



HAL
open science

Supporting Virtual Enterprise Systems Using Agent Coordination

Nacereddine Zarour, Mahmoud Boufaïda, Lionel Seinturier, Pascal Estraillier

► **To cite this version:**

Nacereddine Zarour, Mahmoud Boufaïda, Lionel Seinturier, Pascal Estraillier. Supporting Virtual Enterprise Systems Using Agent Coordination. Knowledge and Information Systems (KAIS), 2005, 8, pp.330-349. 10.1007/s10115-004-0183-4 . inria-00485957

HAL Id: inria-00485957

<https://inria.hal.science/inria-00485957>

Submitted on 25 May 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Supporting Virtual Enterprise Systems Using Agent Coordination

N. Zarour¹, M. Boufaida¹, L. Seinturier² and P. Estraillier³

¹University of Constantine, Computer Science Department, Laboratory LIRE, Constantine,
25000, Algeria

²University Pierre et Marie Curie, Laboratory LIP6, Paris, 75005, France

³University of La Rochelle, Laboratory L3i, La Rochelle, 17042, France

Abstract. Open environments like the Internet or corporate intranets enable a large number of interested enterprises to access, filter, process, and present information on an as-needed basis. These environments support modern applications, such as virtual enterprises and inter-organizational workflow management systems, which involve a number of heterogeneous resources, services, and processes. However, any execution of a virtual enterprise system would yield to disjoining and error-prone behavior without appropriate techniques to coordinate the various business processes. This paper reports on the design and implementation of a flexible agent-based framework for supporting the coordination of virtual enterprises and workflow management systems. The paper also shows how an agent coordination infrastructure, which is explained by social constraints, can impact on the engineering of highly dynamic virtual enterprises and workflow management systems by presenting a simple case study.

Keywords: Virtual enterprise; agent; coordination technologies; social constraints; workflow management system.

1. Introduction

Over the past few years Multi-Agent Systems (MAS) have been perceived as a crucial technology for building large, complex, and robust distributed information processing systems which exploit the efficiencies of organized behaviors. Agent societies, in particular, seem to be the most promising candidates for the development of the virtual enterprise management (Fischer et al, 1996).

Received May 25, 2003

Revised Nov 29, 2003

Accepted June 1, 2004

Virtual Enterprises (VE) are an application domain in which the MAS technology is appropriate since they include generic characteristics of complex applications (Lesser, 1999):

1. Distributed situation assessment, which emphasizes how agents, with different spheres of awareness and control, should share their local interpretations to arrive at consistent and comprehensive explanations and responses.
2. Distributed resource scheduling and planning, which emphasizes how agents should coordinate their schedules to avoid and resolve conflicts over resources, and to maximize system output.
3. Distributed expert systems, which emphasize how agents share information and negotiate over collective solutions with their different expertise and solution criteria.

The advantages of using the agent paradigm within VE applications are its autonomy and abilities to perceive, reason, and act in their environments, as well as to socially interact and communicate with other agents (Huhns and Singh, 1998). As part of MAS, agents can capture and apply the semantic constraints among heterogeneous components in order to enact distributed workflows. Moreover, MAS are well adapted to interactive machines where the exchange is exclusively done by interaction (e.g. sending of messages, Agent Communication Language). In our context, the characteristics of an agent that can be a workflow, person, an information system or any combination of them include autonomy, social ability, and pro-activity (Zarour et al, 2000a). Our agents are also informational since they perform the role of managing, manipulating or collecting information from many distributed sources.

However, VE systems are made of components that are autonomous, heterogeneous, distributed, cooperative, and communicating. Moreover, they use shared and dynamic environments where resources are limited. Therefore, some problems may occur like the lack of a global state due to asynchronous communications and the management of incomplete information and partial vision of agents. In fact, the main issue is to conciliate between the local autonomy of an agent and the global coherence of the system. We believe that this conciliation would be reached only by building appropriated coordination models.

Hence, we define a VE as a temporary aggregation of autonomous and possibly heterogeneous enterprises, meant to provide the adaptability to frequent organizational and technical changes, and the flexibility that characterize both the openness of service activity and business process scenarios. The heterogeneity term means that the involved information resources provided by different participants are developed under different data modeling formalisms and platforms. On the other hand, coordination strategies enable groups of agents to solve problems effectively through decisions about which agents should perform specific tasks and when, and to whom they should communicate the results (Lesser, 1999). The potential complexity involved in making these decisions can be seen in the simple situation where one agent needs the results of a sub-problem that another agent is solving. Let us assume that the producing agent has other tasks to do with their own deadlines in addition to producing a result for the other agent. To further complicate this decision process, the producing agent may have alternative methods for doing those tasks that trade of the quality of the task solution against the time to complete the task. An additional complexity is introduced when neither the time that a method takes nor the quality of its results are

known precisely that can rather be described by a statistical distribution. Another complexity of the coordination mechanism is the reactive planning. A new event occurring in the environment may cause a reorganization of the considered system. The participating components must be able to re-plan their course of actions if necessary.

Thus, the dependencies among autonomous and heterogeneous VE's activities concern the sharing and the exchange of heterogeneous information resources, the task assignment for business processes, as well as temporal and prerequisite constraints among heterogeneous activities (Licci et al, 2001). Hence, in order to reach VE global goals, these dependencies must be rationally managed and governed. This rationality can be obtained only if the main above issues can be understood as agent coordination problems (Malone and Crowstone, 1994).

With the aim to build autonomous agents that work coordinately in a dynamically changing world, the proposed approach tries to answer two simple questions. The first one is how an agent chooses a particular course of actions and how its choices change in front of events happening in the world. The second question is how agents execute coordinated actions. Actually, the answer to these questions consists in constructing agents as rational decision makers that exist within organizations. We consider an organization as a system that constrains the actions of member agents by imposing mutual obligations and interdictions. The association of obligations and interdictions is mediated by the roles that agents play in a VE. Neither fulfilling an obligation nor interdiction is sanctioned by paying a cost or by a loss of utility. Accordingly, the interpretation of workflow processes as coordination ones can be explained by the social constraints (social laws) (Barbuceanu, 1997) that agents in VE are subject to. Once a goal is adopted, a set of plans, called conversation plans, is available to agents for effectively carrying out a coordination action. The conversation plans explicitly represent interactions by message exchange and their actions are dynamically reordered using the theory of MDP (Markov Decision Process) (Bellman, 1957) to ensure the optimization of various criteria.

The objective of this paper is to build a flexible and formal framework for engineering the coordination requirements of VE. This framework provides a greater flexibility than traditional systems because it should be possible to cancel or otherwise modify the plans. Another advantage of this framework is to preserve the agent autonomy, i.e. how to maximize agent freedom without letting the systems evolve into chaos. The management of this autonomy lies in social constraints like obligations, interdictions, and permissions. The proposed framework is based on the extension of the social constraint entities by adding some concepts and mechanisms like assignation, delegation, branch alternatives, and allowing agents to carry out several conversations in parallel.

