An Operational Semantics Dedicated to the Coordination of Cooperating Agents

Mohamed Jmaiel, Ahmed Hadj Kacem and Amira Rgaieg

LARIS Laboratory
FSEG-SFAX B.P. 1088
3018 SFAX - TUNISIA
Email:Mohamed.Jmaiel@enis.rnu.tn,Ahmed@fsegs.rnu.tn

Abstract. This paper presents a contribution towards rigourous reasoning about coordinating agents. First, it defines formal models for coordination and coordinating agents. These models enable to specify the relations between the concepts of: plan, plan proposal and resource allocation, on the one hand, and concepts of: knowledge, belief and capability, on the other hand. Second, it provides a structured coordination language enabling to specify primitives, protocols and processes of coordination. This language is defined by a precise syntax, and it is formally interpreted using a transition system leading to an operational semantics for coordinating agents.

Keywords: Operational semantics, Specification language, Formal validation, Transition system, Coordinating agent, Plan, Proposition, Resources allocation.

1 Introduction

In contrast to traditional Artificial Intelligence (AI) which models the intelligent behaviour of a single agent, the Distributed Artificial Intelligence (DAI) is interested in intelligent behaviour resulting from cooperative activities of several agents. The transition from individual behaviour to collective behaviour is considered as an enrichment of the AI, from which new properties and activities emerge. The *interaction* between agents is one of the main results of the collective activity, which tries to increase the performances of the problem resolution at both individual and collective level. Indeed, the interaction constitutes a central concept in Multi-agent Systems (MAS). It occurs due to the inevitable interdependence between agents, and it appears in different forms, namely cooperation, coordination, negotiation and communication.

In this paper, we mainly focus on the coordination as a fundamental mechanism to maintain coherence between agents and to solve possible conflicts which may occur between them. In general, the coordination may take different forms, such as organizational structuring, contracting, planning and negotiation [HNJ96]. Many models and protocols have been designed to support the coordination activity, such as coordination through coalition formation [SSJ97], coordination by plans conversation, and coordination using contract nets [YHH^{+}98]. Most of the proposed models are based on the concept of *plan*. Accordingly, during the coordination process, agents communicate in order to build and to update their individual plans and others plans, while trying to avoid possible conflicts and to enhance performance. This technique was mainly adopted by Durfee [DL89] in its model of Partial Global Planning (PGP), and also by Decker and Lesser [DEC 94] in their model of Generalized Partial Total Planning.

In most case, the proposed models are specific to particular domains, and they handle coordination at a low level. This makes them concrete and relatively vague, what complicates reasoning about them. Indeed, neither the process of coordination is formally specified nor the corresponding communication primitives are rigourously clarified. We notice an absence of a methodological approach to specify coordination at both collective (MAS) and individual (agent) levels.

The work presented in this paper contributes to the top-down design of coordinating agents. Our approach is based on negotiation of plans and resources allocation, and on coordination relationships. Indeed, we propose two models for coordination and coordinating agents which are domain-independent. The first one is based on the generation, the evaluation and the agreement with a proposition. The second expresses the individual aspects of an agent allowing it to coordinate while remaining coherent with the coordination model. Besides, we propose a structured language to specify communication primitives, protocols and coordination processes. This language is defined with an accurate syntax, and it is formally interpreted using a transition system which provides an operational semantics for coordinating agents.

The originality of our approach is mainly located in the complementarity of the suggested model which covers three facets. First, a communication language for negotiation according to agents communication languages (ACL) [FIP97], like COOL [BF95], and KQML [FF94]. Second, a formal language to specify a coordination protocol by means of parallel process in the style of process algebra, like CSP [Hoa85], and CCS [Mil80]. Finally, a formal tool (transition system) enabling to verify the dynamic behaviour of an individual agent and a system of coordinating agents.

2 Formal model and notations

This section provides formal notations and definitions of concepts related to coordination among multi-agent systems. These concepts are detailed in [HKJ01]. This formalization, which is based on a set theoretical language, enables to highlight the main elements characterizing coordination and to develop a formal operational semantics for the coordination process. We contribute to the design of this process by specifying two models, namely a model of coordination and a model of coordinating agents. The first model describes the main elements related to the coordination process, while the second specifies the internal concepts and the properties of a coordinating agent.

2.1 Coordination model

The coordination model constitutes the basis of our formal definition. In this definition we suppose the existence of three sets. The set $\mathcal{AG}$ of agents, $\mathcal{BG}$ of global goals, and $\mathcal{BL}$ of local goals. A global goal is common to several agents, while a local goal belongs to only one agent. A *coordination model* is defined by a couple (Ag, Oc), where $Ag \subseteq \mathcal{AG}$ and $Oc \subseteq \mathcal{BG}$ (Ag and Oc should be not empty). The set Ag represents the society of agents, while Oc denotes a *common goal*. The language for global and local goals may be based, for examples, on a propositional logic [CHJ01] or a modal logic [Sho93].

