



**HAL**  
open science

# An Optimistic Approach for the Specification of more Flexible Roles Behavioural Compatibility Relations in MAS

Nabil Hameurlain

► **To cite this version:**

Nabil Hameurlain. An Optimistic Approach for the Specification of more Flexible Roles Behavioural Compatibility Relations in MAS. 2007. hal-00192869

**HAL Id: hal-00192869**

**<https://hal.science/hal-00192869>**

Preprint submitted on 29 Nov 2007

**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.

# An Optimistic Approach for the Specification of more Flexible Roles Behavioural Compatibility Relations in MAS

Nabil Hameurlain  
nabil.hameurlain@univ-pau.fr

Laboratoire LIUPPA, Université de Pau  
Avenue de l'Université 64012 Pau – FRANCE

## Résumé :

Dans cet article, nous nous focalisons sur une nouvelle approche de définition d'une compatibilité plus flexible des rôles dans les SMA. Nous proposons une architecture formelle pour la spécification des rôles et leur composition, prenant en compte la préservation de propriétés comme la complétion et la terminaison propre des rôles. Nous mettons en évidence le lien existant entre la compatibilité et la substitutabilité des rôles, et plus particulièrement, nous montrons que les relations de compatibilité ainsi définies sont préservées par la substitutabilité.

**Mots-clés :** Rôles, interaction, compatibilité optimiste, substitutabilité.

## Abstract :

In this paper we focus on a new approach to the definition of more flexible roles compatibility in MAS. We provide a formal framework for modeling roles together with their composition, taking into account the property preservation such as the completion and the proper termination of roles. We show the existing link between roles compatibility and substitutability, namely the preservation of the proposed compatibility relations by substitutability.

**Keywords:** Roles, interaction, components, optimistic compatibility, substitutability.

## 1 Introduction

Roles are basic buildings blocks for defining the organization of multi-agent systems (MAS), together with the behaviour of agents and the requirements

on their interactions. Usually, it is valuable to reuse roles previously defined for similar applications, especially when the structure of interaction is complex. To this end, roles must be specified in an appropriate way, since the composition of independently developed roles can lead to the emergence of unexpected interaction among the agents.

Although the concept of role has been exploited in several approaches [2, 3, 9] in the development of agent-based applications, no consensus has been reached about what is a role and how it should be specified and implemented. In our previous work [4], we have shown that the facilities brought by the Component Based Development (CBD) approach [8] fit well the issues raised by the use of roles in MAS. In this context, we have proposed RICO (Role-based Interactions COmponents) model for specifying complex interactions, and study the compatibility semantics of roles. The RICO model is based on the Component-nets formalism which combines Petri nets and the component-based approach.

In this paper, we focus on a new approach to the definition of role-components compatibility, and provide a formal framework for modelling roles and their

composition. The contributions of this paper are: (1) to provide a new approach to the definition of more flexible role-components compatibility and substitutability relations, (2) to show the existing link between compatibility and substitutability relations, namely the preservation of the compatibility by substitutability.

## 2 Roles modelling

### 2.1 The Component-nets formalism

**Backgrounds on Labelled Petri nets.** A marked Petri net  $N = (P, T, W, M_N)$  consists of a finite set  $P$  of places, a finite set  $T$  of transitions where  $P \cap T = \emptyset$ , a weighting function  $W : P \times T \cup T \times P \rightarrow \mathbb{N}$ , and  $M_N : P \rightarrow \mathbb{N}$  is an initial marking. A transition  $t \in T$  is enabled under a marking  $M$ , noted  $M(t >)$ , if  $W(p, t) \leq M(p)$ , for each place  $p$ . In this case  $t$  may occur, and its occurrence yields the follower marking  $M'$ , where  $M'(p) = M(p) - W(p, t) + W(t, p)$ , noted  $M(t > M'$ . The enabling and the occurrence of a sequence of transitions  $\sigma \in T^*$  are defined inductively. The preset of a node  $x \in P \cup T$  is defined as  $\bullet x = \{y \in P \cup T, W(y, x) \neq 0\}$ , and the postset of  $x \in P \cup T$  is defined as  $x^\bullet = \{y \in P \cup T, W(x, y) \neq 0\}$ . We denote as  $LN = (P, T, W, M_N, l)$  the (marked, labelled) Petri net in which the events represent actions, which can be observable. It consists of a marked Petri net  $N = (P, T, W, M_N)$  with a labelling function  $l : T \rightarrow A \cup \{\lambda\}$ . Let  $\varepsilon$  be the empty sequence of transitions,  $l$  is extended to an homomorphism  $l^* : T^* \rightarrow A^* \cup \{\lambda\}$  in the following way:  $l(\varepsilon) = \lambda$  where  $\varepsilon$  is the empty string of  $T^*$ , and