The rest of the paper is organized as follows. Section 2 describes the required infrastructure for VE, it shows how the global coordination mechanism helps to model VE and Workflow Management Systems (WfMS). A discussion of the proposed coordination based framework is provided in section 3. Section 4 describes the implementation of the prototype using an application in the manufacturing domain. In section 5, we present some related works. Section 6 summarizes the contributions of this paper and gives some directions for future research.

2. An Infrastructure for VE

VE typically requires combining the core competencies of several autonomous and heterogeneous enterprises in a new agile and flexible enterprise, addressing specific and original industrial business targets (Rocha and Oliveira, 1999). In order to reach these targets, we have developed an agent-based architecture, named DAARACHE¹ (Zarour et al, 2000a), supporting a VE environment. In this architecture an agent can be an information system, a workflow, a human or any combination of them depending on the level of automation. In this paper, the proposed coordination mechanism is implemented as a service in this environment. It permits to achieve the coordinated behavior of a VE.

2.1. VE, WfMS, and Agent Coordination

In our approach, we consider two types of agents: the leader agent that receives a customer request and accepts to achieve it, and the member one that agrees to cooperate after being asked by a leader agent indicating the capability it is interested in. The infrastructure supporting VE management has to address two main issues:

1. the valuation of achieving a global goal, and
2. the execution of VE specific business processes.

In the first issue, the communication is only done between agents. The leader agent must fully recognize the goals, characteristics, and principles of business system in order to elaborate the strategic planning of the global activity (Fig. 1.a). The methodology of this planning is inspired from the (Brumec et al, 2000) one. The result of applying this methodology should contain the business processes, business data, technical resources, and development activity plan. This planning methodology is used in our coordination infrastructure because it can be easily inspected, elaborated, and possibly changed dynamically by (intelligent, human or artificial) agents. Then, the leader agent negotiates with other agents that attempt to satisfy some sub-goals with some attributes whose values model the quality of service. Once the best offer from each agent is given, the leader agent evaluates whether the global goal could be achieved or not according to the required needs in terms of costs, delivery time, and other performances.

In the second issue, the communication is carried out in two ways. In the first way, the communication is only established between agents while in the second one, it is done between agents and their underlying physical entities. These two modes of communication are more described in (Zarour et al, 2000b). The leader agent supervises the coordinated tasks among the VE members by taking into account agents' commitments that have been established during the first issue. An agent that is a member of a VE can collaborate with other agents and play in this case the role of the leader, since it must ensure at least the coordination between these agents. Consequently, the interaction model between agents is a peer-to-peer one since it provides agent uncoupling from both the

¹ Distributed Agent-based Architecture for Autonomous Cooperating Heterogeneous Enterprises.

space and time viewpoints. The peer-to-peer model allows to go beyond point-to-point communication protocols, supporting location-independent disconnected workflow participant.

2.2. Global and Local Coordinating Rules

The coordination mechanism is composed of two parts: the global coordinating rules (Fig. 1.a) that perform the supervision of an organization global behavior, and the local coordinating rules (Fig. 1.b) that perform the supervision of tasks which are locally executed by an agent according to an organization global behavior. Both types of rules are represented using workflows (Fig. 1). A WfMS permits to improve the process throughput, promote a better use of resources, and enable efficient process tracking (Santanu et al, 1997). Moreover, since workflow applications are subject to frequent changes due to the business environments, flexibility and adaptability are key features to face the necessary dynamic evaluation of coordination policies, and to react suitably to unpredicted situations.

In Fig. 1, the workflows are represented using a derived form of Petri nets like MCT (Task Conceptual Model) of the MERISE method (Tardieu et al, 1985) where a place represents an event (extern, intermediary or result) and a transition represents an operation which is a sequence of uninterrupted actions. The pre-condition for executing an operation is expressed using the synchronization of input events. The post-condition produces events according to emission rules.

In Fig. 1.a, after having received the customer request, the leader interprets, decomposes, and schedules the global goal into sub-goals (elaboration of plan operation). Once the different roles are identified, the leader contacts several competent agents asking each of them to achieve one or more sub-goals (negotiation operation). Then, based on the best offer of each sub-goal, the leader tries to evaluate whether the global goal could be reached or not (evaluation operation). Whenever the customer agrees (after a possible negotiation with the leader), the leader assigns definitive roles to agents (members) that have presented the best offers. Once the execution of sub-goals starts, the leader must coordinate between distributed actions taking into account the agents' commitments (execution operation). If no problems occur, the global goal will be achieved according to initial constraints; else the leader must identify the nature of the problem and re-plan the course of actions (elaboration plan operation triggered by revision of planning event). The agent that has caused the problem is sanctioned.

In Fig. 1.b, an agent that agrees to carry out one or more sub-goals asked by the leader after negotiation (negotiation operation) becomes a member of the organization. Then, this member plans its course of actions (planning operation). During the execution of member's actions (execution operation), if no problems occur, the sub-goal will be achieved.

In the contrary, if the member has the intention to stop the execution (situation operation), the sub-goal will be assigned to another agent; else the member must re-plan its course of actions. In both cases, this member will be sanctioned by the leader. However, if the problem comes from the leader, this latter will be sanctioned too.

Now, we give an example that shows how a member plans its course of actions. A plan organizes a collection of actions which can be performed sequentially or concurrently in some specific order. Let us assume that the assembler agent plan

