A Timed Colored Petri-Net-based Modeling for Contract Net Protocol with Temporal Aspects

Djamila Boukredera
Laboratoire des mathématiques appliquées
Université Abderrahmane Mira
Béjaia, Algérie
boukredera@hotmail.com

Ramdane Maamri
LIRE Laboratory
Université Mentouri
Constantine, Algérie
rmaamri@yahoo.fr

Samir Aknine
GAMA Laboratory
Université Claude Bernard, Lyon 1
Lyon, France
samir.aknine@univ-lyon1.fr

***Abstract*—Contract Net Protocol (CNP) is a high level communication protocol. It is one of the most widely used in multi-agent system (MAS) to resolve decentralized task allocation problem. The main aim of the protocol is to facilitate contract negotiation between a manager agent and many contractor agents. A lot of works have been done for the verification of the protocol and its extensions, but there still lacks a formalism for representing temporal interaction aspects which are an essential parameter in the protocol modeling. This paper proposes to use Timed Colored Petri Nets (TCPN) to model correctly and formally this temporal dimension often defined as interaction duration and message deadlines. We will verify by means of simulation techniques and state space analysis important properties namely model correctness, deadline respect, absence of deadlocks and livelocks, absence of dead code, agent terminal states consistency, concurrency and validity.**

Keywords-Negotiation protocols; Contract net protocol; Multi-agent systems; Timed Colored Petri Nets.

I. Introduction

A multi-agent negotiation protocol is a specification of the rules that govern interaction among negotiation agents. Formal modeling as well as validation and verification of such protocols are of crucial importance in the design of automated negotiation systems. Based on FIPA standards [2], the CNP, originally proposed by Smith [8], is one of the most popular interaction protocols used in diverse negotiation contexts. Developed to resolve decentralized task allocation, the CNP represents a distributed negotiation model based on the notion of call for bids. In this protocol, agents can dynamically take two roles: manager or contractor (initiator or participant according to FIPA terminology). CNP is currently used as the basis for developing more complex agent negotiation protocols, that is why it is important to analyze this protocol and to verify that it satisfies various key properties before implementation. Several formal models were proposed in the literature [1], [4], [12]–[14], [17], [18], but few works tackled the modeling of temporal interaction aspects which are specified by FIPA.

This paper addresses this issue and proposes to use Timed Colored Petri Nets (TCPN) to model formally the CNP with two temporal constraints:

- Deadlines: it is a time constraint for message exchange. They denote the time limit by which a message must be sent. Once the deadline expires, the manager starts the evaluation of the received proposals. All proposals which arrive after the due time will be considered to be invalid and consequently ignored.
- Duration: it is the interaction activity time period. It represents the time elapsed between the sending of a request message and the reception of the response. Duration includes two periods: transmission time and response time (task duration).

To model these issues, we adopt TCPN models because, besides their simplicity, they are particularly suitable in the modeling, simulating and analyzing of timed concurrent systems and, moreover, they enhance powerful tools for validation and verification. Our model proposes the modeling of these time constraints as well as the interaction sequence following the contracting phase. Our work contributes to the formal design of the temporal interaction aspects for negotiation systems. This contribution can be enumerated as follows: firstly, we present and we implement the proposed model using CPN Tools. We analyze it by means of the simulation and the state space techniques for various values of the protocol parameters namely the deadline and the number of participants. Secondly, we prove that the above mentioned key properties of the protocol are satisfied.

The rest of this paper is structured as follows: Section II introduces the contract net protocol. In Section III, the temporal interaction constraints are described and a formal definition of TCPN is presented. Section IV shows how the CNP enriched by temporal aspects is modeled in terms of TCPN. We verify this model in Section V. Lastly, Section VI concludes the paper and gives some perspectives.

II. The Contract Net Protocol

In CNP as described by FIPA [2], a manager and participants interact with one another to find a solution for a problem through a four-stage negotiation process. The manager initiates the negotiation process by issuing a Call

 ISBN: 978-1-61208-202-8

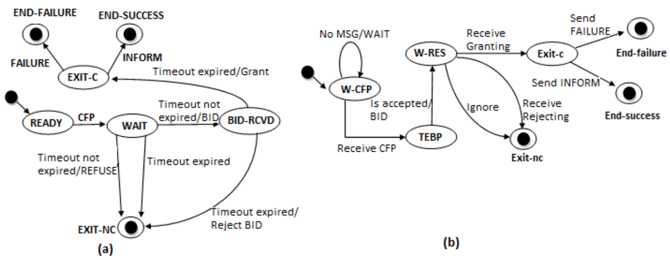

Figure 1. Internal behavior of the manager and the participant agents.