The coordination process is carried out in four main phases which may overlap. The first one consists of proposing global plans for a given global goal. Once a commitment on a global plan is made, the agents start the second phase aiming at assigning to each agent the local goals it has to achieve. The third phase enables the agents to coordinate their access to the shared resources. Finally, the agents coordinates their tasks according to a causality relation.

- *Global plan:* this step consists of decomposing a given global goal in a non empty set of local goals. Formally, the set of global plans, denoted by $\mathcal{L}_{gp}$, is defined as the set of subsets of $\mathcal{BL}$. During a coordination process, an agent $A_i \in Ag$ can propose a global plan for a given global goal $Bg_j \in Oc$. This plan is denoted by $PG(A_i, Bg_j)$.
- *Local plan:* this step distributes the retained global plan between the agents. Formally, each proposition is considered as an element of a partition of the retained global plan. Let Bg_j be a global goal, Pg a retained global plan for Bg_j, and A_i an agent. A local plan made by the agent A_i according to Pg is, in fact, a subset of Pg. The set of these propositions is denoted by $PL(A_i, Pg)$. $\mathcal{L}_{lp}$ denotes the set of possible local plan propositions. When agents propose

local plans, two situations may occur. In the first case, the propositions made are disjoint and consequently there is no conflict and no need to negotiation. In the other case, the agents have to solve the conflict and select the appropriate agent for carrying out a local goal proposed by more than one agent. After negotiation, each agent A_i is assigned a set of local goals called local plan, denoted by $Plan(A_i, Pg)$.

- *Resource allocation*: according to its plan, each agent A_i defines an agenda of tasks to accomplish. This agenda includes a partially ordered set of tasks. While performing their tasks, usually, agents access not only to local resources, but also to shared resources. This involves simultaneous accesses to a same resource, which can generate conflicts. In order to better manage shared resources, it is necessary that the agents put an ordering between tasks accessing to a shared resource. Whenever there is two tasks t_1 and t_2 accessing to a same resource r, an execution ordering between them has to be defined. A proposition for tasks ordering is of the form $\mathbf{after}(t_1, t_2)$ saying that t_1 should be performed after t_2. We denote the set of propositions for tasks ordering by $\mathcal{L}_{ra}$.
- *Tasks coordination*: in this phase the agents set a partial order on generated tasks according to hard and soft constrains. A hard constraint expresses the fact that a task has to precede another. A soft constraint states that a task may facilitates or hinders another. Accordingly, we enrich our formalism, with the following primitives which principle was evoked by [DL94] under the name of *coordination relationship*. The first, $\mathbf{enables}(t_1, t_2)$ is a hard coordination relationship stating that t_1 must be completed before starting t_2. The second, $\mathbf{facilitates}(t_1, t_2)$ is a soft coordination relationship saying that the result of t_1 may help to perform t_2. The third, $\mathbf{hinders}(t_1, t_2)$ is a soft coordination relationship indicating that the result of t_1 decreases the performance of t_2.

The union of the three proposition sets gives rise to a proposition language $\mathcal{L}$ defined as $\mathcal{L} \overset{\text{def}}{=} \mathcal{L}_{gp} \cup \mathcal{L}_{lp} \cup \mathcal{L}_{to}$. During a coordination process, agents will exchange messages containing propositions. In our approach, an agreement about a proposition is achieved with a total acceptance made by all agents. A proposition $p \in \mathcal{L}$ is *retained*, if it is accepted by all agents of the society, or if it is imposed by the agent possessing the highest confidence coefficient, in the case of disagreement. Let A_i be an agent and $p \in \mathcal{L}$ a proposition made by A_i :

$$\begin{aligned} AcceptTot(Ag,\, A_i,\, p) \text{ iff } & (\forall A_k \in Ag\ Accept(\theta_{A_k}, p)) \quad \text{or} \\ & (\forall A_k \in Ag \backslash \{A_i\}\ Coef_{A_i} > Coef_{A_k}) \end{aligned}$$

where $Accept(\theta_{A_k}, p)$ is a predicate stating that the agent A_k accepts the proposition p according to its mental state θ_{A_k}, and $Coef_{A_i}$ is a value associated to each agent that describes its degree of confidence.