$l^*(\sigma.t) = l^*(\sigma)$  if  $l(t) \in \{\lambda\}$ ,  $l^*(\sigma.t) = l^*(\sigma).l(t)$  if  $l(t) \notin \{\lambda\}$ . In the following, we denote  $l^*$  by  $l$ ,  $LN$  by  $(N, l)$ , and if  $LN = (P, T, W, M_N, l)$  is a Petri net and  $l'$  is another labelling function of  $N$ ,  $(N, l')$  denotes the Petri net  $(P, T, W, M_N, l')$ , that is  $N$  provided with the labelling  $l'$ . A sequence of actions  $w \in A^* \cup \{\lambda\}$  is enabled under the marking  $M$  and its occurrence yields a marking  $M'$ , noted  $M(w >> M')$ , iff either  $M = M'$  and  $w = \lambda$  or there exists some sequence  $\sigma \in T^*$  such that  $l(\sigma) = w$  and  $M(\sigma > M'$ . The first condition accounts for the fact that  $\lambda$  is the label image of the empty sequence of transitions. For a marking  $M$ ,  $\text{Reach}(N, M) = \{M'; \exists \sigma \in T^*; M(\sigma > M')\}$  is the set of reachable markings of the net  $N$  from the marking  $M$ .

**Components nets (C-nets).** A Component-net involves two special places: the first one is the input place for instance creation of the component, and the second one is the output place for instance completion of the component. A C-net (as a server) makes some services available to the nets and is capable of rendering these services. Each offered service is associated to one or several transitions, which may be requested by C-nets, and the service is available when one of these transitions, called *accept-transitions*, is enabled. On the other hand it can request (as a client) services from other C-net transitions, called *request-transitions*, and needs these requests to be fulfilled. These requirements allow focusing either upon the server side of a C-net or its client side.

**Definition 2.1 (C-net)** Let  $CN = (P \cup \{I, O\}, T, W, M_N, l_{Prov}, l_{Req})$  be a labelled Petri net.  $CN$  is a *Component-net* (C-net) if and only if:

1. The labelling of transitions consists of two labelling functions  $l_{Prov}$  and  $l_{Req}$ , such that:  $l_{Prov} : T \longrightarrow Prov \cup \{\lambda\}$ , where  $Prov \subseteq A$  is the set of provided services, and  $l_{Req} : T \longrightarrow Req \cup \{\lambda\}$ , where  $Req \subseteq A$  is the set of required services.
2. *Instance creation*: the set of places contains a specific *Input* place  $I$ , such that  $I^\bullet = \emptyset$ ,
3. *Instance completion*: the set of places contains a specific *Output* place  $O$ , such that  $O^\bullet = \emptyset$ .

**Notation.** We denote by  $[I]$  and  $[O]$ , which are considered as bags, the markings of the Input and the Output place of  $CN$ , and by  $Reach(CN, [I])$ , the set of reachable markings of the component-net  $CN$  obtained from its initial marking  $M_N$  within one token in its Input place  $I$ . Besides, when we deal with the graphical representation of the C-nets, we use ! and ? keywords for the usual sending (required) and receiving (provided) services together with the labeling function  $l$  instead of the two labeling functions  $l_{Prov}$  and  $l_{Req}$ .

**Definition 2.2 (soundness)** Let  $CN = (P \cup \{I, O\}, T, W, M_N, l)$  be a *Component-net* (C-net).  $CN$  is said to be *sound* iff the following conditions are satisfied:

1. *Completion* option:  $\forall M \in Reach(CN, [I]), [O] \in Reach(CN, M)$ .
2. *Reliability* option:  $\forall M \in Reach(CN, [I]), M \geq [O]$  implies  $M = [O]$ .

The *Completion* option states that, if starting from the initial state, i.e. activation of the C-net, it is always possible to reach the marking with one token in the output place  $O$ . *Reliability* option states that the moment a token is put in the output place  $O$  corresponds to the *termination* of a C-net without leaving dangling references.