Table I
REPRESENTATION OF STATES.

Manager (Initiator)	Participant
READY (READY to send a CFP)	W-CFP (Waiting for CFP)
WAIT (Waiting for bids or for time-out)	TEBP (Task evaluation and bid preparation)
BID-RCVD (Bid received)	W-RES (Waiting for result)
EXIT-NC (EXIT with no contract)	Exit-nc (exit with no contract)
EXIT-C (EXIT with contract)	Exit-c (exit with contract)
END-SUCCES (END of negotiation with SUCCESS)	End-success (end of task execution with success)
END-FAILURE (END of negotiation with FAILURE)	End-failure (end of task execution with failure)

For Proposals (CFP) announcing the task specification to a number of potential participants. The CFP includes a deadline by which the participants must respond with bids. Participants evaluate the CFP and decide whether to answer with a refusal message or with a proposal to perform the task. Once the deadline expires, the manager evaluate all the received proposals (in due time) and, in turn, awards the contract to the most appropriate participant which becomes a contractor. The manager ignores any proposal that arrives beyond the deadline. The contractor performs the task and sends to the manager an informing message, which can be an error one in the case of a failure. Consequently, the negotiation process includes several scenarios depending on whether the bid process ends with or without a contract, and as the execution of the task ends with or without a success. Therefore, the manager and the participants can reach various states during this process. We suggest to represent the internal behavioral of both types of agents by means of AUML2 statesharts diagrams [3] . These diagrams define the different states that will be later used in the TCPN model of the protocol. Figure 1 (a) and Figure 1 (b) illustrate respectively the internal behavior of the manager and the participant agents. Table I summarizes the various states and their semantics.

III. MODELING TEMPORAL ASPECTS OF INTERACTION

Two temporal interaction aspects are specified by FIPA [2]: duration constraint and deadline constraint. The first one is the interaction activity time period which includes the two periods: transmission time and response time. In our model, we have assumed that the transmission time is infinitesimal and can consequently be ignored. On the other hand, the response time would depend on the defined deadline and hence we would propose a function to estimate it. The second temporal aspect, deadlines, is a time limit for the message exchange. The manager sets a time constraint (timeout) on the CFP message and participants must respond within this time limit, otherwise the response will be ignored. It is a synchronous communication with a limited waiting time. The expiration of the deadline implies the execution of other alternatives, that is why we propose to model this constraint by a timeout mechanism. We adopt using TCPN techniques to represent these temporal interaction aspects. In doing so, we assume a global clock.

A. *Timed CPN*

The concept of time was not explicitly provided in the original definition of Petri nets. As described in [10], we distinguish three basic ways of representing time in CPN: Firing Durations (FD), Holding Duration (HD) and Enabling Duration (ED). Choosing one of these three techniques depends strongly on the system to be modeled and its specifications. We should note, however, that it is natural to use HD technique in modeling the most processes as transitions represent operation event which, once starts, it does not stop until it ends. It is exactly the case of the system we are modeling. In HD technique, there are two types of tokens: available and unavailable. Available tokens can enable transitions whereas unavailable ones cannot. When a transition, which is assigned a duration, fires, removing and creating tokens are done instantaneously. However, the created tokens are not available to enable new transitions until they have been in their output place for the time specified by the transition which created them. For more details concerning these three techniques of time modeling, the reader can refer to [10]. CPN versions which use HD technique define implicitly the notion of tokens's unavailability by attaching to these tokens a timing attribute called a timestamp.

B. *Formal definition of TCPN with Holding Durations*

To represent tokens with timestamps we adopt the notation given by [11]. Each token carries a timestamp preceded by the @ symbol. For instance, 2 tokens with timestamp equal to 10 are noted 2@10. The timestamp specifies the time at which the token is ready to be removed by an occurring transition. Timestamps are values belonging to a Time Set TS which is equal to the set of non negative integers N+. The timed markings are represented as collection of timestamps, there are multi-sets on TS: TS_{MS}. The formal definition of TCPN using holding durations is as follows: TCPN = (Σ, f, M_0) where:

- Σ is a colored PN as described in [11]
- **f**: T $\rightarrow$TS represents the transition function which assigns to each transition t $\in$ T a non negative determinist duration

Table II
REPRESENTATION OF MESSAGES IN THE TCPN MODEL.

Messages issued by the manager	Messages issued by the participant
CFP (Call For Proposals)	BID (BID)
GB (Grant Bid)	REFUSE (REFUSE CFP)
RB (Reject Bid)	FAILURE (task Execution FAILURE)
CB (Cancel Bid)	INF-DONE (INForm-Done)
	INF-RES (INForm-RESults)

- **M**: P $\rightarrow TS_{MS}$ is the timed marking, M_0 represents the initial marking of TCPN.