2.2 Coordinating agent model

The coordinating agent model expresses the individual aspects of an agent. Such an agent is characterized by its mental state which is the result of the interactions with its environment. Some properties characterize a coordinating agent such as communication, autonomy, reasoning, evaluation, etc. A description of these aspects, namely mental state and properties, is detailed in the following:

- *Mental State* is a set of knowledge, a retained global plan, a retained local plan and a tasks agenda. *Knowledge* is represented by a fact of which the agent is sure of its truth. The set of knowledge, denoted by $\mathcal{K}$. The retained propositions (global plan and local plan) are elements belonging to the proposition language $\mathcal{L}$. An agenda includes the set of tasks T_A that the agent A should perform as well as two partial ordering on tasks: a local ordering $O_l \subseteq T_A \times T_A$ which defines a causality relation on the tasks of A, and a global ordering $O_g \subseteq T_A \times T \cup T \times T_A$ (T

is the set of tasks of the other agents) which sets a causality relation between tasks of A and tasks of other agents.
Formally, the mental state θ_A of an agent A is a structure $\langle \mathcal{K}_A,\ Ret_{gp},\ Ret_{lp},\ (T_A,\ O_l,\ O_g)\rangle$ including the concepts presented above.

- *Properties* describe the essential properties of a coordinating agent. Such an agent should be able to propose, to evaluate and to take decisions. We retain the following four properties:
  - *Communicating* is the ability to exchange information of various types: knowledge or propositions for global plans, local plans or resource allocations. This ability is ensured by three primitives, namely **send**, **receive**, and **broadcast**.
  - *Proposing* is the ability to generate propositions or counter-propositions, according to a mental state and already made propositions. This ability is guaranteed by primitives such as **propose** and **counter-propose**.
  - *Evaluating* is the ability to evaluate propositions made by other agents and to answer by an acceptance or a rejection. An evaluating agent should be able to execute primitives like **evaluate**, **accept**, and **reject**.
  - *Ordering* is the ability of detecting and defining ordering between its proper tasks, or between its tasks and tasks of other agents. An ordering agents is able to execute primitives like, **After**, **Enables**, **Facilitates** and **Hinders**.

  These primitives are introduced in a structured manner and formally interpreted in section 3.

2.3 Coordination structure

A part of its mental state, which contains permanent information, an agent maintains during its execution a coordination structure including auxiliary information related to conversations he made with other agents. This structure helps the agent to take decisions according to what he sent and received. It includes three sequences representing histories related to coordinations made about a global plan, a local plan and tasks ordering. We distinguish three sets of histories each corresponds to a proposition type. The set of histories H_{gp} for global plans is defined as follows:

$$H_{gp} \overset{\text{def}}{=} \{(gp, act, tp) |\ gp \in \mathcal{L}_{gp},\ act \in \{snd,\ rcv\}, tp \in \{prop,\ accept\}\}^*$$

That is, in the history we store after each negotiation action the content of the proposition made, the exchange action, and the proposition type. The sets H_{lp} and H_{ra} of histories for local plans and tasks ordering are defined similarly to H_{gp}. A *coordination structure* σ_{coord} is a triplet belonging to the set $\Sigma_{coord} \overset{\text{def}}{=} H_{gp} \times H_{lp} \times H_{ra}$.

3 Coordination Language

In this section we define the syntax of a coordination language between agents. This language builds upon concepts underpinning several well-understood concurrent programming paradigms; viz Communicating Sequential Processes (CSP) [Hoa85] and Algebra of Communicating Processes (ACP) [BK84]. In our approach, we consider a coordination task as a set of concurrent processes for the following two reasons :

- *generality*: an agent may participate, at the same time, to several coordinations with different agents;
- *flexibility*: a parallel process is, in general, more flexible than a sequential one. A parallel process does not impose any execution order of actions in different processes. Consequently, in our formalization, an agent does not specify the moment at which it answers a previous request.

In our coordination language a MAS is composed of a set of autonomous agents which share a work environment and a common objective. Each agent is represented by a mental state, a coordination plan to be executed and a coordination structure which maintains the history of the conversations carried out by the agent during a cooperative activity. It will allow the agent to take decisions, such as acceptance or rejection of a proposition.

Let $p \in \mathcal{L}$ be a proposition, θ a (mental) state of an agent, σ_{coord} a coordination structure, id_A the identifier of an agent A and ID_{Ag} the set of identifers of the agents set Ag.

