

A New Formalism for Modeling a Multi Agent Systems: Agent Petri Nets

Borhen Marzougui¹, Khaled Hassine², Kamel Barkaoui³

¹Higher Institute of Technological Studies of Mednine, Tunisia; ²Faculty of Science, Gabès, Tunisia; ³National Conservatory of Arts and Crafts, Paris, France.

Email: marzougui_bor@yahoo.fr, Khaled.hassine@gmail.com, kamel.barkaoui@cnam.fr

Received November 11th, 2010; revised November 22nd, 2010; accepted December 1st, 2010

ABSTRACT

In this paper, we present a new formalism for Modeling Multi Agent Systems (MAS). Our model based a PN is able to describe not only not the internal state of each agent modeled but also its behavior. Owing to these features, one can model naturally the dynamic behavior of complex systems and the communication between these entities. For this, we propose mathematical definitions attached to firing transitions. To validate our contribution, we will deal with real examples.

Keywords: Multi Agent Systems, Method, Agent Petri Nets, Formalism

1. Introduction

Since the appearance of the Petri Nets [1], the inventors have not stopped proposing new models, either to improve an already existing formalism or to create a new model. These formalisms depend from the studied system types. They permitted to make the conception more natural, more intuitive and more familiar by Petri Nets (PN). Indeed, the Petri Nets can be considered as graphic and mathematical tools of modeling. To modelize and analyze the discreet system, particularly the competitive, parallel and non-determinist ones, it is necessary to choose the appropriate type of PN to be used. This type must be capable of modelizing rigorously the systems of large size as the multi agent systems. Such systems permit to coordinate the intelligent agent behavior interacting and communicating in an environment to achieve some tasks or to solve some problems [2].

According to [3], the modeling of Multi Agent Systems (MAS) proves to be applicable to represent the actions of the agents and their consequences in the environment that can be complex and of an autonomous evolution. In fact, the complexity of the system studied is increasing. The precision, reliability and the hardiness have become difficult factors to reach. The previous works of the Petri Nets concentrated on their uses and not on the creation of the new models as the works of [4-6]. The research of a new model has been ignored for a long time. However, there were some works that took into account the exten-

sion of some classic types of the Petri Nets to reach a more or less generic model to satisfy a need of modeling. [7] proposed a Petri net model approach to formal specification of holonic control systems for manufacturing. To build up models of transition operations robotic, [8] presented a specific algorithm. Therefore, the integration of a mathematical tool offers an exact way, in presence of graphic tools, to succeed the conception of these systems, particularly the multi agent systems.

The objective of the present paper consists in proposing a new type of Petri Nets based on the agents that help us understand the functioning of the MAS. In order to describe the behavior and the interactions of the entities of the system or the constraints on the variable characteristics of the system, we should make a dynamic modeling. This modeling must be achieved by an adequate formalism that will be presented in our work.

This paper is organized as follows. The second section describes the Multi Agent Systems and their modeling by the Petri Nets. Besides, the third section explains the limits of the classic PN. Our new model titled Agent Petri Nets is presented and the definitions formulating this formalism are interpreted in the fourth section. Next, in the fifth section, we present the correspondence between the MAS approaches and APN. The sixth section presents an example of Modeling by Agent Petri Nets. Section Seven finally concludes the paper and outlines future works.

2. Modeling Multi Agent Systems by Petri Nets

A multi agent system permits to coordinate the behavior of agents, interacting and communicating in an environment, to achieve some tasks or to solve some problems [9-11]. It allows the decomposition of complex task in simple sub-tasks which facilitates its development, test and updating.

The modeling of MAS requires toll to check first characteristics and properties of agents, then, those of the system itself. That's why, several works have been achieved on the formalization of the MAS by different formal methods as those of work [3,12-15]. In [13] the proposed approach is focused on the formalisms as a language for describing the models produced in each phase of the development process. But this still requires a specific agent. Also, in [15] an Agents designed thanks to CLAIM are endowed with cognitive capabilities, are able to communicate with other agents and are mobile. But these primitives are limited only to the cognitive agent and do not address the other classes of agents. So, we must think to generalize these primitives.