**Composition of C-nets.** The *parallel composition* of C-nets, noted  $\oplus : C\text{-net} \times C\text{-net} \rightarrow C\text{-net}$ , is made by communication places allowing interaction through observable services in *asynchronous* way. Given a client C-net and a server C-net, it consists in connecting, through the communication places, the request and the accept transitions having the same service names: for each service name, we add one *communication-place* for receiving the requests/replies of this service. Then, all the accept-transitions labelled with the same service name are provided with the same *communication-place*, and the client C-net is connected with the server C-net through these communication places by an arc from each request-transition towards the suitable communication-place and an arc from the suitable communication-place towards each accept-transition.

## 2.2 Specification of roles

In our RICO model [4], a role component is considered as a component providing a set of interface elements (either attributes or operations, which are provided or required features necessary to accomplish the role's tasks), a behaviour (interface elements semantics), and properties (proved to be satisfied by the behaviour).

In this paper, we only consider behavioural interface of roles that is their behaviour specified by the C-nets together with the set of (provided and required) services.

**Definition 2.3 (Role Component)** A Role Component for a role  $\mathcal{R}$ , noted RC, is a 2-tuple  $RC = (\text{Behav}, \text{Serv})$ , where,

- Behav is a C-net describing the life-cycle of the role  $\mathcal{R}$ .
- Serv is an interface, a set of public elements, through which RC interacts with other role components.  $\text{Serv} = (\text{Req}, \text{Prov})$ , where Req is a set of required services, and Prov is the set of provided services by RC.

Since the life-cycle of roles is specified by C-nets, we say that a component role satisfies the completion (resp. terminates successfully) if and only if its behaviour that is its underlying C-net satisfies the completion option (resp. terminates successfully). The composition of two role-components is also a role-component, and this composition is *associative*.

**Definition 2.4 (Roles composition)** A Role  $RC = (\text{Behav}, \text{Serv})$  can be composed from a set of (primitive) Roles,  $RC_i = (\text{Behav}_i, \text{Serv}_i)$ ,  $i = 1, \dots, n$ , noted  $RC = RC_1 \otimes \dots \otimes RC_n$ , as follows:

- $\text{Behav} = \text{Behav}_1 \oplus \dots \oplus \text{Behav}_n$ .
- $\text{Serv} = (\text{Req}, \text{Prov})$ ,  $\text{Req} = \cup \text{Req}_i$ , and  $\text{Prov} = \cup \text{Prov}_i$ ,  $i=1, \dots, n$ .

### 3 Compatibility of roles

In component-based software engineering, classical approaches for components compatibility deal with

components composition together with their property preservation [1]. In our previous work, we have used this approach for role-based interaction components and study some compatibility relations [5]. In this paper, the basic idea behind the optimistic approach for role-components compatibility is to consider explicitly the context of use of roles (environment) in the definition of roles compatibility relations. First, let define the notion of role's environment.

**Definition 3.1 (Environment)** Let  $RC_1 = (\text{Behav}_1, \text{Serv}_1)$  and  $RC_2 = (\text{Behav}_2, \text{Serv}_2)$ , be two roles such that  $\text{Serv}_i = (\text{Req}_i, \text{Prov}_i)$ ,  $i=1, 2$ .

$CP_2$  is called an environment-role (or environment) of  $CR_1$ , and vice versa, iff  $\text{Req}_1 = \text{Prov}_2$ ,  $\text{Req}_2 = \text{Prov}_1$ .

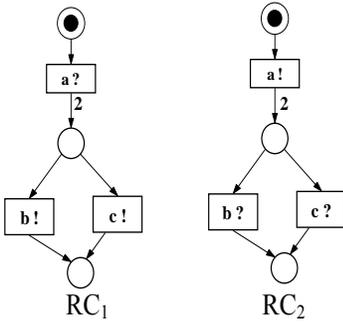
We let  $\text{ENV}(RC)$ , the set of the environments of the role component RC.

The role component  $RC_1$  is considered an *environment* of  $RC_2$  iff both their sets of interfaces completely match.