We define the syntax of a message msg which is exchanged between agents, an atomic action a, a coordination plan S, a coordinating agent A and a multi-agent system Mas.

$$\begin{array}{lll} msg & ::= & \mathbf{Propose}(p) \mid \mathbf{Accept}(p) \mid \mathbf{Reject}(p) \mid \mathbf{TotAccept}(p) \mid \mathbf{CounterPropose}(p) \\ & & \mathbf{Facilitates}(t_A, t_B) \mid \mathbf{Enables}(t_A, t_B) \mid \mathbf{Hinders}(t_A, t_B) \\ a & ::= & \tau \mid \mathbf{send}(id_A, msg) \mid \mathbf{receive}(id_A, msg) \mid \mathbf{broadcast}(ID_{Ag}, msg) \mid \mathbf{evaluate}(id_A,\ p) \\ S & ::= & \mathbf{skip} \mid a; S \mid S_1 || S_2 \\ A & ::= & id_A : \langle \theta, S, \sigma_{coord} \rangle \\ Mas & ::= & \langle A_1,\ A_2,\ \ldots,\ A_n \rangle \end{array}$$

An exchanged *message* in a coordination process may be a proposition of a global plan, a local plan or a resource allocation, an acceptance or a reject of a proposition, a coordination relationship or a relation between two tasks. An *action* is an emission, a reception, a diffusion or an evaluation of a proposition. A *coordination plan* is a process *skip* which ends its execution immediately, a plan which performs an action a and continues its execution with another process, or a parallel composition of two coordination plans. An *agent* is represented by a triplet composed of a mental state, a coordination plan and a coordination structure. Note that a unique identifier is associated to each agent. A *multi-agent system* is a system made up of a non-empty set of agents.

4 Operational semantics

In this section we develop operational semantics by means of a transition system (due to Plotkin cf. [Plo81]). A *transition* takes a system from a configuration to a subsequent configuration. The transition $< \theta,\ S,\ \sigma_{coord} > \xrightarrow{\alpha} < \theta',\ S',\ \ \sigma'_{coord} >$ denoting that the agent represented by $(\theta,\ S)$ in the coordination state σ_{coord} performs an action that modifies its coordination state to σ'_{coord}. $(\theta',\ S')$ represents the part that still needs to be executed. The label α is either an internal action τ or a communication action as defined in the previous section. The transition system is composed of a set of transition rules of the form:

$$\frac{\langle \theta_1,\ S_1,\ \sigma_{coord_1} \rangle \xrightarrow{\alpha_1} \langle \theta'_1,\ S'_1,\ \sigma'_{coord_1} \rangle\ \ldots \langle \theta_n,\ S_n,\ \sigma_{coord_n} \rangle \xrightarrow{\alpha_n} \langle \theta'_n,\ S'_n,\ \sigma'_{coord_n} \rangle}{\langle \theta,\ S,\ \sigma_{coord} \rangle \xrightarrow{\alpha} \langle \theta',\ S',\ \sigma'_{coord} \rangle} \quad cond$$

This rule means that the transition below the line can be derived if the transition above the line can be derived and the validity of the condition *cond*. We distinguish three classes of actions done by an agent, notably sending and receiving messages, and evaluating propositions. This section defines the transition rules for each classe. Also, for each action, we present a rule that summarizes its effect on the executing agent's state.

4.1 Send message rules

According to our approach, a message sent by an agent is either a proposition, an acceptance, a reject or a coordination relationship.

Definition 1 [Proposition sending]

Let $a = \mathbf{send}(id_A, \mathbf{Propose}(p))$ be an action where A an agent and $p \in \mathcal{L}$ a proposition. This action specifies the identifier of the receiver agent and the content to send. The following rule states that a proposition sending updates the coordination history of the agent sender.

$$\langle \theta,\, a; S,\, (h_{gp},\, h_{lp},\, h_{ra}) \rangle \xrightarrow{a} \langle \theta,\, S,\, (h'_{gp},\, h'_{lp},\, h'_{ra}) \rangle$$

$(h'_{gp},\, h'_{lp},\, h'_{ra})$ is defined as follows:

$$(h'_{gp},\, h'_{lp},\, h_{ra}) \overset{\text{def}}{=} \begin{cases} (h_{gp}.\langle p,\, snd,\, prop\rangle,\, h_{lp}, h_{ra}) & \text{if } p \in \mathcal{L}_{gp} \\ (h_{gp},\, h_{lp}.\langle p,\, snd,\, prop\rangle,\, h_{ra}) & \text{if } p \in \mathcal{L}_{lp} \\ (h_{gp},\, h_{lp},\, h_{ra}.\langle p,\, snd,\, prop\rangle) & \text{if } p \in \mathcal{L}_{ra} \end{cases}$$

Usually, if an action is achieved it will be removed from the stack actions of a coodination plan.

