International Journal of Sciences and Techniques of Automatic control & computer engineering
IJ-STA, Volume 4, N° 2, December 2010, pp. 1402–1419.

Agents Petri Nets: Theory and Application

Adel Mahfoudhi, Borhen Marzougui, Mohamed Abid
CES Laboratory E.N.I.S
University of Sfax
BP 1173-3000, Sfax, TUNISIA
(adel.mahfoudhi@fss.rnu.tn).

Abstract. *The domain of Petri Nets (PN) engineering has been the concern of many researchers. The present paper lies within this context aiming at the definition of a new formalism for the modelling at the multi agents systems (MAS), which is based on agents called Agents Petri Net (APN). That is why, the definitions that treat the internal state of the agent and its behavior are proposed. The suggested mathematical definitions help us to model the interactive systems in a rigorous manner and without any ambiguity. It is through simple examples that our approach is proven to be valid.*

Keyword. *Agents Petri Net, Formal method, Multi Agents Systems, Agent.*

1 Introduction

Since the appearance of the Petri Net [1], scientists have kept on proposing new mod-els, either to improve an already existing formalism or to create a new model. Such formalisms resulting from the studied system types have made the conception more natural, more intuitive and more familiar by means of Petri Net. Indeed, the Petri Net can be considered as a tool of both graphic and mathematical modeling. To model and analyze the discreet systems, particularly the competitive, parallel and non-determinist systems, it is necessary to choose the appropriate type of PN to be used. This type must be capable of modeling the large-sized systems as the multi agents systems rigorously. Such systems allow the coordination of the intelligent agent behavior interacting and communicating in an environment to achieve some tasks or to solve some problems [2].
According to [3], the MAS modeling proves to be applicable for the representation of the actions of the agents and their consequences in the environment that can be of a complex structure and autonomous evolution. Indeed the complexity of the studied system is increasing and the precision, reliability and durability have become difficult factors to reach. Therefore, the integration of a mathematical tool, alongside with graphic tools, offers an exact way to achieve the conception of these systems, particularly the multi agents systems. The objective of the present paper consists in proposing a new type of Petri Net based on the agents that describe the functioning of the multi agents system. The previous work of the Petri Net concentrated on their uses and not on the creation of

This paper was recommended for publication in revised form by the editor Staff.
Edition: CPU of Tunis, Tunisia, ISSN: 1737-7749

the new models as the work of [5], [6], and [7]. The research for a new model has been ignored for a long time. However, there was some work that took into account the extension of some classic types of the Petri Net to attein a more or less generic model to satisfy a need for modeling. 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 should be achieved by an adequate formalism that will be presented in our work.

This paper is organized as follows. The second section describes the formal methods. As for the third section, it invokes the system multi agents and their modeling by the Petri Net. The fourth section explains the limits of the classic PN. In the fifth section, the proposed model titled Agent Petri Net is presented and the definitions formulating this formalism interpreted. Then in the sixth section, a way of correspondence between the two MAS approaches and APN is given. Finally, in seventh section, this document concluded, giving some perspectives.

2 Related Work

The formal methods have been used to assure a level of precision, consistency and quite elevated accurateness. They are based on mathematical foundations to decrease the risks of uncertainty and ambiguity. In the phase of the software conception, the formal methods help a particular language to express the properties descended of the problem specifications very rigorously [8].

However, these methods use the notations and the specific concepts that often generate a weak legibility and a difficulty of integration in the processes of development and certification [9]. The formal specifications are expressed in languages using syntaxes and semantics that are precise and strict. The automatic validations result from a strong theoretical basis, so the integration of several formal methods is indeed difficult [10].

At the level of the system specification, two approaches can be used: one classic and the other formal. Concerning the classic approach, it starts with a functional specification phase, then the general conception followed by a detailed conception and ends with the obtention of the code. As for the formal approach, it starts with a general model that will be verified mathematically to get a refined primary model that undergoes other verification. This treatment is repeated until the obtention of a refined and detailed model that leads to the coding.

The methods of formal specification, which can bring many advantages and allow a better representation of the static and dynamic aspects of the system [10] can be classified into two groups. The first is the approaches based on the states as Z [11], object - Z [12], B [13], VDM [14], etc. Such approaches, among which we can mention : the Petri Networks [1], LOTOS [15], CSP [16], CCS [17], etc., represent the system through two parts. The static part permits the description of constituents and their states, while the dynamic one describes the changes of states.