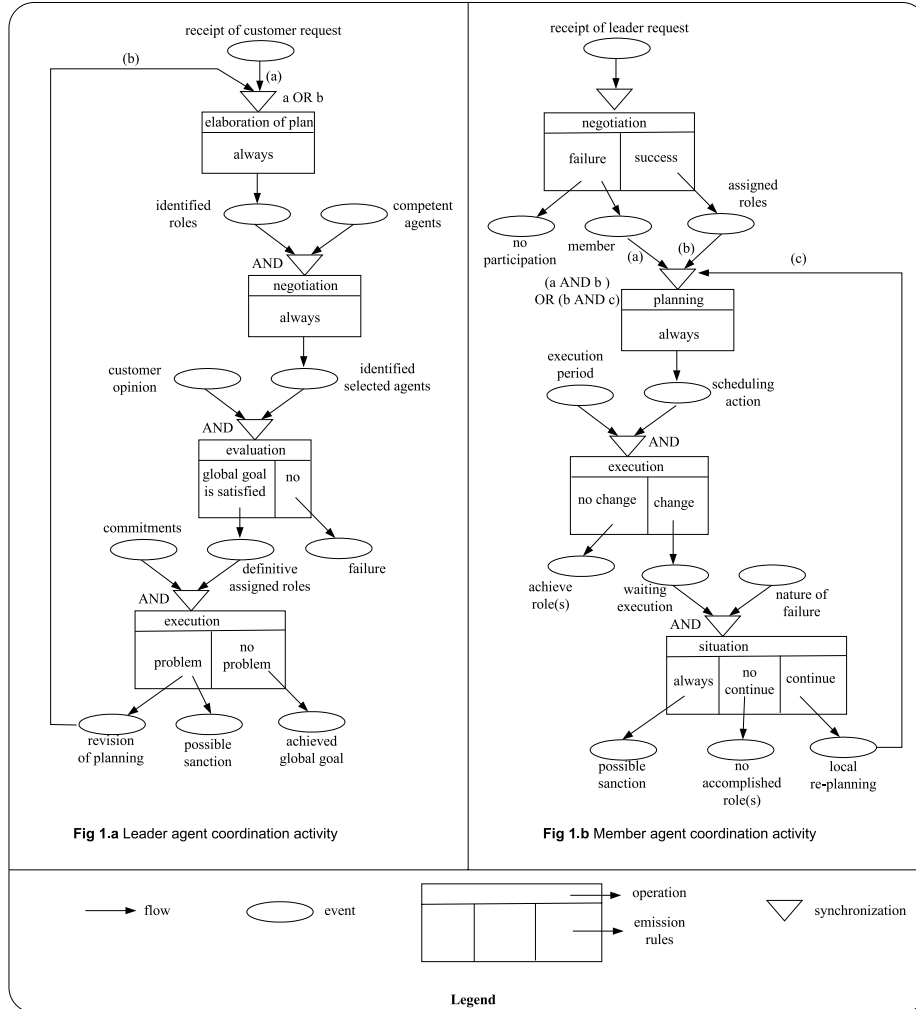


Fig. 1. Interactions between an organization leader and members

(Fig. 2) contains four operations. The *pre-assembly* one consists of assembling available pieces. The *assembly1* and *assembly2* operations are concurrently executed. *Assembly1* operation assembles a part of the pre-assembly result (*part1* event) with the complementary pieces (*pieces-X* event) that are manufactured by the agent *X*.

The same reasoning is applied for *assembly2* operation. Then *final-assembly* operation is triggered by the synchronization of the two events *assembly1-done* and *assembly2-done* (produced respectively by operations *assembly1* and *assembly2*) and the complementary pieces (*pieces-Z* event) that are manufactured by the agent *Z*.

In VE applications, WfMS also have to face the issue of distribution, having to coordinate heterogeneous activities spread over the network. In the next section, we will show how to represent workflow rules as agent coordination laws.

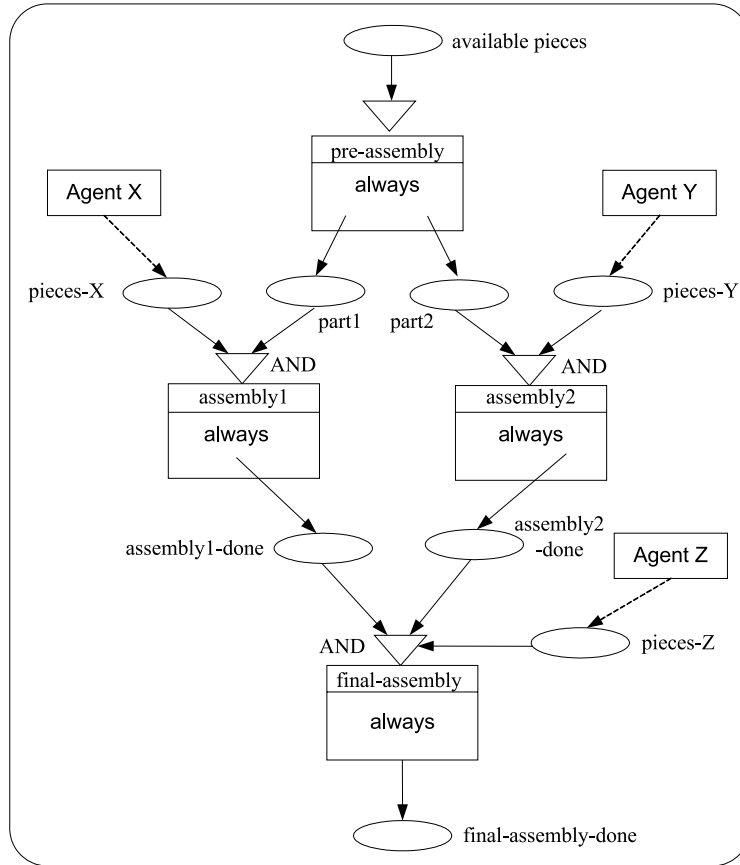


Fig. 2. The assembler agent plan

3. Social Constraints in the Coordination Mechanism

Research on social laws in computational environments has proved the usefulness of the law-based approach for the coordination of MAS (Fitoussi and Tennenholtz, 1998). An artificial social system institutes a social law that the agents shall obey (Shoham and Tennenholtz, 1992) (Minsky, 1991). Consequently, we explore the view that the coordinated behavior is driven by the social constraints that agents in VE are subject to. In the following, we overview the social constraints entities, then we discuss how such a coordination based framework could be fruitfully exploited in the context of VE's WfMS.

3.1. Social Constraints Overview

Social laws are objective forces that provide the ultimate motivation for coordinated actions at the organizational level and to a large extent determine the mental states at the individual agent level. Agent desires and intentions are con-

cepts that are requested by their current obligations, knowing that otherwise there will be a cost to pay. In fact, these sanctions allow agents to apply rational decision making when choosing what to do.

The social constraints are made of agents, organizations, roles, obligations, interdictions, permissions, goals, constraints, plans, etc. Agents determine what obligations and interdictions currently apply and on this basis decide on their goals. Once an agent has chosen a goal, it selects a plan to be executed. A conversation plan, which is described in a planning formalism, is a description of how an agent acts and interacts in certain situations. It consists of states and rule governed transitions together with a control mechanism and a local database that maintains the state of the conversations. These conversations can be mapped to fully-observable, discrete-state MDP. In this mapping, the conversation states correspond also to states and the conversation rules become actions.

The vocabulary of social constraints is described as follows: an organization consists of a set of roles fulfilled by a number of agents. Each agent has its local store of beliefs which is taken as a database rather than mental states. A role describes a major function together with the obligations, interdictions, and permissions attached to it. An agent $A1$ in role $R1$ has an obligation towards an agent $A2$ in role $R2$ for achieving a goal G with respect to some constraints C if and only if non-performance by $A1$ of the required actions allows $A2$ to apply a sanction to $A1$. In the same way, the interdiction (the performance of the goal is sanctioned) and permission (neither the performance nor non-performance are sanctioned) are defined. Semantically, interdictions, obligations, and permissions are modeled using the reduction of deontic logic to dynamic logic due to (Mayer, 1988).