The transition rule for a message sending the form **CounterPropose**(p) is analogous to the one of proposition sending. The sending of such messages does not generates any modification of the agent's mental state.

Definition 2 [Acceptation sending]

Let $a = \mathbf{send}(id_A, \mathbf{Accept}(p))$ be an action where A an agent and $p \in \mathcal{L}$ a proposition. The rule $\langle \theta,\, a; S,\, \sigma_{coord} \rangle \xrightarrow{a} \langle \theta,\, S,\, \sigma_{coord} \rangle$ states that the acceptance sending does not modify the mental state nor the coordination history of the sender agent.

Definition 3 [Reject sending]

Let $a = \mathbf{send}(id_A, \mathbf{Reject}(p))$ be an action where A an agent and $p \in \mathcal{L}$ a proposition. In our approach, we suppose that the proposition reject must be followed by a counter-proposition.

$$\langle \theta,\, a; S,\, \sigma_{coord} \rangle \xrightarrow{a} \langle \theta,\, a'; \mathbf{skip} \| S,\, \sigma_{coord} \rangle$$

where $a' = \mathbf{broadcast}(\{id_{A_1},\, \ldots,\, id_{A_n}\}, \mathbf{CounterPropose}(\mathbf{p}'))$.

This rule will not make an effect on the mental state nor on the coordination history of the sending agent. It modifies the coordination process while inserting a new action that broadcasts a new proposition. The diffusion action of a counter-proposition is processed in parallel with all actions of the process. In this way, we impose no order of execution of this action in relation with the other actions.

Definition 4 [Coordination relationship sending]

Let $a = \mathbf{send}(id_A, msg)$ be an action where A an agent and $msg \in \{\mathbf{Facilitates}(t_A, t_B), \mathbf{Enables}(t_A, t_B), \mathbf{Hinders}(t_A, t_B)\}$ is a coordination relationship. This action specifies the identifier of the agent receiver and the information to send.

$$\langle \theta,\, a; S,\, \sigma_{coord} \rangle \xrightarrow{a} \langle \theta,\, S,\, \sigma_{coord} \rangle$$

This rule says that the sending of a coordination relationship has no effect on the mental state nor on the coordination historic of the sending agent.

Definition 5 [Total acceptation broadcasting]

Let $a = \mathbf{broadcast}(ID_{Ag}, \mathbf{TotAccept}(p))$ be an action where Ag is a set of agents and $p \in \mathcal{L}$ a proposition. The following rule considers that the total acceptation broadcasting generates the update of the mental state and the coordination history of the sender agent by saving the retained proposition.

$$\langle \theta,\, a; S,\, (h_{gp},\, h_{lp},\, h_{ra})\rangle \xrightarrow{a} \langle \theta',\, S,\, (h'_{gp},\, h'_{lp},\, h'_{ra})\rangle$$

$(h'_{gp},\, h'_{lp},\, h'_{ra})$ is defined as follows:

$$(h'_{gp},\, h'_{lp},\, h'_{ra}) \overset{\text{def}}{=} \begin{cases} (\epsilon,\, h_{lp}, h_{ra}) & \text{if } p \in \mathcal{L}_{gp} \\ (h_{gp},\, \epsilon, h_{ra}) & \text{if } p \in \mathcal{L}_{lp} \\ (h_{gp},\, h_{lp},\, \epsilon) & \text{if } p \in \mathcal{L}_{ra} \end{cases}$$

θ' is defined as follows:

$$\theta' \overset{\text{def}}{=} \begin{cases} \theta[Ret_{gp}/p] & \text{if } p \in \mathcal{L}_{gp} \\ \theta[Ret_{lp}/p] & \text{if } p \in \mathcal{L}_{lp} \\ \theta[O_g/O_g \cup\ \{(t_2, t_1)\}] & \text{if } p = after(t_1, t_2) \end{cases}$$

The broadcasting of a total acceptance comes after a long coordination process. Such information is broadcasted by the initiator agent after receiving individual acceptances, with respect to a proposition, from all agents. Only the initiator agent of a coordination process can take a decision of a total acceptance. The rule 7 presents the formalization of such situation.

4.2 Reception rules

This section describes the behavior of an agent receiving a message. Analogous to the previous section we consider containing messages: propositions, acceptances or rejects. For each type of message, we define the transformation of the configuration after the reception of that message.