Based on the events, the latter describes the system by processes communicating independent entities.

3 Multi Agents System and Petri Net

A multi agents system permits to coordinate the behavior of agents interacting and communicating in an environment to achieve some tasks or solve some problems [18], [19] and [20]. It allows the complex task decomposition in simple tasks which facilitates its development, test and updating.
The modeling of the MAS requires a tool verifying both the features and properties of agents and those of the system itself. The different applications of the multi agents system raise four domains: the resolution of problems by emergency, simulation, control of complex system and the environments of man-machine interaction (MMI). That is why, a great deal of work has been carried out on the formalization of the MAS by different formal methods as that of [3], [21] [22], [23] and [24].
Little formalisms have been defined as the automatons of finished states, which prove to be inefficient if the aspects of parallelism are taken into account [5]. The algebraic models of difference equation are inefficient as well, but essentially at the level of the representation of the agents in interactions. Besides, formalism must be capable of expressing the internal state of the agents, their behaviors and the interac-tions between them. In this context, the Petri Net, whose use to model a MAS presents a major contribution, can be mentioned. For example, a Colored Petri Net can model the simultaneous communications of the agents with the help of the functions manipulating some colors. This has been justified in a large amount of work as [25]. So, the Objects Petri Net (OPN) [26] presents a power to model the dynamic aspects of the agents.

4 Limits of the Classic Petri Nets

The classic Object (OPN), Place/Transition (PT) and Colored Petri Nets present a deficiency at the level of their expression when it is about the system of large size as the multi agents system. These systems are characterized by the interactivity of the elements that they compose. Concerning the Colored PN, the classes of colors cannot express the state of the elements such as the system token or the relations between them. However, the OPN can describe the internal state of the tokens but not the relations between them in an efficient manner because it requires the places and the supplementary transitions that put in game the utilized methods. Indeed, an Objects Petri Net models a multi agents system by a quite elevated number of places and transitions by the invocation of a set of methods that essentially describes the behaviors of the agents around their environments.
The multi agents 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 (cf. Fig. 1) [27]. The agents are

autonomous entities whose behavior does not depend on an outside expression, contrary to the objects.

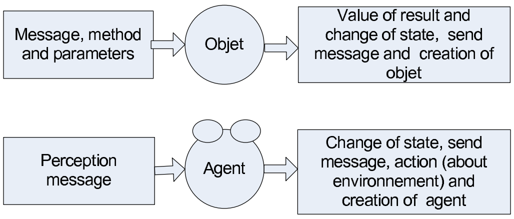
Message, method and parameters
Objet
Value of result and change of state, send message and creation of objet
Perception message
Agent
Change of state, send message, action (about environnement) and creation of agent

Fig. 1. Difference between object and agent [27]

The already achieved work is around the modeling of the SMA by a Petri Net res-pecting the load notebook. It is often needed to make a coupling between two types of Petri Nets to satisfy a possibly determinist aspect in the system specification as the interaction and the communication between the different entities that compose it. Therefore, our idea consists in taking advantage of the properties and features of agents and integrating them in a classic Petri Net. Furthermore, we propose an approach that consists in defining a new model of Petri Net called Agents Petri Nets. First of all, the general definition will be given and then each property with explanatory illustrations will be presented.

5 Proposed Formalism: Agents Petri Nets

An agent is defined as an autonomous entity capable of communicating with other agents to distinguish at least its environment partially and the objects that are situated there, and to have valid or invalid representations about the behaviors of a part or the set of the other agents of the environment [28]. So, contrary to the objects, an agent possesses an autonomous behavior. It is able to take some decisions and establish action plans to accomplish complex activities. All agents do not have this degree of autonomy [7].

5.1 Definition 1: Agents Petri Nets

An Agents Petri Net is defined as being an oriented bipartisan graph that possesses two types of knots (places and transitions).
The bows are ties between these knots that indicate the conditions of activation of a transition. Every transition carries the functions that manipulate the internal state and the behavior of an Agent (Token) in its environment. The distribution of the tokens in the places in a given instant is called marking of the Agents Petri Nets.