Few formalisms have been defined (as finite state automata) which are ineffective when must consider aspects of parallelism [7]. The algebraic models based on differential equations are also inefficient to represent agent's interactions. So, it is imperative to dispose a formalism capable of expressing the internal state agents, their behaviors and the interactions between them. In this context, the use of the PN to modelize MAS presents a major contribution. For example, a Colored Petri Net (CPN) is used in [16-18] to modelize the simultaneous communications of the agents by using the functions manipulating colors. Also, to describe the architecture and the behavior of systems, [19] propose a new model of transformation for the inversion of the original CPN and the implementation of the analysis phase. This technique can provide a formal basis but which requires enrichment of marking. However, the Object Petri Nets (OPN) [18] presents a power to modelize the dynamic aspects of the agents, but use a lot of places and transitions which makes the model rather complex.

3. Limits of the Classic Petri Nets

The classic Petri Nets as those of Place/Transition, Colored, and Object present a deficiency at the level of their expression if it relates to the system of large size as the multi agent system. These systems are characterized by the interactivity of the elements that they compose. For the Colored PN, for example, the classes of colors cannot directly express the state of the elements (tokens) of the system or the relations between them. In addition, the

OPN can describe effectively the internal state of the tokens but not the relations between them because this requires additional places and transitions which brings into play the methods used. Indeed, an Objects Petri Nets modelizes a multi agent system by a rather high number of places and transitions. The firing of a token is assured by the invocation a set of methods. This describes with difficulty the behaviors of the agents around their environments [6].

The multi agent approach can be considered as an evolution of the object-oriented paradigm. From a conceptual viewpoint, an object is merely a data structure which is associated with the functions. The agents are autonomous entities whose behavior does not depend on an outside expression, contrary to the objects.

The already achieved works concern the modeling of the MAS by a PN respecting some specific needs. It is often needed to make a coupling between two types of Petri Net to satisfy a possibly determinist aspect in the system specification as the interaction and the communication between the different entities that compose it. In [20] a description of the abstract architecture for multi-agents systems with an indirect interaction has been presented but with only two agents. Therefore, our idea consists in benefiting from the properties and characteristics of agents and integrating them in a classic Petri Net. Thereafter, we propose our approach that consists in defining a new model of Petri Nets called Agent Petri Nets. Previous work modeling MAS by classical Petri nets [8,13,18] does not allow a direct passage to the implementation phase. Because this requires changes to rules implementing a model of agent oriented languages. So, we must define a model based agent to simplify its implementation.

4. Proposed Formalism: Agent Petri Nets

An agent is defined as an autonomous entity capable of communicating with other agents to partially discern at least its environment and the objects that surround it, and to have correct or erroneous representations about the behaviors of a part or the set of the other agents of the environment [21]. So, contrary to the objects, an agent possesses an autonomous behavior. It is capable of taking some decisions and establishing plans of actions to accomplish complex activities.

4.1. Definition 1: Agent Petri Nets

An Agent Petri Nets is defined as being a directed bipartisan graph that has two types of nodes (place and transition). The arcs are bonds between these nodes which indicate the conditions of activation of a transition. Every transition carries the functions that manipulate the inter-

nal state and the behavior of an Agent (Token) in its environment. The distribution of the tokens in the places at a given moment is called marking of the Agent Petri Net.

A marking gives the state of the system that depends on the interaction between the entities that compose it. The change in internal of the state or the behavior of every Agent, in the first place or the whole system, in the second place, is assured by functions.

In a formal manner, Agent Petri Net is defined by the 9-uplet:

$$Q = \langle P, T, A, Meadow, Post, Pr_j, F, Ft, Env_k \rangle$$

where:

- P is a non-empty finished whole of places;
- T a non-empty finished whole of transitions;
- A is a non-empty finished whole of agents;
- Meadow: $P \times T \rightarrow N$ an application of front incidence;
- Post: $P \times T \rightarrow N$ a back application of incidence corresponds to the arcs;
- Pr_j : pre condition of firing;
- $F(A_i, A_j)$: agent relation function presenting the condition of firing;
- F_t : function agent that uses three variables; $F_t(t_k) = \langle Per, value, Inter \rangle$;
- Env_k : Environment of work that describes multi agent system.