Definition 6 [Proposition reception]

Let $a = \mathbf{receive}(id_A, \mathbf{Propose}(p))$ be an action where A is an agent and p a proposition in $\mathcal{L}$. The following rule says that if an agent receives a proposition, then it will evaluate it. This agent must note the reception of this message by adding to its history the proposition (p), the kind of action (rcv) and the sender agent.

$$\langle \theta,\, a; S,\, (h_{gp},\, h_{lp},\, h_{ra})\rangle \xrightarrow{a} \langle \theta,\, \mathbf{evaluate}(id_A,\, p); \mathbf{skip} || S,\, (h'_{gp},\, h'_{lp},\, h'_{ra})\rangle$$

$(h'_{gp},\, h'_{lp},\, h'_{ra})$ is defined as follows:

$$(h'_{gp},\, h'_{lp},\, h'_{ra}) \overset{\text{def}}{=} \begin{cases} (h_{gp}.\langle p,\, rcv,\, prop\rangle,\, h_{lp},\, h_{ra}) & \text{if } p \in \mathcal{L}_{gp} \\ (h_{gp},\, h_{lp}.\langle p,\, rcv,\, prop\rangle, h_{ra}) & \text{if } p \in \mathcal{L}_{lp} \\ (h_{gp},\, h_{lp},\, h_{ra}.\langle p,\, rcv,\, prop\rangle) & \text{if } p \in \mathcal{L}_{ra} \end{cases}$$

The proposition evaluation is an internal action that allows to take decision about the acceptance of a proposition. Definition 12 presents a formalization of this behavior.

The transition rule of messages of the form $\mathbf{CounterPropose}(p)$ is analogous to the one of $\mathbf{Propose}(p)$.

Definition 7 [Acceptance reception]

Let $a = \mathbf{receive}(id_A, \mathbf{Accept}(p))$ be an action where A is an agent and p a proposition that belongs to $\mathcal{L}$. If an agent receives an acceptance to a proposition then three different behaviors are possible:

- If the proposition is the one of the receiver agent and the number of acceptances received is equal to $(n-2)$ (where n is the number of agents), then a total acceptance must be broadcasted.

$$\begin{array}{l}\langle \theta,\ a;S,\ \sigma_{coord}\rangle \xrightarrow{a} \langle \theta,\ a';\mathbf{skip}\|S,\ \sigma_{coord}\rangle \\ \quad \text{if } (\sigma_{coord} \downarrow (p, snd)) \neq \emptyset \quad \text{and} \quad \sigma_{coord} \downarrow (p, rcv, accept)) = n-2\end{array}$$

  where $a' = \mathbf{broadcast}(\{id_{A_1}, \ldots, id_{A_1}\}, \mathbf{TotAccept}(p))$
- If the proposition is the one of the receiving agent and the number of acceptances received is lower than $n-2$, then a trace of the received message will be kept in the coordination history.

$$\begin{array}{l}\langle (\theta, a;S),\ (h_{gp},\ h_{lp},\ h_{ra})\rangle \xrightarrow{a} \langle (\theta, S),\ (h'_{gp},\ h'_{lp},\ h'_{ra})\rangle \\ \quad \text{if } ((h_{gp},\ h_{lp},\ h_{ra}) \downarrow (p, snd)) \neq \emptyset \\ \quad \text{and} \quad (h_{gp},\ h_{lp},\ h_{ra}) \downarrow (p, rcv, accept)) < n-2\end{array}$$

  $(h'_{gp},\ h'_{lp},\ h'_{ra})$ is defined as:

$$(h'_{gp},\ h'_{lp},\ h'_{ra}) \overset{\text{def}}{=} \begin{cases} (h_{gp}.\langle p,\ rcv, accept\rangle,\ h_{lp},\ h_{ra}) & \text{if } p \in \mathcal{L}_{gp} \\ (h_{gp},\ h_{lp}.\langle p,\ rcv, accept\rangle,\ h_{ra}) & \text{if } p \in \mathcal{L}_{lp} \\ (h_{gp},\ h_{lp},\ h_{ra}.\langle p,\ rcv, accept\rangle) & \text{if } p \in \mathcal{L}_{ra} \end{cases}$$

- If the proposition is not the one of the agent receiving then no modification will be made on the agent's mental state.

$$\langle (\theta, a;S),\ \sigma_{coord}\rangle \xrightarrow{a} \langle (\theta, S),\ \sigma_{coord}\rangle \quad \text{if} \quad (\sigma_{coord} \downarrow (p, snd)) \ = \ \emptyset$$

Definition 8 [Reject reception]

Let $a = \mathbf{receive}(id_A, \mathbf{Reject}(p))$ be an action where A is an agent and p is a proposition that belongs to $\mathcal{L}$. The following rule affirms that if an agent receives a reject to a proposition, this proposition will be deleted from its coordination history.