A marking gives the state of the system that depends on the interaction between the entities that compose it. The change in the internal 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, the Agents Petri Net is called the 9-uplet:

$$\mathrm{Q} = < P, T, A, Meadow, Post, Pr_j, F, Ft, Env_k > (1)$$

where:

- P is a finite set of places ,
- T is a finite set of transitions,
- A is a finite set of agents,
- $Pre : P \times T \mapsto N$ is the flow-in function,
- $Pre : P \times T \mapsto N$ is the flow-out function,
- Pr_j: meadow condition of clearing,
- $F(A_i, A_j)$: function relation agent that presents the condition of clearing,
- Ft: function agent that uses three variables: $F_t(t_k) =< Per, value, Inter >$
- Envj: Environment of work that describes system multi agents.

5.2 Definition 2: Constraints of an Agents Petri Net

A constraint on an Agent is defined as: $Cont(A_i, K, j)$.
Cont (Ai, k, j) is defined as being a meadow condition of the clearing of a T transition descended of a P place.
In a formal manner, the constraint on an agent exit of a P place can be defined as

$$\forall k \subset I, j \in J, \exists Cont(A_i, K, j) = b \ (2)$$

where:

- I: a set of tokens of a place,
- J: a set of places of a Petri Net,
- K: under set of I,
- j: number of place belonging to the network,
- A_i: Agent of indication (number) i,
- b: Boolean (0 or 1).

5.3 Definition 3: Function Meadow condition

Either $Cont(Ai, K, j) = b$
Either nk: number of coins elements - together with K. For a number nk of agents that enters into an environment:
$Cont(A1, k, j)$ and $Cont(A2, k, j)$ and $Cont(Ank, K, j) = b$
That gives:

$$\prod_{i=1}^{i=nk} Cont(A_i, k, j) = b \ (3)$$

The function meadow thus Prj condition descended of a P place of indication j is defined as:

$$Prj = \prod_{i=1}^{i=nk} Cont(A_i, k, j) = b \ (4)$$

By hypothesis, an agent Ai debit is hired only Approx. in only one environment. Card (Env(Ai thus)) can be defined as =1.

Interpretation of the possible values of Prj

- If Prj = 0 then the condition of clearing is not valid and in this case at least an agent that did not respect the principle of uniqueness exists. Of course it is al-ready engaged in another environment.
- If Prj = 1 then the condition of clearing is valid and in this case, it can be guaranteed that all the agents in question respect the principle of uniqueness.

Illustration (cf. Fig.2) It is supposed that:

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

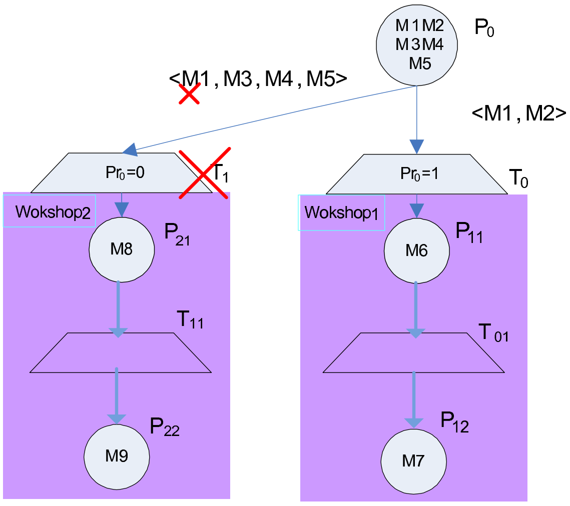

Fig. 2. Illustration of the Function Meadow condition

- Case 1: the two machines M1and M2 belong to the same workshop (Environment): Workshop1. In this case their use is permitted: meadow Pr0=1 condition,
- 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, the transition T1 cannot be cleared.

5.4 Definition 4: Function of adherence (relative to an agent)

This function gives birth to a relation between an agent and its environment. The engagement of an Agent A_i in an Env_j environment describes a criterion of adherence; i.e., the number of times that this agent has been engaged in Env_j. Indeed, it offers more explanation mechanisms and minimizes the difficulties with the tasks that require knowledge of the world (Env) that can only be got by the memorization or the reasoning rather than by the perception. The definition of Ferber [2] is used not only to show that a cognitive agent has the capacity to reason on the representations of the world, but also to memorize some situations as well as to analyze them. Such definition is also used to foresee some reactions possible for any action and to draw the conducts of the future events in order to plan their own behavior.
In a formal manner, we define the adherence function of an agent A_i, in an Env_j environment Apa_i noted by:

$$\forall A_i \in A \text{ et } Env_j \subset Env,\ \exists Apa_i = Apa(A_i, Env_j, b, d)\ (5)$$

with:

- $d \in Z^+$
- b: constraint =Prj (b=0 or 1): the engagement of A_i in Env_j,
- d: degree of adherence: a set gives the number of times that the agent A_i has been engaged in Env_j.