4.2. Definition 2: Constraints of an Agent Petri Net

A constraint of an agent A_i is defined as:

$$Cont(A_i, P_j, T_k) = b.$$

It's defined as being a firing of a T_k transition descended of a P_j place.

In a formal manner, the constraint on an agent it's defined as:

$$\forall i \in I, j \in J \text{ et } \forall k \in K, \exists Cont(A_i, P_j, T_k) = b.$$

Where:

- I: set of numbers of tokens of a place;
- J: set of numbers of places;
- K: set of numbers of Transitions;
- i: index of an Agent;
- j: index of a Place;
- k: index of a Transition;
- b: Boolean (0 or 1).

4.3. Definition 3: Pre-condition Function: Pr_j

That is to say $Cont(A_i, P_j, T_k) = b$, for a number n_i of agents which is engaged in an environment, we get:

$$Pr_j = \prod_{i=1}^{i=n_i} Cont(A_i, P_j, T_k) = b$$

where Pr_j it's the pre-condition function descended of a place P_j .

By hypothesis, an agent A_i must only be engaged in only one environment. It is defined as:

$$Card(Env(A_i)) = 1$$

The Boolean value sent back by Pr_j gives the starting point to an action (transition). The engagement of an agent in a well-defined environment will be preceded by the control made by this function. The subset n_i of the agents has an equitable environment cardinality that is equal to 1 or 0.

- 1) If $Pr_j = 0$ then the pre-condition is not valid and in this case at least exists an agent that is already engaged in another environment.
- 2) If $Pr_j = 1$ then the pre-condition is valid and in this case it is guaranteed that all agents are not already engaged in another environment.

Illustration (Figure 1): It is supposed that:

Workshop1 contains the Machine M1 and M2 but Workshop2 contains the Machine M3, M4 and M5:

- 1) Case 1: the two machines M1 and M2 belong to the same workshop (Environment): Workshop1. In this case their use is permitted: pre-condition $Pr_0 = 1$.
- 2) Case 2: the Machine M1, M3, M4 and M5, cannot belong to the same workshop (Environment: Workshop2) because the Machine M1 is already engaged in another environment. In this case one cannot cross the transition T1.

4.4. Definition 4: Function of Adherence (Relative to an Agent)

This function gives rise to a relation between an agent and its environment. The engagement of an Agent A_i in an environment Env_k describes initially an adherence criterion then the number of times that this agent has been engaged in Env_k .

This offers more mechanisms of explanations and minimizes the difficulties with the tasks that require knowledge of the world (Env) which can be obtained only by memorizing or reasoning and not by perception. The definition of Ferber [2] that showed that a cognitive agent has the capacity to reason on representations of the world, to memorize some situations, to analyze them, to foresee some reactions possible for any action, to draw the conducts of the future events and therefore to plan its own behavior is used.

In a formal manner, the adherence function of an agent A_i , in an environment Env_k noted Apa_i is defined by:

$$\forall A_i \in A \text{ and } \forall Env_k \subset Env,$$

$$\exists Apa_i = Apa(A_i, Env_k, b, d)$$

$$d \in \mathbb{Z}^+$$

where:

- a) b: constraint = Pr_j (b = 0 or b = 1):the engagement of A_i in Env_k .
- b) d: adherence degree: altogether gives the number of times that the agent A_i has been engaged in Env_k .

Importance of the adherence function

Constantly, this function gives a description of the relation between the agent and a well defined environment. It guarantees the update of the knowledge base of an agent. An agent's reaction depends on its environment. The evolution of Agent Petri Nets depends on the system to study what implies, implicitly, that each agent seeks the criterion of competence of another. That's why it must interpret the value of d.

This value d can enrich the knowledge base of an agent.

Illustration: (Figure 2) let's take the example of the Figure 1 with some modifications:

- c) Before the firing of T_0 and T_1 , agent M1 admits $d=0$ for the environment of Workshop1 and Workshop2 respectively:

- $ApM1 = Apa(M1, Workshop1, 1, 0)$;
- $ApM1 = Apa(M1, Workshop2, 1, 0)$.