In our context, the central notion of the deontic logic will be represented by a modal operator $[\alpha]$ associated with an action α . The expression $[\alpha] \Phi$ means that the precondition is required to ensure that the proposition Φ will hold after α has been done. We respectively define interdiction, obligation, and permission as follows:

$F^{ij} \alpha \equiv [\alpha]^i V_\alpha^{ij}$ which means that agent i is forbidden by agent j to execute α ;

$O^{ij} \alpha \equiv F^{ij}(\text{not } \alpha)$ which means that agent i is forbidden by agent j to do $\text{not } \alpha$, i.e. agent i is obliged by agent j to execute α ;

$P^{ij} \alpha \equiv \text{not } F^{ij} \alpha$ which means that agent i is permitted by agent j to execute α ;

where V_α^{ij} denotes a violation by agent i of a constraint imposed by agent j with respect to α (associated with a cost to pay).

The reduction of deontic logic generates a number of theorems which permit to efficiently reason about the obligation, interdiction, and permission constraints in action networks. The main theorems that are necessary in real applications are as follows (indices are dropped for clarity) where ";" denotes sequential composition, " \cup " non-deterministic choice, and "&" parallel composition of actions.

$$\models F(\alpha; \beta) \equiv [\alpha] F\beta \quad (1)$$

a sequence is forbidden iff after executing the first action, the remaining subsequence is forbidden.

$$\models F(\alpha \cup \beta) \equiv F\alpha \wedge F\beta \quad (2)$$

a choice is forbidden iff all components are also forbidden.

$$\models F(\alpha \vee F\beta) \supset F(\alpha \& \beta) \quad (3)$$

if at least one component of a parallel composition is forbidden, the parallel composition is forbidden as well.

$$\models O(\alpha; \beta) \equiv (O\alpha \wedge [\alpha] O\beta) \quad (4)$$

a sequence is obliged iff the first action is obliged and after executing it the remaining subsequence is obliged as well.

$$\models (O\alpha \vee O\beta) \supset O(\alpha \cup \beta) \quad (5)$$

if at least one component of a choice is obliged, the choice is also obliged.

$$\models O(\alpha \& \beta) \equiv (O\alpha \wedge O\beta) \quad (6)$$

a parallel composition is obliged iff all components are obliged.

$$\models P(\alpha; \beta) \equiv \langle \alpha \rangle P\beta \quad (7)$$

a sequence is permitted iff there is a way to execute the first action after which the remaining subsequence is permitted.

$$\models P(\alpha \cup \beta) \equiv (P\alpha \vee P\beta) \quad (8)$$

a choice is permitted iff at least one component of it is permitted.

$$\models P(\alpha \& \beta) \supset (P\alpha \wedge P\beta) \quad (9)$$

if a parallel composition is permitted, then all components must be permitted.

$$\models O(\alpha \cup \beta) \wedge F\alpha \wedge P\beta \supset O\beta \quad (10)$$

if a choice is obliged and one component is forbidden while the other is permitted, then the permitted component is obliged.

The use of a formal semantics of the three constraints (obligation, interdiction, and permission) allows us to verify that the coordination specifications possess correctness properties, namely that the responsibilities assigned to roles are fulfilled, and that constraints are maintained as a result of the coordination process execution.

3.2. SoCCoF: A Coordination Framework in VE

In this section, we present a framework, called SoCCoF², which is both a model and an infrastructure for the coordination of cooperative agents. In order to improve the performance of the global coordinated behavior, an agent response must not exceed the required maximum reply time. Also, this framework may be used when two agents share the same resources and allow an agent to carry out several conversations in parallel without leading the MAS to inconsistent states. In addition, the basic concepts that we use are the *alternative branches*, *subcontracting*, and *commitments*.

² Social Constraints-based Coordination Framework.

3.2.1. Mechanisms of the Framework

Some mechanisms are considered in SoCCoF. The most important one is to keep the MAS in a consistent and valid state in spite of allowing agents to carry out several conversations in parallel. A hierarchical organization of conversations allows parent conversations to control the execution of their children ones. This hierarchical organization forces agents to consult their chronological account of commitments to avoid eventual contradictions (e.g. overlap of commitments), and to maintain the consistency of the MAS. In fact, the various conversations carried out by an agent are not independent and the negotiated solutions are constrained by each other.

How does the framework solve the inconsistency problem?

In order to avoid contradictions, an agent may store the chronological account of its commitments in its local database. Whenever a contractor has to take a decision, it must first consult its historical commitments. After dealing with some competent contractors for executing a task, the leader classifies them according to their offer. The task will be assigned to the contractor that has presented the best offer.

Now we give some details that show how a contractor reasons about its commitments to make new decisions. Let us consider the following rules: a contractor may lead several conversations in parallel. It negotiates with one party not with a third one. We are only interested in contractors' commitments (not to all conversations) which would lead to undesirable situations. We assume that a task is elementary, i.e. it is carried out by one (business) process. A task is characterized by a beginning date and a length of its execution.

The formal structure of these commitments is composed of the obligations to which we add some attributes that characterize each obligation. The structure of a commitment is defined by the 4-tuple:

$$C(Obl, P, D, L)$$

where $C = \{c_h\}$ is the set of a contractor commitments;

$Obl = \{obl_{ij\alpha}^k\}$ means that the agent i is obliged by the agent j to execute an action α in an organization k ;

$P = \{p_m\}$ is the set of a contractor processes;

$D = \{d_n\}$ is the set of the beginning dates of task executions;

$L = \{l_q\}$ is the set of the lengths of task executions.

The formal presentation of commitments described above allows agents to manage several conversations in parallel without leading the MAS to inconsistent states.

How does the framework face failures?

The coordination processes will terminate even if there is a failure within the system. Before bargaining, a leader must consult the contractor perception interface (Zarour et al, 2003) to verify that it is operational. Whenever a contractor withdraws before execution process and cannot inform the leader, this latter will detect that once the contractor has exceeded the time out, and therefore, the process with this contractor will terminate. In this case, in order to replace the abandoned agent by another one without doing a negotiation process again, the

leader may consult the contractor classification to select another agent. The second case is when a member fails during a task execution; the leader will detect the failure in the same way as the previous one. Then the leader will re-negotiate with other agents to select one and will take into account the propagation of delay by re-planning the different tasks.

3.2.2. Concepts of the Framework

The concept of alternative branches allows two agents to explore several possibilities to find different solutions for a same problem. For example, an agent A asks an agent B if it is capable of achieving a goal G with constraints C and optimizations O , otherwise it gives some counter-proposals (other constraints and/or optimizations). The agent B may replies in three possible ways:

1. it cannot achieve the goal G ,
2. it can achieve the goal G with the required constraints C and the required or not optimizations O , and
3. it can achieve the goal G but with other constraints C' and the required or not optimizations O .

We have constrained the number of conversations between two agents to secure the termination of the process and reduce its length.

The concept of subcontracting allows a VE member to subcontract with other agents, and thus, becomes itself a leader. In this case, the leader agent ignores the existence of such transactions.

The major actions on commitments are the creation, assignation, delegation, and cancellation.