Interpretation of the adherence function
At any time, this function gives a relation description between the agent and the very specific environment. It also guarantees the updating of the basis of an agent's knowledge. An agent's reaction depends on its environment. The evolution of Agents Petri Net depends on the system to be studied which implies that each agent looks for the competency of another. That is why it must interpret the value of **d**.
Illustration: (cf. Fig. 3) Let us take the example of Figure 2 with some modifications:

- Before the clearing of T0 and T1, agent M1 admits as a degree of adherence 0 for the Workshop1 environment and Workshop2, respectively:

  ApM1=Apa(M1, Workshop1, 1, 0)
  ApM1=Apa(M1, Workshop2, 1, 0)

- After the clearing of T0, agent M1 will have as degree of adherence 1 for the Workshop1 environment and 0 for Workshop2:

  ApM1=Apa(M1, Workshop1, 1, 1)
  ApM1=Apa(M1, Workshop2, 1, 0)

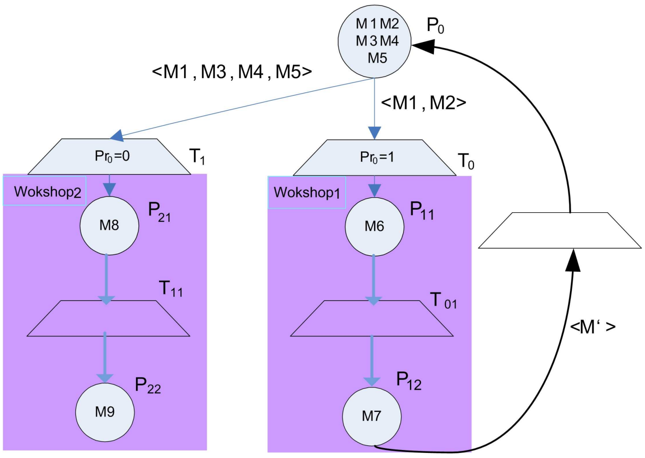

Fig. 3. Illustration of the adherence Function

- Agent M1 leaves his Workshop1 environment: the clearing of the T2 transition will allow him to be free.
- After the shooting of T1, agent M1 will have as a degree of adherence 1 for the Workshop1 environment and Workshop2, respectively:

$$\text{ApM1}=\text{Apa}(\text{M1, Workshop1, 1, 1})$$
$$\text{ApM1}=\text{Apa}(\text{M1, Workshop2, 1, 1})$$

The function of adherence of every agent allows us to deduce the Marking of a RPA (Fig. 4).