After the firing of T_0 , agent M1 will have $d = 1$ for the Workshop1 environment and 0 for Workshop2:

- $ApM1 = Apa(M1, Workshop1, 1, 1)$;
- $ApM1 = Apa(M1, Workshop2, 1, 0)$.

Agent M1 leaves his Workshop1 environment: firing of the T_2 transition. It will allow it to be free.

After the shooting of T_1 , agent M1 will have $d = 1$ for the environment of Workshop1 and Workshop2 respectively:

- $ApM1 = Apa(M1, Workshop1, 1, 1)$
- $ApM1 = Apa(M1, Workshop2, 1, 1)$

4.5. Definition 5: Adherence Function (relative to an environment)

The creation of the adherence function Apa_i of an A_i agent in an environment Env_k allows us to find an adherence function Ape_i conversely. This new function describes the set of the agents which belong to the same environment j with certain degree of adherence di.

The function of adherence related to an Env_k environment is defined as:

$$Ape_j = Ape\left(Env_k, \bigcup_{i=1}^{i=ni} (A_i, d_i)\right)$$

where:

- ni: number of agents in the environment;
- A_i : Index Agent I;
- d_i : degree of adherence of agent of i index.

Thus, this function can be simplified as:

$$Ape_j = Ape\left(Env_k, \bigcup_{i=1}^{i=ni} (d_i)\right)$$

Illustration: From the APN of Figure 2 the degree of adherence of every environment can be deduced:

$$ApeWorkshop1 = Ape\left(Workshop1, \bigcup_{i=1}^{i=5} (d_i)\right)$$

$$ApeWorkshop2 = Ape\left(Workshop2, \bigcup_{i=1}^{i=5} (d_i)\right)$$

with 5 is the number of machinery (Agent) used. The following adherence matrix can be deduced:

	M1	M2	M3	M4	M5
Workshop1	1	1	0	0	0
Workshop2	1	0	1	1	1

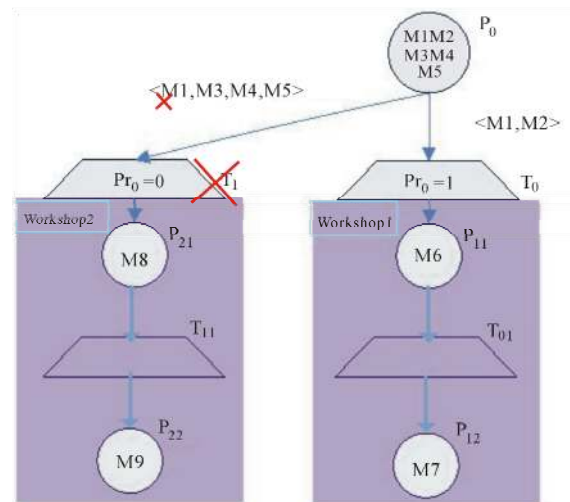


Figure 1. Illustration of the pre-condition function Pr.

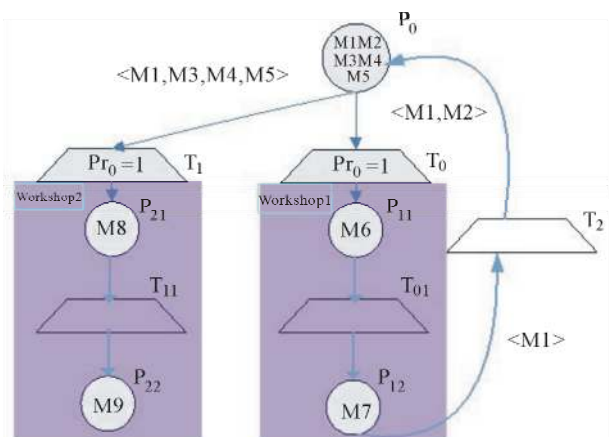


Figure 2. Illustration of the adherence function.

4.6. Definition 6: Moderator Agent

An agent is said to be moderator if it is priority than another. This indicates that the moderator dominates at the time of a communication, or it possesses a hierarchical degree (dh) less elevated (dh = 2 dominates dh = 3). An agent is said to be total moderator if it is important by contribution to all agents of its environment. Thus, it possesses a hierarchical degree dh that is equal 1.