- The *creation* is an instantiation of an obligation that links an agent to its leader. The server performs a creation only if the leader agrees with its proposition. Once the creation has been done, the server becomes a member of the VE and may have one or more roles.
- The *assignation* allows a leader to assign another agent when it got held up. The assigned agent must continue to respect the ex-leader's commitments that have been held with the VE members and ensure the coordinated actions among these members during the execution stage.
- The *delegation* permits a VE member that has had a better offer or a failure, to delegate another agent to play its role in the organization. The delegated agent must respect the ex-member's commitments, i.e. the delegated agent will be sanctioned if some obligations or interdictions are not respected.
- The *cancellation* allows an agent to remove its commitment into an organization by releasing the appropriate instance of obligation.

The way that an agent (leader or member) may pay the "decommitment" penalty is not detailed here; however it is inspired from (Exelente-Toledo et al, 2001) one which introduces the notion of *variable penalty contracts*.

The mechanisms and concepts that are considered in the proposed framework are expressed using different social constraints as it will be shown in the next section. By describing the obligations and interdictions as relations among organizational roles, we use them to shape the social behavior. The internal representation of each agent and an engine for reasoning about their obligations and interdictions allow effectively integrating these entities with the agent's beliefs, goals,

and plans, extending the BDI (Beliefs, Desires, and Intentions) model (Parsons et al, 1998) in a new direction. Indeed, these concepts and mechanisms allow SoC-CoF to provide the flexibility and adaptability required to face rapid changes in business environments, as well as to react to unpredicted situations. On the other hand, they lead to an improvement of the VE behavior and performance.

4. Applications in Manufacturing

The modern manufacturing that is naturally distributed deals with information management in open environments. It involves a large number of autonomous commercial entities with a variety of heterogeneous information systems. Furthermore, it makes use of human decision making and faces the realities of failure and exception in physical processes and commitments. Thus, the requirements for the control of coordinated processes in manufacturing applications correspond well to our motivations.

Actually, the management and manipulation of business data is a difficult task for many developers because data can take many forms. It can be a part of a corporate document, a message from another system, an interface to a transaction system, a database record, or a document on a web site. The crucial missing element has been the ability to enable interchange between heterogeneous systems by adopting a universal interchange format that serves as the single output format for all exporting systems and the single input format for all importing systems. The XML (eXtensible Markup Language) technology can resolve the crucial problem of heterogeneity in VE since it helps developers to build and deploy sophisticated Web applications faster and allows the VE components to understand each other. XML's human readability is another significant advantage for VE.

We sketch a simple case study, where SoCCoF is exploited to support a "Mono-spindle horizontal lathes production" VE. A supply chain is a globally extended network of suppliers, factories, warehouses, distribution centers through which raw materials are acquired, transformed into products, delivered to customers, serviced and enhanced. The different local factories are seen as cells where each one, while operates to achieve its best advantages, cooperates with other cells to improve the overall performance of the whole system in terms of quality, costs, and response time. It is obvious that the key to efficient operations of such a VE is a tight coordination among components.

For the entire system to operate efficiently, each participant may apply the global or local workflow according to the leader or member role. Then, agents communicate directly by exchanging social constraints, which also support data integrity.

4.1. Exchanging Social Constraints Using XML

We give some scenarios from the application described above about exchanged social constraints. Indeed, these scenarios show how to benefit from XML technology in VE.

4.1.1. A VE Specification Example

Let us consider in Fig. 3 an example which specifies the result of the planning, negotiation, and evaluation operations of the leader activity coordination (global coordinating rules). First, the result of the planning operation is the set of goals obtained by the interpretation, decomposition, and scheduling of the global goal. Then the result of planning operation is used as an input for the negotiation one that consists of founding the best offer for each goal. Finally, the leader evaluates whether the global goal could be reached or not using the evaluation operation. Hence, the result of the third operation is to assign definitive roles to selected agents (see Fig. 3) if the evaluation is positive. Thus, this result expresses the VE global goal, whereby some constraints and optimizations are associated to it, and the set of roles played by a number of agents permitting to achieve the global goal.

In Fig. 3, the global goal of "mono-spindle horizontal lathes production" VE is to "produce 100 mono-spindle horizontal lathes". The associated constraints of this goal are "delay" and "dimensions". The execution of the goal may be optimized for "time" and "quality". To argue about whether constraints are satisfied or not, the theory *T1*, which consists of a set of Horn clauses (Angluin et al, 1990) and will be explained in the next section, can be used. Four agents ("studies", "specialized components", "treatments", and "assembly") playing each one some roles participate in this VE in which the "studies" agent is the leader.

4.1.2. An Alternative Branches Scenario

Let us consider an "alternative branches" scenario which is done during a negotiation operation (of Fig 1.a) and involves the leader and a number of competent agents. The goal to be achieved is to assemble mechanical systems. We simplify this scenario on considering the leader (*L*) and only one competent agent (*A*). Three stages are required to achieve this scenario:

1. *L* asks *A* to achieve a goal using a goal definition;
2. *A* replies in three ways: it cannot; it can; or it counter-proposes;
3. If *A* accepts or counter-proposes and *L* agrees, then *A* and *L* create each one an instance of obligation using an obligation definition.

The goal definition that is specified in 1 is presented in Fig. 4 as an XML document. In this example (Fig. 4), *L* asks *A* for achieving a goal that consists of assembling mechanical systems. The constraints that can be meaningfully satisfied to this goal are that the sub-unit mounting would be complete by bolting-screwing and the respect of the required maximum reply time.

Whatever plan is used for this goal, its execution may be optimized for time. Finally, to reason about whether constraints are satisfied or not, a theory "theories" can be used. A theory is a set of *Horn* clauses that can be applied to determine if constraints are satisfied.

For instance, suppose that the message from *L* requires a reply in at most 10 units of time, and "theories" has a clause:

$$(\text{satisfied-max-reply-time } ?t1 \ ?t2) \Leftarrow (\text{req-max-reply-time } ?t1) \text{ and } (\text{max-reply-time } ?t2) \text{ and } (\leq ?t2 \ ?t1)$$

```

<?xml version="1.0" encoding="us-ascii"?>
<Virtual-Enterprise name-VE="mono-spindle horizontal lathes production">
<goal name-goal="to produce 100 mono-spindle horizontal lathes ">
  <constr-goal>delay</constr-goal>
  <constr-goal>dimension</constr-goal>
  <optim-goal>time</optim-goal>
  <optim-goal>quality</optim-goal>
  <theories><rule rule1="shop trial">T1</rule></theories>
</goal>
<agent name-agent="studies cell">
  <role>engineering design</role>
  <role>mechanical design</role>
  <role>tooling design</role>
  <role>3D rapid prototyping</role>
</agent>
<agent name-agent="specialized components cell">
  <role>tool making</role>
  <role>injection moulding</role>
  <role>moulding casting</role>
  <role>engraving-marking</role>
</agent>
<agent name-agent="treatments cell">
  <role>mechanical treatment</role>
  <role>heat treatment</role>
  <role>chemical treatment</role>
</agent>
<agent name-agent="assembly of mechanical systems cell">
  <role>complete sub-unit mounting</role>
  <role>complete sub-unit mounting by welding</role>
  <role>complete sub-unit mounting by bolting-screwing</role>
  <role>complete sub-unit mounting by sticking</role>
</agent>
</Virtual-Enterprise>