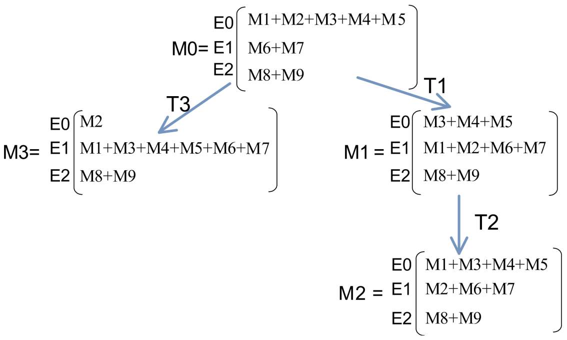

Fig. 4. The evolution of the marking in an Agents Petri Net

5.5 Marking of an Agents Petri Net

A marking of an Agents Petri Net R is a family indexed by P. It is a Vector column whose composition is the number of marks in the place Pi at a given instant. A marked Petri Net of agents is a couple $< R, M0, E0 >$ in which R is an Agents Petri Net, M0 is a marking of R named initial marking and E0 is called Environment initial. A p place is a meadow condition of a t transition if an oriented arc from p to t exists. Symmetrically, p will be a post condition of t if an oriented arc from t to p exists. The marking of a Petri Net evolves in each transition activation. Such an event is governed by rules of clearing: a transition can only be activated if the marking of the set of the places meadow condition allows it. With the activation of a transition, there is a use of a number of marks adequate in the meadow condition places and production of marks in the post condition places.
If the marking *Mi* is open to leave from the *M0* marking after clearing the sequence of *S* transition = *(S1, S2,...,Si)*, then *Mi=M0+CS*.
Where *S* is the vector of n-size in which every *SJ* represents the number of time in which the transition *tj* is cleared in the S sequence. *C* is the adjacency matrix of size n x n.
In the Agents Petri Net, the clearing of a transition implies the evolution of the system. A sequence of clearing gives a casual manner and a relation history between two or several agents. The intelligence of such an agent is therefore based on its capacity to interpret this sequence. Besides the following the following transition, the agent must take into consideration the visited transitions; that is to say the actions that have already been realized. Hence, the marking of the Agents Peri Nets presents a dynamic description of the agents in an SMA and a help of choice for the agents; i.e., the choice of expertise criterion of the other agents.
The *E* and *T* sets are finished and discreet; the Meadow applications and the Post under matrix shape can therefore be represented (Fig. 5).

$$\textbf{Input Matrix :}\quad \text{Pre} = \begin{array}{c|ccc|} & \textbf{T1} & \textbf{T2} & \textbf{T3} \\ \textbf{E0} & M1+M2 & M1+ M3+M4+M5 & 0 \\ \textbf{E1} & 0 & 0 & M1 \\ \textbf{E2} & 0 & 0 & 0 \end{array}$$

$$\textbf{Output Matrix :}\quad \text{Post} = \begin{array}{c|ccc|} & \textbf{T1} & \textbf{T2} & \textbf{T3} \\ \textbf{E0} & 0 & 0 & M1 \\ \textbf{E1} & M1+M2 & 0 & 0 \\ \textbf{E2} & 0 & M1+M2+M3+M4 & 0 \end{array}$$

$$\textbf{Incidence Matrix :}\quad C = \begin{array}{c|ccc|} & \textbf{T1} & \textbf{T2} & \textbf{T3} \\ \textbf{E0} & -(M1+M2) & -(M1+M3+M4+M5) & M1 \\ \textbf{E1} & M1+M2 & 0 & 0 \\ \textbf{E2} & 0 & M1+M2+M3+M4 & 0 \end{array}$$

Fig. 5. Input, Output and Incidence matrices

5.6 Definition 5: Function of adherence (relative to an environment)

The creation of the function of Apai adherence of an agent Ai in an Envi environment allows us to find a function of Apej adherence inversely. This new function describes the set of the agents that belong to the same environment j with certain degree of adherence di.
The function of adherence related to an Envj environment can be defined as:

$$Ape_j = Ape(Env_j, \bigcup_{i=1}^{i=nk}(A_i, d_i)) \ (6)$$

where

- nk : number of agents of the environment,
- Ai: Agent of indication i,
- di: degree of agent adherence of i indication.

Thus, such function can be simplified as:

$$Ape_j = Ape(Env_j, \bigcup_{i=1}^{i=nk}(d_i)) \ (7)$$

Illustration:
From the RPA of Fig. 2 the adherence degree of every environment can be deduced:

$$ApeWorkshop1 = Ape(Workshop1, \bigcup_{i=1}^{i=5}(d_i))$$
$$ApeWorkshop2 = Ape(Workshop2, \bigcup_{i=1}^{i=5}(d_i))$$

with 5 the number of used machinery.
In (Fig. 6), the adherence matrix can be deduced as:

$$\begin{array}{r|ccccc|}
 & \textbf{M1} & \textbf{M2} & \textbf{M3} & \textbf{M4} & \textbf{M5} \\
\text{Workshop 1} & 1 & 1 & 0 & 0 & 0 \\
\text{Workshop 2} & 1 & 0 & 1 & 1 & 1
\end{array}$$

Fig. 6. adherence matrix

5.7 Definition 6: Moderator Agent

An agent is said to be a moderator if it is important by contribution to another . The important term indicates that the moderator dominates at the time of

a communication, or possesses a hierarchical degree (hd) that is less elevated (dh=2 dominates dh=3).
An agent is said to be a total moderator if it is important by contribution to all the agents of its environment. Thus, it possesses a hierarchical degree hd that is equal to 1 (Fig. 7).