To determine whether tokens are available or unavailable, we define functions over the marking set M. So, For a marking M and the given model time (global clock), we have:
m: $P \times M \times TS \rightarrow$ N which defines the number of available tokens and n: $P \times M \times TS \rightarrow$ N which defines the number of unavailable tokens for each place of the TCPN model at a given instant k where k and the model time belong to TS.
There are several computer tools which perform automatic validation and verification of Petri net models. Nevertheless, only CPN Tools permits, besides time representation, the modeling of high level petri nets particularly colored and hierarchical ones.

IV. TCPN MODEL OF THE CONTRACT NET PROTOCOL

When modeling a protocol, there are several design requirements and key characteristics that this protocol should satisfy. Authors in [13] have summarized these issues in 5 factors: state set, role set, rule set, action set and message set. By analogy with our case, study Table I describes the various states that negotiation process should reach and Table II defines messages exchanged between the manager and the participants. This section highlights our contribution and presents how Contract Net Protocol extended with the temporal aspects described in section II can be modeled as TCPN using CPN Tools. When creating the model, we have assumed some assumptions such as the reliability of the communication channel, and that participants have to reply to the CFP. Moreover, when modeling the interaction following the contracting phase, we should not take into consideration task duration, given that this work focuses on temporal interaction aspects. The manager starts evaluating bids after deadline expiration and lastly, the details of messages exchanged are excluded for an abstraction concern.

A. Declarations

Being inspired by [1], our TCPN model is readable and has a compact structure: For each type of agents, we use a single place which would store all its possible states. Similarly, we distinguish two places which represent a reliable channel for both directions of the communication. Figure 2, taken directly from CPN Tools, shows all the declarations used in the model.

Figure 2. Declarations for the TCPN model of the CNP.

B. Model structure

Figure 3 shows the TCPN diagram of CNP. The manager with the timeout mechanism is modeled in the left, the participants in the right. They communicate via a reliable not ordered channel represented by the two places INIT2PART and PART2INIT. The place INIT2PART only contains messages issued by the manager to the participants. Respectively, PART2INIT only contains messages of the participants to the manager. In this model the timed messages carry timestamps indicating when they should be available. Initially, the manager is in the state READY with respect to all the participants. Whereas, all the participants are in the state W_CFP. The place GRonly1 contains one token GR1 and all the other places are initially empty.

V. VERIFICATION OF THE MODEL

Verification is a method to exhaustively examine a design and check to make sure certain predefined key properties are met. There are several software tools to automate this task, however, CPN Tools [9] is currently the most used tool for high level Petri nets particularly for the timed colored ones (TCPN). This tool helps us to assess the correctness of the model.

A. Simulation

Using CPN simulator, we have conducted several automatic and interactive simulations which help us to identify and resolve several omissions and errors in the design. In addition to that, these simulations show that the protocol always seems to terminate in the desired coherent state. That is, it works correctly. Simulation also shows that the characteristics such as concurrency and validity are satisfied. This makes it likely that the protocol works