```

Fig. 3. A VE specification using an XML document

```

<?xml version="1.0" encoding="us-ascii"?>
<goal name-goal="to assembly mechanical systems">
  <constr-goal>complete sub-unit mounting</constr-goal>
  <constr-goal>by bolting-screwing</constr-goal>
  <constr-goal>req-max-reply-time 10</constr-goal>
  <optim-goal>time</optim-goal>
  <theories>
    <rule rule1="satisfied-max-reply-time ?t1 t2
      (req-max-reply-time ?t1) and (max-reply-time ?t2)
      and (<= ?t2 ?t1)" />
  </theories>
</goal>

```

Fig. 4. A goal definition specifying a part of the alternative branches scenario using an XML document

4.1.3. A Delegation Scenario

Now, let us consider another scenario that describes the delegation concept during the execution operation (of Fig 1.b). We suppose that the "treatment Agent" (*A*) wants to interrupt the mechanical treatment role (due to a failure) and delegate another agent (*D*) to continue the process. Therefore, this delegation scenario goes through four stages:

1. *A* asks *D* if it accepts to delegate it, this request is expressed using the goal definition;
2. if *D* accepts the delegation principle, then *A* sends to *D* a message specifying what to do;
3. if *D* accepts this delegation, it creates an instance of obligation using the obligation definition which ties it to *A*, and finally;
4. *A* informs its leader about this delegation.

First, let us consider an XML document (Fig. 5), which specifies the delegation content (second stage of the delegation scenario) that *A* (treatment agent) sends to *D*. In the example of Fig. 5, *A* plays the mechanical treatment role while agent *D* plays the delegate mechanical treatment one. *A* describes the delegation detail using an XML document to send it to *D*. The delegation object is that *D* will continue to treat horizontal lathes according to some features. For instance, the model must be two pilot axes and the due date is June, 25th 2003. *D* does not exceed five units of time to reply. Finally, the execution of the delegation may be optimized for accuracy and/or time.

We assume that *D* accepts the delegation content, so it must create an instance of obligation (third stage of the delegation scenario) that forces it to achieve the delegation goal. This obligation definition is represented as an XML document in Fig. 6.

The XML documents presented in Fig. 3, 4, 5, and 6 are well-formed because they respect rules defined by the XML specification and are valid because they are conform to their XML-schemas (which are not represented here).

On the one hand, we have shown how the used concepts in the framework are exploited during the valuation and execution stages of the coordination mechanism and expressed using the social constraint entities. On the other hand, we have shown how to benefit from XML when exchanging messages during coordination processes in VE. However, XML presents some limits that are inherited from the document field, such as the imposed order and the different possible representations with attributes.

4.2. A Prototype Implementation

To show the feasibility of SoCCoF, we have implemented a prototype using two standards, XML and CORBA. This prototype provides facilities to manage intra and inter-VE conversations and execute business processes. It runs with JacORB 1.2 that is a CORBA 2.4 ORB with a Java language mapping, JDK 1.3, and XML.

In order to provide services or products to other participants, the business processes are executed locally to each agent.

In Fig. 7, we show how the business processes of an agent are executed starting by the invocation of this agent services by another one. An agent perception


```

<?xml version="1.0" encoding="us-ascii"?>
<delegation>
<from>
  <agent> urlTtreatment </agent>
  <role> mechanical treatment </role>
</from>
<to>
  <agent> urlD </agent>
  <role> delegate-mechanical treatment </role>
</to>
<content>
  <inquiries>
    <inquiry> continue-treatment: horizontal lathes </inquiry>
    <inquiry> model: two pilot axis </inquiry>
    <inquiry> power: 30 Kw </inquiry>
    <inquiry> machine-diameter: lower than 400 mm </inquiry>
    <inquiry> amount: 20 </inquiry>
    <inquiry> due-date: June, 25th, 2003 </inquiry>
  </inquiries>
  <satisfy> request-max-reply-time 5 </satisfy>
  <optimize> accuracy </optimize>
  <optimize> time </optimize>
</content>
</delegation>

```

Fig. 5. The delegation content in the delegation scenario expressed as an XML document

```

<?xml version="1.0" encoding="us-ascii"?>
<obligation name-obligation = "accept delegation">
  <obliged> urlD </obliged>
  <authority> urlTreatment-agent </authority>
  <condition>
    <received-delegation>
      <from> treatment-agent </from>
      <by> delegate-agent-D </by>
    </received-delegation>
  </condition>
  <goal> delegation to produce 20 horizontal lathes
    (mechanical treatment) </goal>
</obligation>

```

Fig. 6. An obligation definition in the delegation scenario expressed as an XML document

interface, which is described using the CORBA IDL (Interface Definition Language), contains the specification of these agent capabilities. The implementation of this interface invokes the underlying component interface, which leads to the execution of the appropriate business process (as shown in Fig. 7).

Consequently, XML provides a freely available and widely transportable methodology for well-controlled data interchange in VE, while CORBA enables to create sophisticated and distributed object systems on heterogeneous platforms.

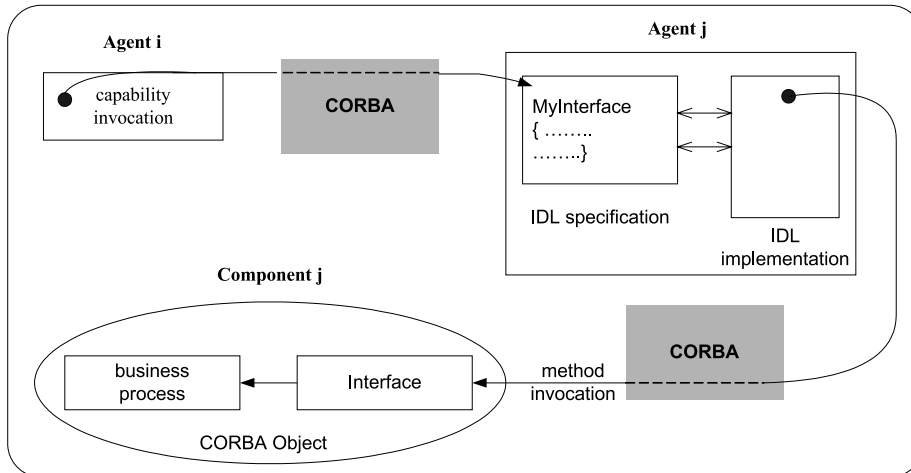


Fig. 7. Invocation mechanism during the business processes execution

5. Related Work

The distributed, autonomous, and heterogeneous nature of information services sets severe technical challenges for VE to provide coherent and compact information. Several architectures and frameworks dealing with the coordination in open environments were developed by the research community. We present those that are directly relevant to the proposed agent-based framework.