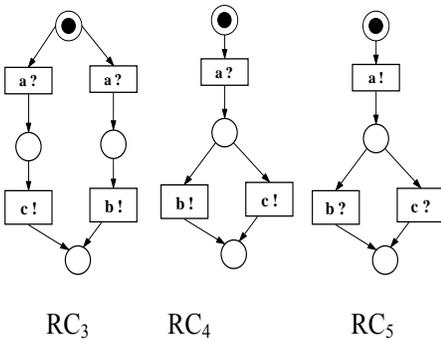
Given a role-component and its environment, it is possible to reason about the completion and the proper termination of their composition. Based on that, we define two notions of usability:

**Definition 3.2 (usability)**

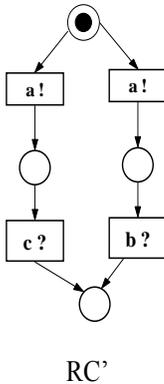
1. RC is weakly usable iff  $\exists \text{Env} \in \text{ENV}(RC)$ ,  $\text{Env} \otimes RC$  satisfies the completion option. We say that  $\text{Env}$  *weakly utilizes* RC.
2. RC is strongly usable iff  $\exists \text{Env} \in \text{ENV}(RC)$ ,  $\text{Env} \otimes RC$  terminates successfully. We say that  $\text{Env}$  *strongly utilizes* RC.



**Fig 1.**  $RC_1$  weakly utilizes  $RC_2$ , where  $l(a)=\text{Ticket}$ ,  $l(b)=\text{Visa}$ ,  $l(c)=\text{eCash}$ .<sup>1</sup>



**Fig 2.**  $RC_3$  strongly utilizes  $RC_5$ ,  $RC_4$  strongly utilizes  $RC_5$ .



**Fig 3.**  $RC'$  is not weakly usable.

**Example 1:** Let's take the example of the ticket service and the customer. Figure 1 shows  $RC_1$  representing the behaviour of the customer, and  $RC_2$  the behaviour of the Ticket-service. The Ticket service initiates the communication by sending (two) Tickets and waits of their payment (VISA and/or eCash). By receiving the Tickets, the customer determines the kind of payment of these two tickets. It is easy to prove that roles  $RC_1$  and  $RC_2$  are weakly usable, since  $RC_1$  weakly utilizes  $RC_2$  and vice versa. The role  $RC_1$  is not strongly usable, since the unique (weakly usable) environment of  $RC_1$  is the role  $RC_2$ , and  $RC_1 \otimes RC_2$  satisfies the completion option but does not terminate successfully. In figure 2, the ticket service  $RC_5$  initiates de communication by sending one Ticket and waits of the payment (either Visa or eCash). The role components  $RC_3$  and  $RC_4$  are two examples of the customer's behaviour. By receiving the Ticket, they solve an internal conflict and determine the kind of payment. The roles  $RC_3$  and  $RC_5$  (resp.  $RC_4$  and  $RC_5$ ) are strongly usable, since for instance  $RC_3$  strongly utilizes  $RC_5$  (resp.  $RC_4$  strongly utilizes  $RC_5$ ) and vice versa. Last but not least, let us take the ticket service  $RC'$  shown in figure 3.  $RC'$  is not weakly usable since there is no environment which can weakly utilize it. Indeed, roles  $RC_3$  and  $RC_4$  are the two possible role-environments of  $RC'$  (according to the behaviour of  $RC'$  described by the language  $\{\text{Ticket!.Visa?}, \text{Ticket!.eCash?}\}$ ), nevertheless, for instance the occurrence of the sequence  $\{\text{Ticket!.Ticket?.eCash!}\}$  in  $RC_3 \otimes RC'$  (as well as in  $RC_4 \otimes RC'$ ) yields a

<sup>1</sup> The names of transitions are drawn into the box.

deadlock- marking that is a marking where no transition is enabled. This is because of an error in role-component RC': an internal decision is made (either Visa? or eCash?), when sending the Ticket, and not communicated properly to the environment [1].

We are finally ready to give adequate definitions for roles behavioural optimistic compatibility relations, which are based on the weak and the strong usability.

**Definition 3.3 (compatibility)** Let  $RC_1$  and  $RC_2$  be two weakly (resp. strongly) usable roles.

$RC_1$  and  $RC_2$  are Weakly (resp. Strongly) Optimistic Compatible, noted  $RC_1 \approx_{woc} RC_2$  (resp.  $RC_1 \approx_{soc} RC_2$ ), iff  $RC_1 \otimes RC_2$  is weakly (resp. strongly) usable.

**Example 2:** As an example, it is easy to prove that roles  $RC_1$  and  $RC_2$ , shown in figure 1, are weakly optimistic compatible that is  $RC_1 \approx_{woc} RC_2$  holds since  $RC_1 \otimes RC_2$  is weakly usable. Indeed,  $RC_1 \otimes RC_2$  satisfies the completion option. Besides, the two roles  $RC_3$  and  $RC_5$  shown in figure 2 are strongly optimistic compatible that is  $RC_3 \approx_{soc} RC_5$  holds since  $RC_3 \otimes RC_5$  is strongly usable. Indeed,  $RC_3 \otimes RC_5$  terminates successfully.