$$\langle (\theta, a;S),\ (h_{gp},\ h_{lp},\ h_{ra})\rangle \xrightarrow{a} \langle \theta, S,\ (h'_{gp},\ h'_{lp},\ h'_{ra})\rangle$$

$(h'_{gp},\ h'_{lp},\ h'_{ra})$ is defined as:

$$(h'_{gp},\ h'_{lp},\ h'_{ra}) \overset{\text{def}}{=} \begin{cases} (delete(h_{gp}, \langle p,\ snd,\ prop\rangle),\ h_{lp}, h_{ra}) & \text{if } p \in \mathcal{L}_{gp} \\ (h_{gp},\ delete(h_{lp}, \langle p,\ snd,\ prop\rangle), h_{ra}) & \text{if } p \in \mathcal{L}_{lp} \\ (h_{gp},\ h_{lp},\ delete(h_{ra}, \langle p,\ snd,\ prop\rangle)) & \text{if } p \in \mathcal{L}_{ra} \end{cases}$$

Definition 9 [Reception of total acceptance]

Let $a = \mathbf{receive}(id_A, \mathbf{TotAccept}(p))$ be an action where A an agent and p a proposition that belongs to $\mathcal{L}$. The following rule proves that if an agent receives a total acceptance for a proposition, it keeps this last in its mental state.

$$\langle (\theta, a;S),\ (h_{gp},\ h_{lp},\ h_{ra})\rangle \xrightarrow{a} \langle (\theta', S),\ (h'_{gp},\ h'_{lp},\ h'_{ra})\rangle$$

where $(h'_{gp},\ h'_{lp},\ h'_{ra})$ and θ' are defined as in the definition 5.

Definition 10 [Coordination relationship reception]

Let $a = \mathbf{receive}(id_A, msg)$ be an action where A is an agent and $msg \in \{\ \mathbf{Facilitates}(t_A, t_B), \mathbf{Enables}(t_A, t_B), \mathbf{Hinders}(t_A, t_B)\}$ a coordination relationship. We consider two cases:

- if the received message is a *hard coordination relationship* ($\mathbf{enables}(t_A, t_B)$) then the updating of the mental state is required. It concerns only the local agenda O_l if t_A and t_B are both tasks of the current agent, otherwise it concerns the O_g.

$$\langle(\theta, a; S),\ (h_{gp},\ h_{lp},\ h_{ra})\rangle \xrightarrow{a} \langle(\theta', S),\ (h_{gp},\ h_{lp},\ h_{ra})\rangle$$

  θ' is defined as follows:

$$\theta' \overset{\text{def}}{=} \begin{cases} \theta[O_l/O_l \cup\ \{(t_A, t_B)\}]\ if\ t_A, t_B \in\ T_A \\ \theta[O_g/O_g \cup\ \{(t_A, t_B)\}]\ if\ t_A \notin T_A\ or\ t_B \notin T_A \end{cases}$$

- if the receive message is a *soft coordination relationship*, then the updating of the local agenda O_l or the global agenda O_g is made if the proposed order is coherent with the mental state of the receiving agent.

$$\langle(\theta, a; S),\ (h_{gp},\ h_{lp},\ h_{ra})\rangle \xrightarrow{a} \langle(\theta', S),\ (h_{gp},\ h_{lp},\ h_{ra})\rangle$$

  θ' is defined as follows:

$$\theta' \overset{\text{def}}{=} \begin{cases} \theta[O_l/O_l \cup\ \{(t_A, t_B)\}]\ if\ t_A, t_B \in\ T_A\ and\ (t_B, t_A) \notin O_l \\ \theta[O_g/O_g \cup\ \{(t_A, t_B)\}]\ if\ (t_A \notin T_A\ or\ t_B \notin T_A)\ and\ (t_B, t_A) \notin O_g \end{cases}$$

Definition 11 [Reception of the tasks order]
Let a ba an action of the form $a = \mathbf{receive}(id_A, \mathbf{OrderlyTask}(t_A, t_B))$ where A is an agent. The following rule proves that the broadcasting of this information involves the updating of *Got* for the mental state of the receiving agent which owns the tasks t_A or t_B.

$$\langle\theta,\ a; S,\ (h_{gp},\ h_{lp},\ h_{ra})\rangle \xrightarrow{a} \langle\theta',\ S,\ (h_{gp},\ h_{lp},\ h_{ra})\rangle$$

θ' is defined as the following: $\theta' \overset{\text{def}}{=}\ \ \theta[O_g/\ O_g \cup\ \{(t_A, t_B)\}]$