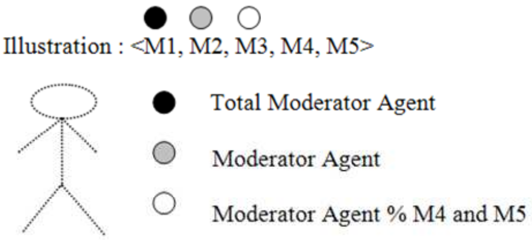
Illustration : <M1, M2, M3, M4, M5>
Total Moderator Agent
Moderator Agent
Moderator Agent % M4 and M5

Fig. 7. Moderator Agent

5.8 Definition 7: Function of the two-order relation

A two-order relation can be defined as a function admitting two entries E1 and E2 and only one Boolean exists of the value S. The entry E1 is imperatively moderator. This function presents a meadow condition of a transition clearing. Thus, this relation is defined by the function F (E1, E2) =S.
Let us have two agents Ai and Aj in the same environment Env.

$$A_i \in A, \forall A_j \in A \exists \text{ a Function as } F(A_i, A_j) = S$$

where

- A: set of the environment agents,
- Ai: is a moderator agent,
- Aj: is not a moderator agent,
- S: Boolean value sending back 1 or 0.

Thus, this function can be generalized to get a function of n-order relation: set of n agents, a function F as F (Ai, Ai, An) =S
Interpretation of the possible values of S:

- If S = 0 then no relation exists between the two agents that communicate between themselves. In this case, the non-moderator agent cannot enter in relation with the agent moderator voluntarily, or forced by the agent moderator of total order, or because it is already occupied.

- If S=1 then a relation between the two concerned agents can be established. In this case, the agent moderator asks for the establishment of a communication with another agent that is called non-moderator and that meets this requirement(Fig. 8).

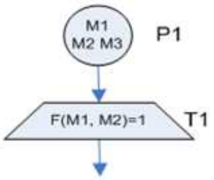

Fig. 8. Function of the two-order relation

5.9 Definition of the Agent Function: Ft

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

$$Ft(tk) =< Per, Inter, Value > .$$

Interpretation of the possible values of Function Ft :

- Initially, $Ft(tk) =< 0, \Phi, 0 >$ implies that there is not any interaction between the agents. If the value of Per = 0, then we directly have Inter = 0. Never can we have Per = 0 nor Inter = 1. The Value $=\Phi$, in this case no action is triggered and the previous situation of the agents is made safe,
- In the course of the clearing of the transition tk, there will be a change of val-ues between the agents. In this case, Per takes the value 1, Inter takes the value 0 and Value defines the action or the task to be achieved. The relation of the already defined order gives the sense of transfer of the information. So:

$$Ft(tk) =< 1, Value, 0 >$$

- After the clearing of the transition tk, Inter takes the value 1; it indicates that the action has been achieved with success. So:

$$Ft(tk) =< 1, Value, 1 >$$

An example is given in Fig 9.

- Initial state of treatment: the two Machine M1 and M2 are waiting: Per=0, $Value = \phi$ and Inter=0
- The M2 machine wants to pass the P4 piece to be treated by M1: Per =1 Value =M2.Passe[P4] Inter=0
- The M1 machine accepts the demand for M2: Per =1 Value =M2.Passe [P4] Inter = 1.

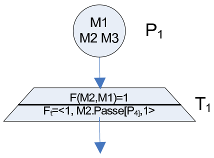

Fig. 9. Example of two-order relation

Interpretation of the possible values of Per :
Per is a Boolean value that puts in relation two agents Ai and Aj through a transi-tion T. If Per = 0, then the agent Ai does not send information to agent Aj,
So Per = 1, then the agent Ai send information to agent Aj.
Interpretation of the possible values of Inter:
Inter is a Boolean value that validates the transmission or the exchange data be-tween two agents Ai and Aj through a transition T.

- If Inter=0, then the agent Aj does not receive the information of agent Aj or it refuses it,
- If Inter = 1, then the agent Ai received information sent by agent Aj. Indeed the task has been achieved. Then it is a value of validation.

