Agent Interactions Through Coordinated Work

DAOUD Faouzi

Laboratoire PRiSM, University Paris VI
45, Av des Etats Unis
78035, Versailles, FRANCE
Tel: 33 1 39254076
fax: 33 1 39254057
(fad@prism.uvsq.fr)

1. Introduction

Systems supporting work processes should achieve high challenges to support integrated features (customer-performer relation, cooperation network, action flow, communication flow, organisational structure, etc).
Our aproach to achieve the challenge is to design an architecture based on distributed intelligent agents. The reflection of the embodiement of the language action perspective, present in workflow, on the architecture and the environment of agents make them efficient enough to achieve our goals.

2. Requirements of Workflow Interactions Supported by Agents

A model of multi-agent system supporting workflow management should enable workflow to comprise the following four phases:
1) preparation: the customer requests (or the performer offers) completion of a particular action according to some stated conditions of satisfaction.
2) negotiation: the two parties come to mutual agreement on the condition of satisfaction, including the times by which further steps will be taken.
3) performance: the performer declares to the customer that the action is complete.
4) acceptance: the customer declares to the performer that the completion is satisfactory.
Recall that the conditions of satisfaction are the conditions agreed upon, by the customer and performer, under which the workflow will be performed. This agreement is only partially explicit in the negotiations, resting on a shared background of assumptions and standard practices.
During the negotiation phase, secondary workflows relate the negotiation activities that can lead to counter-offers.

The performance phase typically has secondary workflows in which the performer delegates work to other process participants, including external performers.
Finally, in the acceptance phase the customer can initiate secondary work flows to evaluate the completed work.
Secondary workflows can be further decomposed until the roles for all process participants have been defined as a network of workflows. This workflow network can reflect any hierarchical, sequential or parallel arrangement that is appropriate for the definition or design of a particular business process.

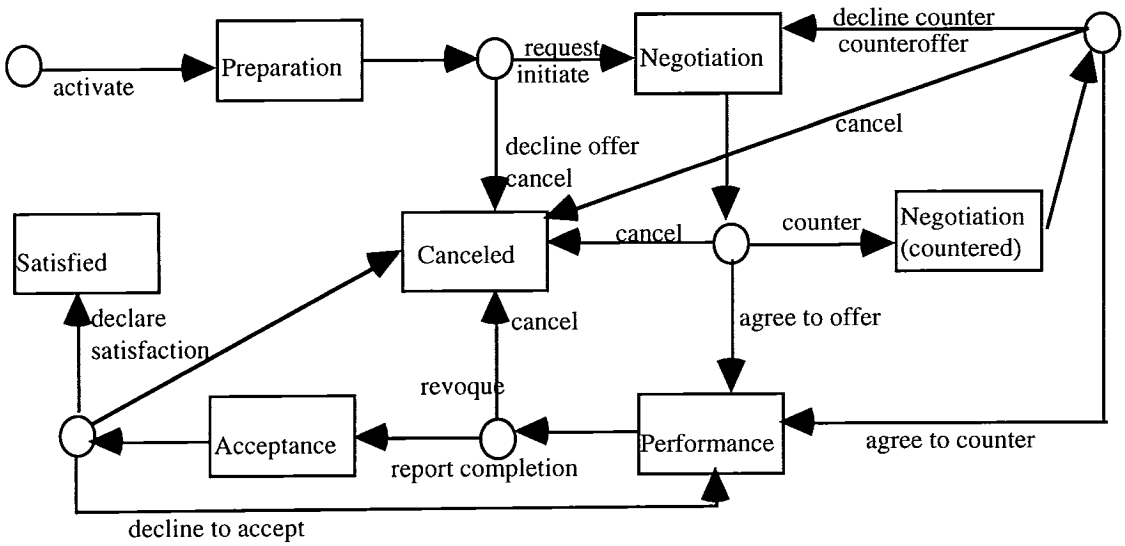


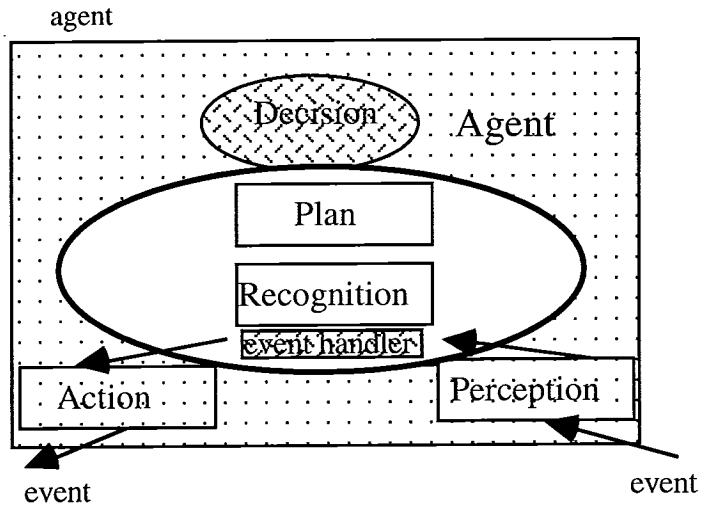
Fig 1: WORKFLOW PHASES

3. Description of the Agent

An agent can be seen as a self contained unit with a set of inputs (perception) an a set of outputs (action). A formal definition of agent properties must include a framework for describe the agent's world(4).