4.7. Definition 7: Function Relationship F:

This function is defined as a function admitting two entries E1 and E2 and in exit only one boolean value S. The entry E1 is imperatively moderator. This function presents a pre-condition of firing of a transition (**Figure 3**).

Thus, this relation is defined by the function

$$F(E1, E2) = S.$$

Let A_i and A_j two agents in the same environment Env_k .

$$\forall A_i \in A, \forall A_j \in A, \exists F(A_i, A_j) = S$$

where:

- A: set of the agents of the environment;
- A_i : is an agent moderator;
- A_j : is not an agent moderator;
- S: Boolean (1 or 0).

Thus, this function can be generalized to get a function of relation agent of n order: set of n agents a function F as:

$$F(A_1, A_2, \dots, A_n) = S$$

- 1) If $S = 0$ then no relation exists between the two agents communicate between each other. In this case, the non-moderator agent cannot enter in relation with the moderator agent whether voluntarily or obligatorily by the moderator agent of total order, or because it is already occupied.

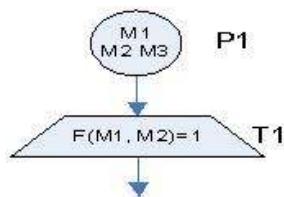


Figure 3. Function relationship.

- 2) If $S = 1$ then a relation between the two agents concerned is established. In this case, the agent moderator asks for the establishment of a communication with another agent that is called non-moderator and which accepts this demand.

4.8. Definition 8: Function Agent Ft

The function agent describes the relation between two agents, the data interchange and the behavior of each of them. It modifies the values descended directly of an agent. These define the capacity to discern and to react to the modifications occurred in its environment. Generally, it is written as follows:

$$F_t(t_k) = \langle Per, Value, Inter \rangle$$

- 1) Initially, $F_t(t_k) = \langle 0, \Phi, 0 \rangle$, it implies that there is not any interaction between the agents. If the value of $Per = 0$ then directly we have $Inter = 0$. Never can we have $Per = 0$ nor and $Inter = 1$. $Value = \Phi$, in this case no action is triggered and the previous situation of the agents is maintained.
- 2) In the course of the firing of the transition t_k , there will be change of values between the agents. In this case Per takes the value 1, $Inter$ takes the value 0 and $Value$ defined the action or the task to achieve. The relation of order already defined gives the sense of transfer of the information.

$$So : F_t(t_k) = \langle 1, Value, 0 \rangle$$

- 3) After the firing of the transition t_k , $Inter$ takes the value 1; it indicates that the action has been achieved successfully.

$$So : F_t(t_k) = \langle 1, Value, 1 \rangle$$

5. From the Multi Agent Systems Toward the Agent Petri Nets

In the form of a table, a kind of correspondence between the two approaches according to well defined characteristics will be given.

Table 1 summarizes the various properties of multi-agents systems and that of our formalism. Indeed, the ability to express our model is the benefit of two important paradigms: Petri nets and MAS. The ability of expression is seen from that carried:

- The agent is presented as a single token. The relationship between the agent and its environment is defined by a function. RdPA model describes the behavior of an agent clearly. The firing of a transition gives a newstate for the agent and therefore for the whole system.
- The control of behavior of agent and its analysis are done easily by interpreting the transition carrying a function.

The underlying idea of Agent Petri Nets is to properly represent the agent and its autonomy in communication with other agents while maintaining a simple graphical representation and understandable. This means they combine clear denotation which can be displayed in the gra-

Table 1. Correspondence between MAS and APN.

Characteristic	Multi agent Systems	Agent Petri Nets
	Agent	Token
	State of the system	Place
Name	Set of rules	Meadow condition (Pr_j)
	Set of relations of actions	Transition
	Agent Administrator	Agent Total Moderator
	Agent Reactive	Agent not Moderator
Class	Agent Cognitive	Agent Moderator
	Agents Hydride	Agent Total Moderator
Autonomy	Interaction between agents	Function relation relationship
Reactivity	Agent - Agents	Function Agent $F_t = \langle Per, Valeur, Inter \rangle$
Heterogeneity	Agent - Environment	Related to adherence Function an agent Related to adherence function an agent (Apa_i) and her environment:
Sociability	Environment - Agents	$Ape_j = Ape \left(Env_j, \bigcup_{i=1}^{i=ni} (d_i) \right)$
Intelligence	Comportment, capacity of interaction	Exploitation of the values possible of Per , $Inter$ and $Value$.

physical representation with an operational semantics that can be executed. The model can be performed for the simulation. This is an advantage for understanding the interaction and ensures its successful implementation. However, Agent Petri Nets can also be used for abstract modeling that can display previews or details (sophistication) of the system.