**Property 3.1 (Hierarchy of compatibility)** Compatibility relations form a hierarchy:  $\approx_{soc} \Rightarrow \approx_{woc}$

## 4 Substitutability of roles

We show the existing link between compatibility and substitutability

concepts, and namely their combination, which seems necessary, when we deal with incremental design of usable components-role. Our main interest is to define behavioural subtyping relations (reflexive and transitive) capturing the principle of substitutability [7]. We define two subtyping relations based upon the preservation of the (weakly and strongly) utilizing of the former role by any role of its environment.

### Definition 4.1 (behavioural subtyping)

Let  $RC_i = (\text{Behav}_i, \text{Serv}_i)$ ,  $\text{Serv}_i = (\text{Req}_i, \text{Prov}_i)$ ,  $i=1,2$ , be two roles, such that:  $\text{Prov}_1 \subseteq \text{Prov}_2$  and  $\text{Req}_1 \subseteq \text{Req}_2$ <sup>2</sup>.

1.  $RC_2$  is less equal to  $RC_1$  w.r.t Weak Substitutability, denoted  $RC_2 \leq_{ws} RC_1$ , iff  $\forall \text{Env} \in \text{ENV}(RC_1)$ ,  $\text{Env}$  weakly utilizes  $RC_1 \Rightarrow \text{Env}$  weakly utilizes  $RC_2$ .
2.  $RC_2$  is less equal to  $RC_1$  w.r.t Weak Substitutability, denoted  $RC_2 \leq_{ss} RC_1$ , iff  $\forall \text{Env} \in \text{ENV}(RC_1)$ ,  $\text{Env}$  strongly utilizes  $RC_1 \Rightarrow \text{Env}$  strongly utilizes  $RC_2$ .

Weak (resp. Strong) Substitutability guarantees the transparency of changes of roles to their environment. In both *weak* and *strong* subtyping relations, the (super-) role component  $RC_1$  can be substituted by a (sub-) role component  $RC_2$  and the environment of the former role  $RC_1$  will not be able to notice the difference since: (a) the sub-role has a larger set of required and provided services ( $\text{Req}_1 \subseteq \text{Req}_2$  and  $\text{Prov}_1 \subseteq \text{Prov}_2$ ) than the super-role, and (b) any environment that weakly (resp. strongly)

---

<sup>2</sup> The sub-role component has a larger set of (required and provided) services ( $\text{Req}_1 \subseteq \text{Req}_2$  and  $\text{Prov}_1 \subseteq \text{Prov}_2$ ) than the super-role component.

utilizes the former role is also able to weakly (resp. strongly) utilize the new role.

**Example 3:** As an example, consider the roles  $RC_4$  and  $RC_1$ .  $RC_1 \leq_{WS} RC_4$  holds since the unique environment that weakly utilizes  $RC_4$  is the role  $RC_5$ , and  $RC_5 \otimes RC_1$  satisfies the completion option. These two roles  $RC_1$  and  $RC_4$  are not related by the strong subtyping relation that is  $RC_1 \leq_{SS} RC_4$  does not hold, since  $RC_5 \otimes RC_1$  does not terminate successfully. Last but not least, consider the roles  $RC_4$  and  $RC_3$ ;  $RC_3 \leq_{SS} RC_4$  holds since the role  $RC_5$  (which is the unique environment) that strongly utilizes  $RC_4$  also strongly utilizes  $RC_3$ . Indeed  $RC_5 \otimes RC_3$  terminates successfully.

**Property 4.1 (Hierarchy of subtyping)**

The relations  $\leq_H$ ,  $H \in \{WS, SS\}$ , are preorder (reflexive and transitive) and form a hierarchy:  $\leq_{SS} \Rightarrow \leq_{WS}$ .

The following core theorem of this paper states two fundamental properties of roles compatibility and substitutability relations. First, substitutability relations are compositional: in order to check if  $Env \otimes RC_2 \leq_H Env \otimes RC_1$ ,  $H \in \{WS, SS\}$ , it suffices to check  $RC_2 \leq_H RC_1$ , since the latter check involves smaller roles and it is more efficient. Second, substitutability and compatibility relations are related as follows: we can always substitute a role  $CR_1$  with a sub-role  $CR_2$ , provided that  $RC_1$  and  $RC_2$  are connected to the environment  $Env = (Behav, Serv)$  by the same provided services that is:  $Req \cap Prov_2 \subseteq Req \cap Prov_1$ . This condition is due to the fact

that if the environment utilizes services provided by  $CR_2$  that are not provided by  $CR_1$ , then it would be possible that new incompatibilities arise in the processing of these provided services.