4.3 Evaluation rule

Definition 12 [Evaluation rule]
The evaluation of a proposition has for consequence either its acceptance or its reject. This decision is taken according to a boolean value returned by an evaluation function *eval* which belongs to the agent's capabilities. We consider the actions of the form: $a = \mathbf{evaluate}(id_A,\ p)$ where p a proposition that belongs to $\mathcal{L}$ and A an agent. The following rule states that the evaluation action execution adds to the coordination process a broadcasting of the agent decision.

$$\langle\theta,\ a; S,\ \sigma_{coord}\rangle \xrightarrow{a} \langle\theta,\ \mathbf{send}(id_A,\ msg'); \mathbf{skip} \| S,\ \sigma_{coord}\rangle$$

msg' is defined as follows:

$$msg' \overset{\text{def}}{=} \begin{cases} \mathbf{Accept}(p) & \text{if}\quad eval(\theta,\ p) = \mathbf{true} \\ \mathbf{Reject}(p) & \text{otherwise} \end{cases}$$

5 Conclusion

The work presented in this paper could be considered as a first step for the development of a theoretically well-founded methodology for modular design of multi-agent systems. To develop such a methodology, we distinguish two major stages. The first stage consists in defining a suitable programming language containing all prominent concepts characterizing multi-agent systems. In this phase, the dynamic behaviour of the language is developed in terms of operational semantics. The second stage uses this operational characterization as a basis for a modular description of a MAS. This kind of description facilitates the analysis and the verification of desired properties in a multi-agent systems [EJT99].

References

[BF95] M. Barbuceanu and M.S. Fox. A language for describing coordination in **MAS**. In *International Conference on Multi-Agent Systems*, 1995.

[BK84] J.A. Bergstra and J.W. Klop. Process algebra for synchronous communication. *Information and Control*, (60):109–137, 1984.

[CHJ01] W. Chainbi, A. Ben Hamadou, and M. Jmaiel. A belief-goal-role theory for multiagent systems. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(3):1–16, 2001.

[Dec87] K. Decker. Distributed problem-solving techniques : A survey. *IEEE transactions on systems, man, and cybernetics*, 17(5):729–740, September/October 1987.

[DL89] E. H. Durfee and V. R. Lesser. Using partial global plans to coordinate distributed problem solvers. *IEEE Knowledge and Data engineering*, 1989.

[DL94] Keith S. Decker and Victor R. Lesser. Communication in the service of coordination. 1994.

[EJT99] J. Engelfriet, C. M. Jonker, and J. Treur. Compositional verification of multi-agentsystems in temporal multi-epistemic logic. In *Proceedings of fifth International Workshop on Agent Theories, Architectures, and Languages (ATAL98)*, volume 1544 of *Lecture Notes in Artificial Intelligent*, Heidelberg, 1999.

[FF94] T. Finin and R. Fritzson. **KQML** : a language and protocol for knowledge and information exchange. In *Proceedings of the Thirteenth International Workshop on Distributed Artificial Intelligence*, pages 126–136, Lake Quinalt, WA, July 1994.

[FIP97] FIPA. Agent communication language. In *Specification FIPA*, 1997.

[HKJ99] A. Hadj-Kacem and M. Jmaiel. Towards a formal definition of the cooperation among multiagent systems. In *Proceedings of the first IFAC Workshop on Multi-Agent-Systems in production*, Vienna, Austria, December, 2-4 1999.

[HKJ01] A. Hadj-Kacem and M. Jmaiel. A formal negociation model for cooperating agents. In *Proceedings of the AAAI Workshop on Negotiation Methods for Autonomous Cooperative Systems*, North Falmouth, Massachusetts, USA, November 2001.

[HNJ96] L. Lee H. Nwana and N. Jennings. Coordination in software agent systems. *BT Technology Journal*, 1996.

[Hoa85] C.A.R. Hoare. *Communicating Sequential Processes*. Prentice Hall International, 1985.

[Mil80] R. Milner. *A Calculus for Communication Systems*, volume 90 of *Lecture Notes in Computer Science*. Springer, Berlin, 1980.

[Plo81] G. D. Plotkin. A Structural Approach to Operational Semantics. Technical Report DAIMI FN-19, University of Aarhus, 1981.

[Sho93] Y. Shoham. Agent-oriented programming. *Journal of Artificial intelligence*, 60:51–92, 1993.

[SSJ97] Onn Shehory, Katia P. Sycara, and Somesh Jha. Multi-agent coordination through coalition formation. In *Agent Theories, Architectures, and Languages*, pages 143–154, 1997.

[YHH$^+$98] J. Yang, R. Havaldar, V. Honavar, L. Miller, and J. Wong. Coordination of distributed knowledge networks using contract net protocol, 1998.