Interpretation of the possible values of value:
Value is an action achieved by the two agents *Ai* and *Aj*. This action represents the transference of data from the moderator agent Ai toward the other agent Aj. It is a data structure going from the simplest (set, real, character, Boolean) to the most complex (table, matrix, object, class of object, composing, Agent). Value modifies the behavior of an agent.

6 From the Multi Agents Systems Towards the Agents Petri Nets

A way of correspondence between the two approaches according to very determined features will be given in the following table.

Characteristic	**Multi Agents Systems**	**Agents Petri Nets**
Name	Agent State of the system Set of rules Set of relations of actions Agent Administrator	Token Place Meadow condition (Prj) Transition Agent Total Moderator
Class	Reactive Agent Cognitive Agent Hydride Agents	no Moderator Agent Moderator Agent Total Moderator Agent
Autonomy	Interaction between agents	Function of two or n relation
Reactivity	Agent - Agents	Function Agent Ft=<*Per, Valeur, Inter*>
Heterogeneity	Agent – Environment	Related to an agent adherence Function:
Sociability	Environment – Agents	Related to an agent adherence function (Apai) and its environment : Apej=Ape (Envj,$\bigcup_{i=1}^{i=nk}(d_i)$
Intelligence	Behaviour, capacity of interaction	Exploitation of the possible values of *Per*, *Inter* and *Value*

7 Discussion and Contributions

It is thanks to the major importance brought by the agents and their contributions to the modeling level of the complex systems that we model systems using several peculiarities such as mobility, cooperation and negotiation. The Petri Net to Agents is concerned with the addition of several other features that are already known in other types of the Petri Nets:

- Simplicity: it is clear that we model using graphic presentations that increase the legibility and ease of understanding of the models. Indeed, the designers will find no difficulty to understand and use it.
- Migration: from another model of the Petri Nets, we can make a fast and simple migration towards an Agents Petri Net. Indeed the new model uses essentially the properties of a colored Petri Net and that in Objects.
- Precision: besides the mathematical tools that are already used by all other models, new mathematical definitions will be added to the agents Petri Nets. These definitions characterize the agent and the SMA. In fact, we have assured the properties and the characteristics of the agents while using functions either to describe the internal architecture of the agent, or to describe its behavior within a multi- agents system. These functions aim at the lucid presentation of the state of each agent, its immediate behavior, the load of the system, the ongoing actions of the realizations and their history.

- Evolutionary: a multi-agents system is characterized by its dynamic aspect. An agent Petri Net is capable of evolving in a simple and quick way. Indeed the addition or the integration of a new agent in the system and the implementation of these actions is a simple task using the order functions.
- Complexity: the modeling of the large-sized systems was so difficult; the agent Petri Net makes a significant reduction at this level. Indeed with the realized tests, we can assure an important reduction of a modeling that has already been realized by another Petri Net.
- Data structure: at the programming level, the agents RdP uses various data structures starting from the simplest to the most complex. This facilitates the programmer's task. At the modeling level, we can indicate the structures of adequate data that is going to be used.
- Knowledge base: the agents RdP favors, in an intelligent way, the basic update of knowledge of every agent participating in an environment. Indeed every transition describes a task or an action that can modify the internal state or the behavior of the participating agents. So, the AAP (Architecture Agent Petri) of an agent gives to each system state a description of its behavior, after having updated its history of actions or events. The knowledge base contains information relative to the environment, to the communication with the other agents, the indications of the visited places, the transitions made and the actions realized by this agent.
- A token is an agent: it represents the big contribution of the Agents Petri Nets. Indeed a token presents the agent as it is involved in its environment, unlike the RPO, whose objects manipulation always remains static, and the evolution does not directly depend on the token but on all the actions and states in places. The manipulation of an object consists in defining all operations. An operation can modify the internal state of the object but does not describe the direct relation with the other entities. In our case, we can treat the internal state of the token, its behavior, its properties (such as sociability, intelligence, autonomy), in parallel, without the need for the addition of other places or transitions.