**Theorem 4.1 (compositionality and compatibility preservation)** Let  $RC_1 = (Behav_1, Serv_1)$ ,  $RC_2 = (Behav_2, Serv_2)$  be two roles where  $Serv_i = (Req_i, Prov_i)$ ,  $i = 1, 2$ . Let  $Env = (Behav, Serv)$  such that  $Req \cap Prov_2 \subseteq Req \cap Prov_1$ .

1.  $Env \approx_{WOC} RC_1$  and  $RC_2 \leq_{WS} RC_1 \Rightarrow Env \approx_{WOC} RC_2$  and  $Env \otimes RC_2 \leq_{WS} Env \otimes RC_1$ .
2.  $Env \approx_{SOC} RC_1$  and  $RC_2 \leq_{SS} RC_1 \Rightarrow Env \approx_{SOC} RC_2$  and  $Env \otimes RC_2 \leq_{SS} Env \otimes RC_1$ .

**5 Conclusion and related work**

The aim of this paper is to present a new and optimistic approach to the definition of role-components behavioural compatibility and substitutability relations. The paper provides a framework for modelling usable role-components together with their composition. This framework is discussed in terms of roles compatibility and substitutability relations. We furthermore investigated the link between compatibility and substitutability relations by showing that substitutability is compositional and the compatibility is preserved by the substitutability.

**Related work.** The optimistic approach to the definition of components compatibility has been originally introduced in [1] for interface automata. Unlike traditional uses of automata, the authors proposed an optimistic approach

to automata composition. Two interface automata are (optimistic) compatible, if there exists a legal environment for these two automata, i.e. an environment such that no deadlock state is reachable in the automata obtained by the composition of the two interface automata and that environment. This work is close to ours, since our weak optimistic compatibility relation for role-components is related to the optimistic compatibility relation defined for automata composition. Our approach can be seen as an extension of this work, since it deals in addition with strong optimistic compatibility, which is related to the proper termination property. In [6], the concept of usability is used for analyzing web service based business processes. The authors defined the notion of usability of workflow modules, and studied the soundness of a given web service, considering the actual environment it will be used in. Based on this formalism together with the notion of usability, the authors present compatibility and equivalence definitions of web services. This approach is close to ours, since the compatibility of two workflow modules is related to our strong optimistic compatibility of role-components. Our approach can be seen as an extension of this work, since we define in addition the notion of weak optimistic compatibility and study the existing link between compatibility and substitutability.

## References

- [1] L. De Alfaro, T.A. Henzinger. Interface Automata. In Proc. of ESEC/FSE, Vol. 26, 5 of Software Engineering Notes, ACM (2001).
- [2] M. Dastani, V. Dignum, F. Dignum. Role Assignment in Open Agent Societies. AAMAS'03, ACM 2003.
- [3] G. Cabri, L. Leonardi, F. Zambonelli. BRAIN: a Framework for Flexible Role-based Interactions in Multi-agent Systems. CoopIS 2003.
- [4] N. Hameurlain, C. Sibertin-Blanc. Specification of Role-based Interactions Components in MAS. In Software Engineering for Multi-Agent Systems III. LNCS, pp 180-197, Vol. 3390, Springer, 2005.
- [5] N. Hameurlain. Formalizing Compatibility and Substitutability of Role-based Interactions Components in MAS. CEEMAS'05, LNAI/LNCS Vol. 3690, pp 153-162, 2005.
- [6] A. Martens. Analyzing Web Service Based Business. FASE'2005, pp 19-33, Vol. 3442, LNCS, Springer, 2005.
- [7] B. H. Liskov, J. M. Wing. A Behavioral Notion of Subtyping. In ACM TPLS, Vol 16, n° 6, Nov. 1994.
- [8] C. Szyperski. Component Software-Beyond Object-Oriented Programming. Addison-Wesley, 2002.
- [9] F. Zambonelli, N. Jennings, M. Wooldridge. Developing Multiagent Systems : The Gaia Methodology. ACM TSEM, Vol 12, N° 3, July 2003, pp317-370.