Papers.* Carnegie Mellon University Computer Science Dept., Pittsburgh, Pennsylvania: 250-253.

[Kau87] Kautz, H. A. (1987). A Formal Theory of Plan recognition. *PhD thesis, University of Rochester*, 87. Also TR215, University of Rochester, dept of Computer Science, Rochester, N.Y.

[MWF92] Medina-Mora R., Winograd T., Flores R., Flores F., The Action Workflow approach to workflow management technology, proceedings of *CSCW '92.*

[Suc94] Suchman, L., Do Categories have politics? The Language/Action perspective reconsidered, *Computer Supported Cooperative Work*, vol 2, 177-190, 1994.

[Vil90] Vilain M., Getting Serious about Parsing Plans: A Grammatical Analysis of Plan recognition. In *Proc. of the IJCAI*, 190-197, Boston, 1990.

[Win87] Winograd, T., A Language/Action Perspective on the Design of Cooperative Work, *Human-Computer Interaction*, vol. 3, 1, 3-30 1987

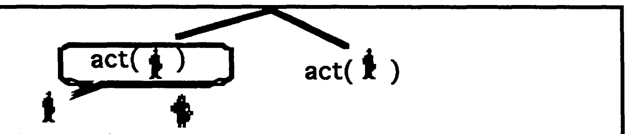
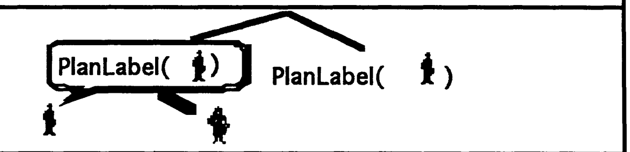
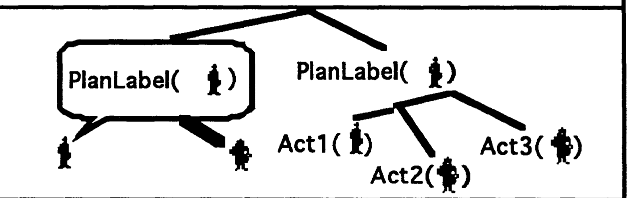
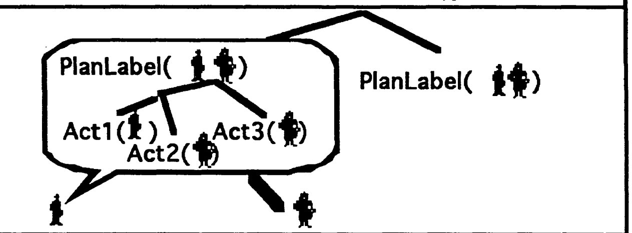
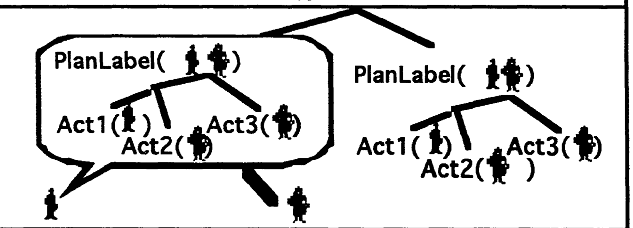
Interaction Protocol1 --> MetaAct({Ag1},Act({Ag2})) Act({Ag2})	
Interaction Protocol2 --> MetaAct({Ag1},Plan({Ag2})) Plan({Ag2})	
Interaction Protocol3 --> MetaAct({Ag1},Plan({Ag2})) Act1({Ag3}) Act2({Ag4}) Act3({Ag5}) with: Plan --> Act1 Act2 Act3 and {Ag2} = U {Ag3} {Ag4} {Ag5}	
Interaction Protocol4 --> MetaAct({Ag1},Act1({Ag2}),Act2({Ag3}), Act3({Ag4})) Plan({Ag5}) with: Plan --> Act1 Act2 Act3 and {Ag5} = U {Ag2} {Ag3} {Ag4}	
Interaction Protocol5 --> MetaAct({Ag1},Act1({Ag2}),Act2({Ag3}), Act3({Ag4})) Act1({Ag2}) Act2({Ag3}) Act3({Ag4})	

Figure 5

In Fig.5: *MetaAct* corresponds with an interaction (dialogue in particular) action or plan, *{Ag}* indicates the action's protagonist (one or more than one), *Act* represents an elementary action, *Plan* represents a complex action. We have neglected the other arguments of the action.