Multi-agent planning is fundamental to the generation of cooperative activities. (El Fallah-Seghrouchni and Haddad, 1996) have proposed a recursive model for the representation and the handling of plans by means of Recursive Petri Nets. This recursive model and that (MCT) used in our framework are both based on the Petri Nets. Furthermore, they present some common properties like support of concurrent activities, reaction to the environment using plan execution dynamicity, abstract actions and dynamic refinement, and interaction resolution using plan coordination. However, the authors assume that no global plan is necessary since their approach is based on the paradigm of agent-driven planning. We think that this assumption is not convenient for VE applications where the planning of a global behavior is necessary.

(Marc et al, 2003) have proposed an approach where the principle seems to be similar to ours. The global behavior of the system is modeled at a given level of abstraction. The paper shows how it is possible to model agents' plans by means of hybrid automata. The MAS associates with each task to be achieved, a graph of dependencies that represents a decomposition of the task into sub-tasks. The main advantage of the formalism is the control and the validation of both individual and multi-agent plans.

In (Licci et al, 2001), VE and workflow managements are modeled as coordination issues. The TuCSoN model and technology have been introduced for the coordination of Internet agents. An objective coordination -that is coordination outside the agents- has been used. It seems to provide a more effective support for the three properties: dynamicity, flexibility, and heterogeneity. In our context, contrarily to the objective coordination, a subjective coordination -encapsulating

social laws directly within agent- is adopted, which also provides a more effective support for the three previous properties. The fact of using a subjective coordination leads us to apply effectively the collaborating distributed agents in VE management, while the high-level control activity is distributed among all agents.

The developed approach in (Jain et al, 1999) addresses the process coherence in a VE. A MAS can be viewed as a sphere of commitments. However, no specific details are given with respect to the heterogeneity issue. Moreover, this approach uses the nested spheres of commitments (SoCom) that are difficult to implement.

In (Gal and Mylopoulos, 2000), the authors have constructed a layered approach towards distributed heterogeneous information services, where the coordination mechanism is based on the agent coordinator. The semantic interoperability capabilities are supported at "the application layer", by using semantic-rich models (e.g. Telos and Mirror). However, the mapping from the different models to Telos requires much effort and is not usually possible. Furthermore, Telos and Mirror are neither portable, nor readable by a human.

(Ben-Ameur et al, 2002) have developed an agent-based approach for multimarket package e-procurement. Two fundamental levels are considered: the coordination level and the service one. The used coordination is implicit and semi-centralized. However, the centralized site, which may be a single point of failure, may result in performance problems and scalability issues.

Russel and Norvig have presented an approach (Russel and Norvig, 2002) that consists to build rational agents. They have proposed methods for problem solving, ways to represent knowledge and how to reason logically with it, and mechanisms using these reasoning methods to decide what to do, particularly by constructing plans. The authors have used theories, which are different from ours, to give semantics for the mechanisms described above. For instance, the first-order logic has been used to reason and plan logically, the utility theory to make simple decisions, and the dynamic decision networks to make complex decisions.

Another interesting approach to support the coordination issue in MAS is the coalition formation one. Coalitions are organizations which provide dynamism and flexibility to grasp situations of cooperation and competitiveness in open environments (Vauvert and El Fallah-Seghrouchni, 2000). Coalitions allow agents to satisfy needs requiring synergy for competences of several agents as, for example, the resolution of complex tasks for which agents acting alone would be unable or much less effective. In SoCCoF, the resolution of complex tasks is done through hierarchical organization of agents (section 3.2.1) since a member of an organization may again collaborate with other agents and will play the leader role.

6. Conclusion and Future Work

VE and WfMS require deployable and flexible infrastructures, enabling the integration of heterogeneous resources and services, and coordinating the development of business processes in terms of workflows. As manufacturing becomes increasingly reliant on the dynamic formation and management of extended and overlapping VE, flexible approaches will play an efficient role. In this paper, we have considered a suitable agent-based coordination infrastructure called SoC-CoF, which provides abstractions to address heterogeneity of systems supporting

different VE participants and to represent workflow rules as agent coordination laws (social constraints). This infrastructure may well fit the needs of a VE management in a highly dynamic and unpredictable environment. Preserving the autonomy of participating components is crucial, but unrestrained autonomy would be risky because it may easily lead to undesirable consequences. Therefore, the social constraints that also express data and flow integrity, are necessary components to coordinate the VE business processes. We have presented a formal coordination framework that goes from the fundamental social constraints (like obligations and interdictions) and concepts (like alternative branches and delegation) to the actual structured conversations by which agents directly interact. Indeed, this framework seeks a coherent state in the ongoing interactions of the participating components. In addition to the proposed coordination infrastructure, the paper reports on a prototype implementation of the suggested framework. This prototype supports an efficient exchange of requests for services and processing of rules, due to the richness of mechanisms and concepts considered in the framework. Furthermore, the used communication language, XML, enabled us to prototype quickly the system and build running versions demonstrating the required behavior. We have found the approach explainable to and usable by industrial engineers interested in modeling manufacturing processes.

As for future work, some concepts used in this framework like failure action need clear semantics. One thing we are looking for is using the appropriate formalism like deontic logic. For example, the failure action really denotes a complete non-action in the sense that no action from the set of atomic actions is selected to be performed. On the other hand, our framework is designed only to be supported by the CORBA middleware. In order to permit software components and applications to communicate despite the heterogeneity of their operating systems, technologies, and programming languages and to use HTTP which is supported by all Internet browsers and servers, we hope to integrate SOAP (Introduction to SOAP, 2003) which is an XML based protocol.

Acknowledgements. We thank anonymous reviewers for their very useful comments and suggestions. This work is supported in part by the CMEP grant 99 MDU 454. It has been evaluated in MTP (Machines/Tools Production) group. We would like to acknowledge the support provided by all the members of the MTP group, which allowed us to undertake this project.

References

- Angluin D, Frazier M, Pitt L (1990) Learning conjunctions of Horn clauses. In Proceedings of the 31st Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, St Louis, pp 186–192.
- Barbuceanu M (1997) Coordinating agents by role based social constraints and conversation plans. In Proceedings of American Association for Artificial Intelligence (AAAI'97), pp 16–21.
- Ben-Ameur H, Chaib-Draa B, Gein-Lajoie R, Kropf PG, Vaucher S (2002) Towards an agent-based approach for multimarket package e-procurement, In Scientific Series, CIRANO, <http://www.cirano.qc.ca/pdf/publication/2002s-73.pdf>
- Bellman RE (1957) Dynamic programming, Princeton University Press, Princeton.
- Brumec J, Dusac V, Vrcek N (2000) Strategic planning of information systems- From theory to practice, In Baake UF, Zobel RN, Al Akaidi M (eds). Proceedings of the 7th European Concurrent Engineering Conference (ECEC'2000), Leicester, UK, April 17-19, pp 13-16.
- El Fallah-Seghrouchni A, Haddad S (1996) A recursive model for distributed planning. In Proceedings of the 2nd International Conference on Multi-Agent Systems (ICMAS'96), AAAI Press, Kyoto, Japan.