We adopt Rasmussen's conceptual model as a framework to develop an agent architecture that evolve in a world inhabited by other agents. This model is driven by the goal of combining the complementary advantages of reactive, planning and decision-making systems in order to take into account different situations which arise in multi-agent environments. First, it needs to be reactive (mechanism) to be able to quickly respond to changes in its environment.

Fig 2: Description of the software agent



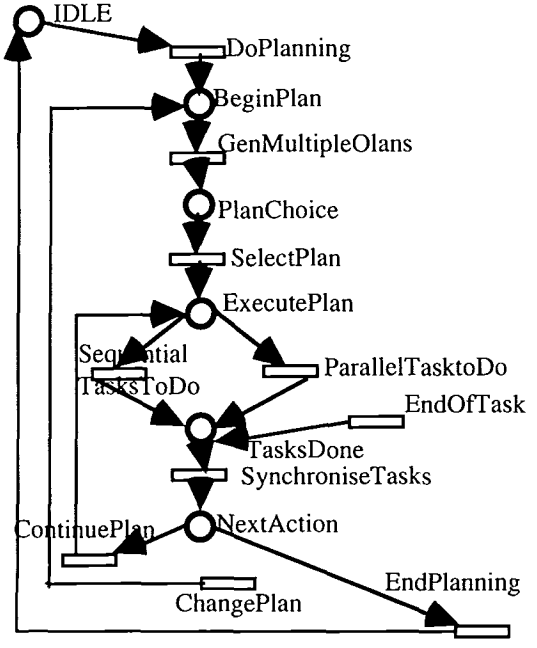
Secondly it should be capable to plan its activities for a recognised task or goal (controller). Finally, the model must also allow reasoning about others (model) since agents should be capable of making decisions that take into account their own intentions and also others' intentions.

First, perceived information from the environment leads the agent to execute an action if the corresponding situation is perceived in terms of action. If this is not the case, the agent tries to recognise the situation. It can recognise the situation in terms of an action or in terms of a goal (or a task). In the first case, it tries to execute the corresponding action, and in the second case it invokes the planning module. Finally, if the agent faces an ambiguity and cannot come to a decision, then it invokes the decision making module (model)to make a decision in order to commit to achieve a goal or an action.

4. Planner Agent in Work

An activity consist of an explicit division and synchronisation of tasks among agents with common goals.
Planning is viewed as an arranged set of operations working out wanted goals, alleging the only changes of the state of the world which are caused by planned operations, the contributions SequentialTasksToDo, ParallelTaskToDo and EndOfTask are speech acts of type request. The set of possible exchanges are (SequentialTasksToDo, TasksDone) and (ParallelTaksToDo, TasksDone)

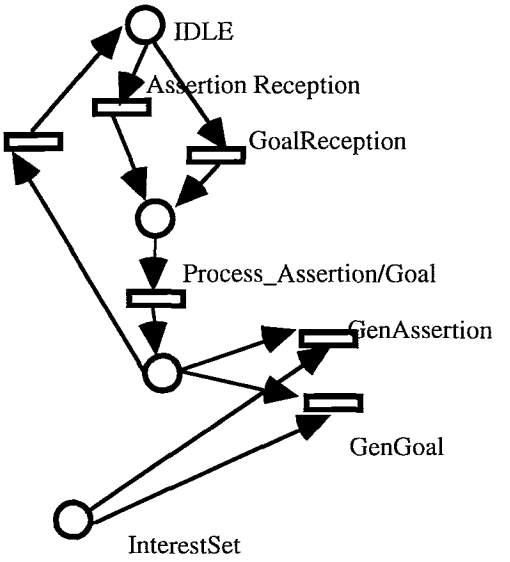
Fig 3: PLANNER



5. Agent Organizer

Organising can be seen as a process based on a set of assertions and refutations of questions applied to an intuitive structured organisation.

Fig 4: ORGANIZER



The contributions AssertionReception(Inform), GoalReception(Request), GenAssertion(Inform), GenGoal(Request) are speech acts. The set of possible exchanges are (AssertionReception, genassertion), (AssertionReception, GenGoal), (GoalReception, GenAssertion) and (GoalReception, Gengoal).