8 Conclusion

The present paper introduces a new formalism stemming from Petri Nets and multi-agents systems, called "Agents Petri Nets". Our approach lies within the framework of research work treating the formal methods for the modeling of the multi-agents systems. Besides, all the definitions pertaining to this new formalism are given, taking advantage of the characteristics of the agents and multi-agents systems. Indeed each token of a place represents an agent and the transition is capable of a set of functions that describes, in particular, the condition of its firings and relations between the agents. Compared to the others, the major contribution of an Agents Petri Net is the power of expression, capacity of modeling of the interactions between the agents, the notable reduction of the size of network and the gain at the modeling time level. The definition of this model helps us to effectively model the internal state of the dynamic behavior of the agent in an SMA.

Références

1. Petri, C.A.: Communication with Automats, thesis of doctorat at the Boon University, (1962).
   AAAAA
2. Fougres, J.: A cognitive architecture of communicating agents in complex information systems, M3M Laboratory, the Belfort-Montbliard University of Technologies, (2002).
3. Ferber, J.: The multi agents system: towards a collective ,intelligence InterEditions, Paris(1995).
4. Tagne, E., Viho, C, Tonye, E. and Akono, A.: Modelling of architecture of system multi agets with Petri Net, IEEE SITIS, (2005).
5. Kefi, M., Korbaa, O., Ghedira, K., Yim, P.: Formalization of muli agents model with Petri Net: application to problem of storage of centenaries, 6th French speaker Conference of Modeling and Simulation-MOSIM'06, Rabat, April (2006).
6. Caspi, P.: Elements for the choice of methods of development of critical software systems, Research report N TR-2005-17, November (2005).
7. Quesnel, G.: Formal and operational approach of multi modelling and simulation of complexe systems: contributions for modelling of multi agents systems, thesis of doctorat, December (2006).
8. Lamy, P., C.Blaise, J.: Use of formal methods, J'automatise N42, September-October (2005).
9. Idani, A .: B/UML: Setting in relation of B Specification and UML description for help of external validation of formal development in B, the Grenoble 1University, thesis of doctorat, November, (2006).
10. Gervais, F.: EB4: Towards a combined method of formal specification of information sys-tems, thesis of doctorat, the Sherbrooke University (Qubec), Canada, June (2004).
11. Spivey, J. M.: The Z Notation, a Reference Manual. Prentice Hall, 2nd edition, (1992).
12. Smith, G.: The Object-Z Specification Language, Kluwer Academic Publishers, (2000).
13. J.R. Abrial, The B-Book: Assigning programs to meanings, the Cambridge Press University, (1996).
14. Jones, C.B.: Systematic Software Development using VDM, Prentice Hall, (1990).
15. Bolognesi, T. and Brinksma, E.: Introduction to the ISO specification language LOTOS, Computer Networks and ISDN Systems, (1987).
16. Hoare,C.A.R.: Communicating Sequential Processes, Prentice-Hall, (1985).
17. Milner, R.: Communication and Concurrency, Prentice-Hall, (1989).
18. Yoo, M-J.: A componential for modeling of cooperative agents and its validation, Thesis of doctorat, the Paris 6University, (1999).
19. Ferber, J.: 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).
20. Seghrouchni, A.: Protocol Engineering for Multi-Agents Interacion, 9th European Workshop on Modelling Autonomous Agents in Multi-Agents

21. World(MAAMAW'99), (1999).
22. Ferber, J. and Gutknecht O.: For an operational semantic of multi agents system, Acts of 8eJFIADSMA, Speak french days of distributed artificial intelligence and multi-agents sy-tem, (2000).
23. Gruer, P., Hilaire, V., Koukam, A. and Cetnarowicz, K.: A formal framework for multi-agent systems analysis and design, Expert Systems with Applications, 23(4):349-355.(2002).
24. Duboz, R., Ramat, E., and Quesnel, G.: Multi agents Systems and theory of monetization and simulation, Acts of French speaker twelve days about Multi-Agents Systems (JFSMA), Paris, (2004).
25. Suna, A.: CLAIM and SyMPA: a Environment for programming of intelligent and mobiles agents, thesis of doctorat , the Paris 6 University, December (2005).
26. Seghrouchni, A., Haddad, S. and Mazouzi, H.: Distributed Observation and analyses inte-ractions in a multi agents system, Speak French days, IAD and SMA, Nancy, (1998).
27. Sibertin, B.: High-Level Petri Nets With Data Structure: Petri Net and Applications, Finland, (1985).
28. Briot, J.P.: Introduction to agents, DEA IARFA, the Paris 6 University - CNRS, (2002).