- Exelente-Toledo CB, Bourne RA, Jennings NR (2001) Reasoning about commitments and penalties for coordination between autonomous agents, In Proceedings of Autonomous Agents (AGENTS01), Quebec, Canada, pp 131-138.
- Fischer K, Muller J, Heimig I, Scheer AW (1996) Intelligent agents in virtual enterprises. In Proceedings of the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'96), April.
- Fitousi D, Tennenholtz M (1998) Minimal social laws. In Proceedings of the 15th National Conference On AI (AAAI-98), 10th Conference On Innovative Applications of AI (IAAI-98), AAAI Press/THEMIT Press, pp 26-31.
- Gal A, Mylopoulos J (2000) Supporting distributed autonomous information services using coordination. International Journal of Cooperative Information Systems, World Scientific Publishing Company, 9(3):255-282.
- Huhns MN, Singh MP (1998) Readings in agents. Morgan Kaufmann, San Francisco.
- Introduction to SOAP (2003) http://www.w3schools.com/soap/soap_intro.asp.
- Jain AK, Aparicio IV M, Singh MP (1999) Agents for process coherence in virtual enterprises. Communications of the ACM, 42(3):62-69.
- Lesser VR (1999) Cooperative multiagent systems: a personal view of the state of the art. IEEE transactions on Knowledge and Data Engineering, 11(1):133-142.
- Licci A, Denti E, Omicini A (2001) Agent coordination infrastructures for virtual enterprises and workflow management. In Proceedings of the 5th International Workshop of Cooperative Information Agent V (CIA 2001), LNAI 2182, M. Klugch and F. Zombonelli (Eds), Italy, pp 235-246.
- Malone T, Crowstone K (1994) The interdisciplinary study of coordination. ACM Computing Surveys, 26(1):87-119.
- Marc F, El Fallah-Seghrouchni A, Degirmenciyan I (2003) Distributed coordination based on temporal planning for tactical aircraft simulation. <http://perso.club-internet.fr/marc.frederic/aamas03.pdf>.
- Mayer JJ (1988) A different approach to Deontic Logic viewed as a variant of Dynamic Logic. Notre Dame, Journal of Formal Logic, 29(1):109-136.
- Minsky N (1991) The imposition of protocols over open distributed systems. IEEE Transactions on Software Engineering, 17(2):183-195.
- Parsons S, Sierra C, Jennings NR (1998) Agents that reason and negotiate by arguing. Journal of Logic and Computation, 8(3):261-292.
- Rocha AP, Oliveira E (1999) An electronic market architecture for the formation of virtual enterprises. In Proceedings of the IFIP/PRODNET Conference on Infrastructures for Industrial Virtual Enterprises, (PRO-VE 99), October.
- Russel S, Norvig P (2002) Artificial intelligence: a modern approach. 2nd edition, Prentice Hall.
- Santanu P, Edwin P, Jarir C (1997) Essential requirements for a workflow standard. In Proceedings of the OOPSLA 1997 Business Object Workshop III, ACM, October.
- Shoham Y, Tennenholtz M (1992) On the synthesis of useful social laws for artificial agent societies. In Proceedings of the 10th National Conference on AI (AAAI-92), San Jose, California, pp 276-281.
- Tardieu H, Rochfeld A, Coletti R (1985) La methode MERISE: dmarche et pratique. In Organisation (ed.).
- Vauvert G, El Fallah-seghrouchni A (2000) Coalition formation for egoistic agents. In Proceedings of the International ICSC Symposium on Multi-Agents and Mobile Agents in Virtual Organizations and E-commerce (MAMA2000), Wollongong, Australia.
- Zarour N, Seinturier L, Boufaïda M, Estraillier P (2000a) An agent-based architecture for the information technology management. In Proceedings of the 2000 IEEE International Conf. on Management of Innovation and Technology (ICMIT 2000), Management in the 21 st century, Singapore, 12-15 November, pp 738-742.
- Zarour N, Boufaïda M, Seinturier L, Estraillier P (2000b) Inter-agent cooperation supported by XML and CORBA in heterogeneous information systems. In Proceedings of the International Workshop on Databases in Networked Information Systems (DNIS 2000), LNCS 1966, S. Bhalla (Ed.), Japan, December, pp 216-227.
- Zarour N, Boufaïda M, Seinturier L (2003) Using information technologies for managing cooperative information agent-based systems. In Proceedings of the 5th International Conference on Enterprise Information Systems, (ICEIS 2003), O. Camp, J. Filipe, S. Hammoudi, and M. Piattini (Eds), Angers, France, 23-26 April, pp 430-437.

Author Biographies



Nacereddine Zarour is Assistant Professor in the Computer Science Department of the University Mentouri, Constantine, Algeria. His current research activities are conducted at the LIRE Laboratory, University of Constantine. His research interests include advanced information systems, particularly cooperative information systems, coordination mechanisms and negotiation models.



Mahmoud Boufaïda is a Professor at the Department of Computer Science of the Mentouri University of Constantine in Algeria. He currently heads the research group "Information Systems and DataBases" of the LIRE Laboratory at the University of Constantine. From 1989 to 1994, he was affiliated to the LIP6 Laboratory at the Pierre and Marie Curie University of Paris in France. His current research interests include cooperative information systems, databases, multi-agent systems and software engineering.



Lionel Seinturier is Assistant Professor in the computer science department of the University Pierre et Marie Curie, Paris, France. He received his PhD degree in computer science from the CNAM, Paris (France) in December 1997. He is currently on leave from the university and he holds a research position at LIFL, a computer science research laboratory that is jointly owned by the University of Lille (France) and INRIA (the French research agency for computer science). Before joining academia, Lionel Seinturier worked as a research engineer for France Telecom's R&D department, on the integration of ATM network technology and CORBA middleware. Lionel Seinturier's research interests include aspect oriented programming and systems, middleware, distributed algorithms and coordination mechanisms in information systems.



Pascal Estrailier is a full professor since 1992, actually in the Computer Sciences department of the University of La Rochelle. He heads the Research Laboratory L3i "Informatics, Image, Interactions" which gathers 62 researchers together. He is Vice-President of the University of La Rochelle, in charge of development and industrial relationships. He is also project manager in the department "Information Technologies" of the French Ministry of research, in charge of Computer Science area. He is co-founder of the first French Educational program on Games at Master Level. Its researches concern the architecture of software components in distributed and cooperative systems. He applies its results on Multi-agent Paradigm and use formal specification theories in order to validate the behaviour and the interactions between components and to manage interoperability constraints. He is in the steering committee of a French Multi-Agent group, in charge of the definition of the future guidance of researches on this topic.