6. Coordination of Agents

Coordination is the control aspect of cooperative work. That is, it deals with the allocation, scheduling, and monitoring of cooperative activities in order to efficiently achieve a goal given that there will be limited resources, agent capabilities, and time.
Coordination between agents generally decreases from routines to unfamiliar situations. In order to strengthen the levels relative to routines and familiar situations, we enrich each agent with social regularities (cooperative rules,

coordinative rules, collective rules) and social collectivities (e.g. roles, groups, organisations..) in the form of social laws. Cooperative rules reflect behaviour rules whose performance of an action by on agent makes sense only if many others do the same thing. Collective rules organise collective actions out of individual effort, they divide labour in an organisation. By doing this, we assume that the agents adopt these social laws and each agent obeys these laws and will be able to assume that others will as well. Agents will then have facilities to coordinate its activities with other agents, and negotiation is requested only when necessary.

An agent (i) perceives a vast array E and elaborate some desired state that it communicates to another agent (j) (under its responsibility). The desired state can be elaborated at any level (skills, rules, knowledge) of agent (i) depending on its experience and on the considered situation . In turn, agent (j) compares the desired state to its goals or its actions. This comparison can lead it to negotiate, to refuse or to act according to the hierarchical links between it and agent(i). In the case where it decides to act, the agent(j) acts on an environment under control of agent(i).

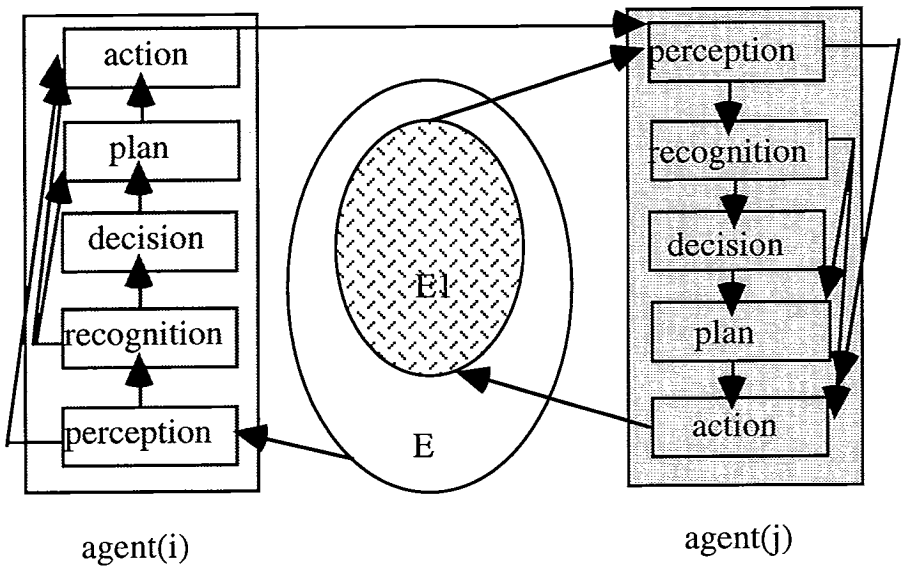


Fig 5: Collaboration between two agents

7. Accountability and Commitment

The key structure used by a manager when making decisions about task allocation is that of accountability defines both for what and to whom an agent is responsible and it can be expressed in the following manner: accountable(Agent1, Agent2, TaskType). The task manager component uses its accountability relations, together with the generic inference rule, to pick the most appropriate contractor for a given task.

Accountability alone does not guarantee commitment: to commit to a specified task., an agent must also have the necessary resources which are required to perform that task. These

necessary resources can be related to both time and material.

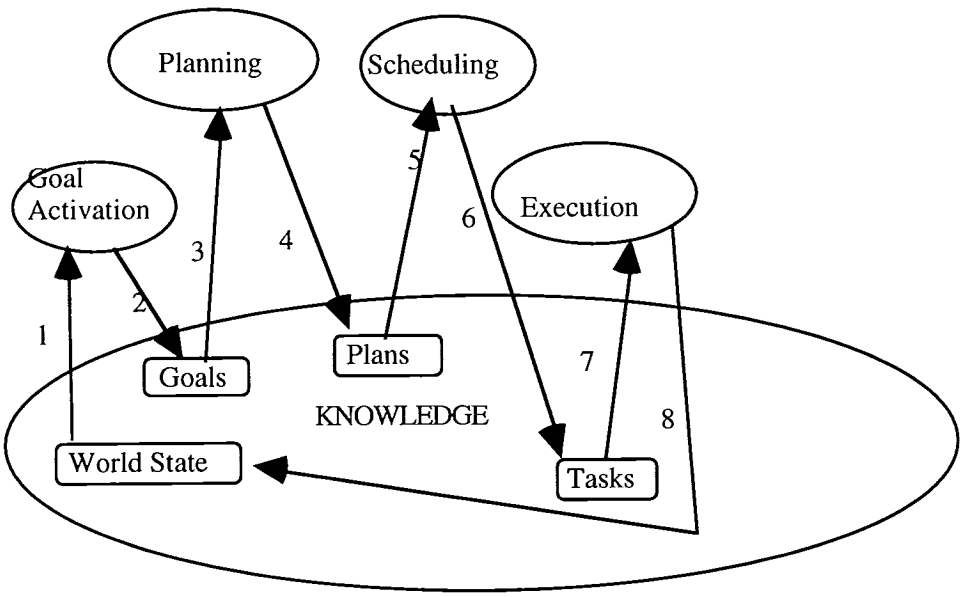
Although agents know what resources are available to themselves, generally they do not have information about the resources of their acquaintances, therefore a task may have to be iteratively delegated to a number of acquaintances until a specific agent becomes committed.