6. Example: Modeling Of print Job

The system is composed by two agents *CPU* and *Printer*. We propose to model this system by two types of PN namely PT PN and APN. **Figure 4** present a modeling of Print Job by PTPN.

where:

- P_1 : CPU pending;

- P_2 : data ready for printing;
- P_3 : printer pending;
- P_4 : print data;
- T_1 : preparation of data to print;
- T_2 : begin print;
- T_3 : end print.

Figure 5 present a modeling of Print Job by APN.

There is reduction (folding) at waist level Petri net. Indeed, the number of places becomes 2 instead of 4 and the transition becomes 2 instead of 3. The initial marking M_0 is the first case $M_0 = (1, 0, 1, 0)$. For the second case, $M_0 = (2, 0)$. This also implies that we can get a tree reduced marking. Thus, the APN model describes an intel-

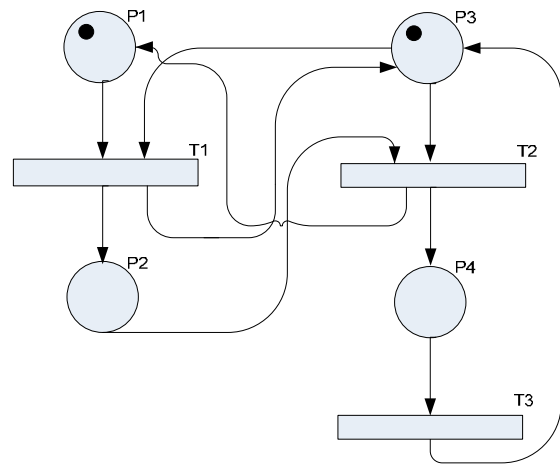


Figure 4. Modeling of print Job by a classic PN.

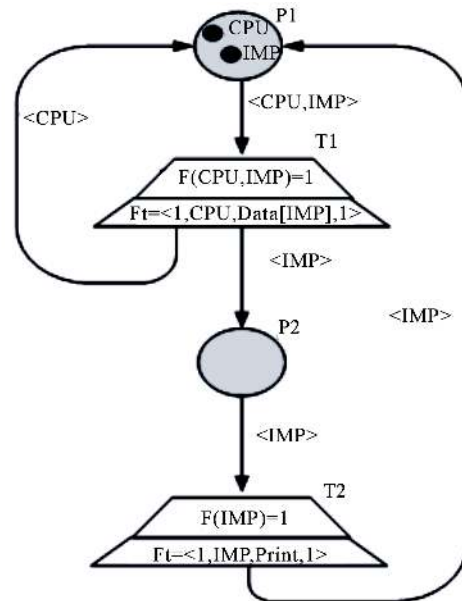


Figure 5. Modeling of print Job by APN.

ligent manner the state and behavior of the system with a simple graphic presentation.

7. Conclusion

We defined the framework of our work by expressing our position compared to works which treated the formal methods for modeling of the systems multi agent. Besides, a formalism that combines the Petri Nets and the MAS is proposed. This combination gives birth to a new formalism called "Agent Petri Nets". Actually all definitions in relation to this type of network are given. It benefits from the characteristics of the agents and multi agent systems. Indeed, each token of a place represents an agent and the transition is equipped with a set of functions that describes particularly the condition of its firing and the relations between the agents. The major contributions of an Agent Petri Net are firstly the power of expression, secondly the modeling of the interactions size of network and finally the gain at the level of modeling time. The definition of this model helps in modeling the internal state and the dynamic behavior of an agent in MAS efficiently. We will explore more deeply the modeling of other real cases of different complex systems. Thus, we must add some functions to model the mobility of an Agent and its migration from one environment to another.