When an agent accepts a request he becomes committed to performing it (i.e. he commits to undertake the role of contractor to the originator for that task):

if Acquaintance is requested by Agent to perform Task,
and Acquaintance is accountable to agent to perform Tasks,
and Task requires Resources,
and Resources are available to Acquaintance
then Acquaintance becomes committed to task

In most cases, when an agent commits itself to perform a task then that task will actually be executed.
If a contractor drops its commitment to a task, then it is imperative that it performs the manager. Similarly, if the manager realises that a task which it has contracted out is no longer valid then it should inform the relevant acquaintance so that the necessary effort is expanded.

Fig 6: Knowledge Processing Within an Agent



8. Agent Control For Dynamic Environment

An agent must adapt its control mode to dynamic goal-based constraints on its actions and uncertainly about its environment. The agent must also adapt its meta-control strategy to its dynamic configuration of demands, opportunities, and resources for behaviour.

We can characterise control situations on several dimensions, including the uncertainly of events in the task environment, the degree of constraint on which sequences of actions will be effective in achieving goals, and the availability and cost of off-line and on-line computational resources. Control modes are characterised by the agent' sensitivity to run-time events and its advance commitment to specific actions. Sensitivity to run-time events measures how much the agent monitors its run time environment. Commitment to specific actions measures how much the agent restricts in advance the actions it will execute at run time.
In a planning mode, which is appropriate for control situations with low environmental uncertainly and high constraints on the selection on sequencing of effective actions, an agent commits in advance to a sequence of actions, and executes it at run time with minimal monitoring of run-time events.

In a reactive mode, which is appropriate for control situations with high uncertainly and high constraints on effective actions, the agent commits in advance to a set of specific actions and conditions for their execution, and monitors run-time events to control invocation of particular actions from the set.
In intermediate modes between these two extremes modes, the agent modulates the amount of run-time monitoring and the balance of top-down versus bottom-up control of actions.

The meta-control problem is: How should an agent allocate its limited computational resources among dynamic configurations of competing and complementary activities so as to achieve a high overall utility of its behaviour.
An agent can adapt its meta-control strategy to its dynamic configuration of potential activities by: analysing control plans representing intended activities to estimate their resource requirements; assessing the availability of required resources in the prospective situation; and making or modifying meta-control decisions that establish appropriate constraints on the constructions of activity-specific control plans.

9. Conclusion

Many Distributed Artificial Intelligence systems do not embody a principled model of collaboration upon which agents can base their decisions and subsequent actions about cooperation.
Our approach have the aim to build a suitable model based on agents for workflow systems, the choice of the capabilities of our agent architecture and the model of agent coordination was fully influenced by workflow features.

REFERENCES

(1) Bracchi.G and Pernicci.B, " Design Requirements of Office Systems", ACM Transaction on Office Information Systems, vol.2, pp151-170, 1984.
(2) Clarence A. Ellis, "Office Information Systems and Computer Science", Computing Surveys, Vol. 12, N 1, pp27-60, 1980.
(3) Hirscheim, "Office Automaton: A Social and Organisational Perspective", John Wiley information system series,1994
(4) Genesereth M R and Ketchpel S, "Software Agents", Communications of the ACM, Vol 37, July 1994.
(5) Lin Q and Srinivas K, "Infrastructure Support for Multimedia Communications", Proceedings of the second Workshop on enabling Technologies, 1993
(6) Baecker RM , "Reading in Groupware and CSCW, Morgan Kaufmann Publishers 1993
(7) Gruber T, " Ontolingua: A mechanism to support portable Ontologies", KSL-91-66, Stanford KSL, 1991
(8) Susan and Al , "Multistage negotiation in Distributed Planning", Readings in DAI- 1988
(9) Sandip, "a formal study of distributed meeting schduling" ACM conference Organisational Computing Systems, 1991
(10) Richard Filkes and Al, " Knowledge Sharing Technology", KSL Stanford , November 1991
(11) Coriat, "Formal specification using agents conceptualization", Labo MASI, Universite PARIS VI