REFERENCES

- [1] C. A. Petri, "Communication with Automats," Thesis of Doctorat, Boon University, 1962.
- [2] J. Fougères, "A Cognitive Architecture of Communicating Agents in Complex Information Systems," Belfort-Montbéliard University of Technologies, 2002.
- [3] J. Ferber, "The Multi Agent System: Towards a Collective Intelligence," Inter-Editions, Paris, 1995.
- [4] E. Tagne, C. Viho, E. Tonye and Akono, "A Modeling of Architecture of System Multi Agent with Petri Nets," IEEE SITIS, 2005.
- [5] G. Quesnel, "Formal and Operational Approach of Multi Modeling and Simulation of Complex Systems: Contributions for Modeling of Multi Agent Systems," Thesis of Doctorate, 2006.
- [6] P. Leitao, A. W. Colombo and F. Restivo, "An Approach to the Formal Specification of Holonic Control Systems. Holonic and Multi-Agent Systems for Manufacturing," *Lecture Notes in Computer Science*, Vol. 2744, Springer-Verlag, 2004, pp. 59-70,
- [7] P. Caspi, "Elements for the Choice of Methods of Development of Critical Software Systems," Research Report N TR-2005-17, November 2005.
- [8] T. Murata, P. C. Nelson and J. Yim. "A Predicate Transition Net Model for Multiple Agent Planning," *Information Sciences*, Vol. 57-58, September-December 1991, pp. 361-384. doi:10.1016/0020-0255(91)90087-B
- [9] A. Idani, "B/UML: Setting in Relation of B Specification and UML Description for Help of External Validation of Formal Development in B," The Grenoble University, Thesis of Doctorat, November 2005.
- [10] G. W. Brams, "Petri Nets: Theory and Practical," Vol. 1-2, MASSON, Paris, 1982.
- [11] M-J. Yoo, "A Componential For Modeling of Cooperative Agents and Its Validation," Thesis of Doctorat, The Paris 6 University, 1999.
- [12] A. Seghrouchni, S. Haddad and H. Mazouzi, "Distributed Observation and Analyses Interactions in a Multi Agent System," Speak French days, IAD and MAS, Nancy, 1998.
- [13] A. Seghrouchni, "Protocol Engineering for Multi-Agents Interaction," *9th European Workshop on Modeling Autonomous Agents in Multi-Agents World (MAAMAW'99)*, 1999.
- [14] J. Ferber, "A Meta-Model for the Analysis and Design of Organizations in Multi agent Systems," *Proceeding of the 3rd International Conference on Multi-Agents Systems (ICMAS'98)*, IEEE CS Press, June, 1998.
- [15] J. Ferber and O. Gutknecht, "For an Operational Semantic of Multi Agent System," Acts of 8eJFIADSMA, Speak French Days of Distributed Artificial Intelligence and Multi-Agents Sytem, 2000.
- [16] R.Duboz, E. Ramat and G. Quesnel, "Multi Agent Systems and Theory of Monetization and Simulation," Acts of French Speaker Twelve Days about Multi-Agents Systems (JFSMA), Paris, 2004.
- [17] B. Sibertin, "High-Level Petri Nets with Data Structure: Petri Nets and Applications", Finland, 1985.
- [18] P. Gruer, V. Hilaire, A. Koukam and K. Cetnarowicz, "A Formal Framework for Multi-Agent Systems Analysis and Design," *Expert Systems with Applications*, Vol. 23, No. 4, 2002, pp. 349-355. doi:10.1016/S0957-4174(02)00070-2
- [19] A. Suna, "CLAIM and SyMPA: A Environment for Programming of Intelligent and Mobiles Agents," Thesis of Doctorat, The Paris 6 University, December 2005.
- [20] C. Maier and Daniel, "Moldt: Object Coloured Petri Nets - A Formal Technique for Object Oriented Modeling. Concurrent Object-Oriented Programming and Petri Nets," *Lecture Notes in Computer Science*, Springer, 2001, pp. 406-427.
- [21] M. Scott, "On Conversation Policies and the Need for Exceptions," Working Notes of the Workshop on Specify and Implementing Conversation Policies, Seattle, May 1999, pp. 19-28.