 ISBN: 978-1-61208-202-8

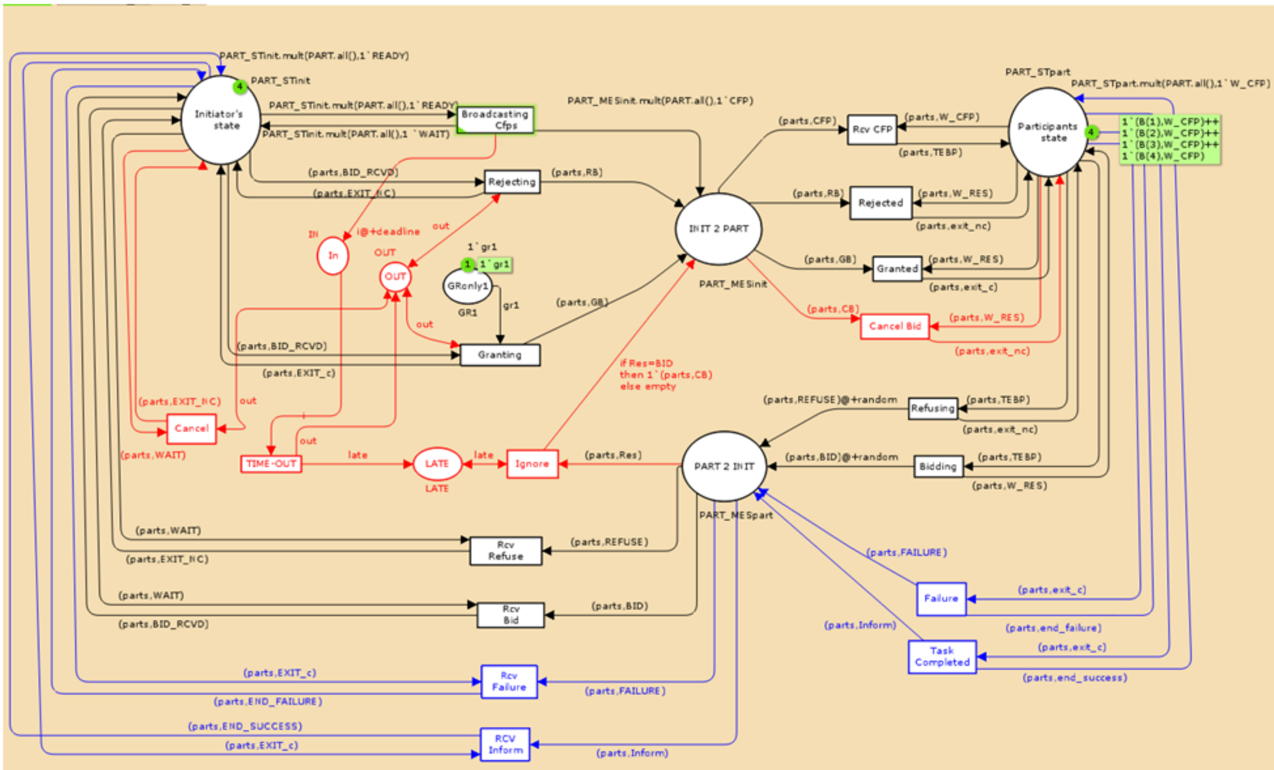

Figure 3. TCPN diagram of the contract net protocol.

correctly but it cannot guarantee that simulation covers all possible executions. That is why simulation cannot be used to verify other functional and performance properties such as the absence of deadlocks and others. However, State space analysis techniques allow us to verify if the system satisfies these behavioral properties.

B. State space analysis

With regard to untimed CPN models, calculating timed state space is a non trivial task and can be quite difficult and time consuming. This is because the reachability graph is too large and can be infinite even if the state space of the corresponding untimed CPN model is finite. This is due to the fact that several timed markings including global clock and timestamps can be different even if the corresponding untimed markings are identical. That is why we have to use some CPN ML queries to verify some properties.

Model Correctness. In this section, we verify the absence of deadlocks and the consistency in beliefs between the manager and the participants. Table III presents the state space analysis results. It shows the properties of the state space obtained by varying the parameter MaxParts from 1 to 4 and the parameter deadline from 1 to 5. The analyzing of the property DeadMraking allows us to verify the model correctness. Each dead marking corresponds to a terminal state of the negotiation protocol. All dead markings are obtained after the deadline expiration, ie, from t=d to t=2*d-1 (proposed estimation for the participants response time), for each discrete value of t belonging to this interval. For any value of MaxParts, one of the dead markings corresponds to an end of negotiation without a contract. In this marking, all the participants are in the state exit_nc and the manager in the state EXIT_NC with respect to all the participants. This is illustrated by the marking 14 in Figure 4. The description of this node shows that the place GRonly1 has still the token GR1 implying that none bid had been granted. The place In is empty, signifying that the deadline has expired and the timeout has fired. This particular dead marking is acceptable because the manager may reject all the bids or may not receive any bid in the due time. Among the rest of the dead markings, we distinguish those calculated at t=d and those obtained at t>d:

At t = d and for any values of MaxPArts: besides the particular dead marking mentioned above, the dead markings calculated at this time corresponds to the end of negotiations where a contract has been awarded to one participant (i=1..MaxParts) while the rest of negotiation with the rest of participants has ended without a contract. Therefore, P_i changes state to exit_c, performs the task which can ends by a success or a failure. P_i can, then, be in the state end_success or end_failure respectively. At the same time, the manager which was in the state

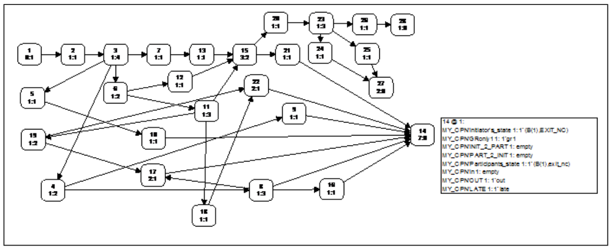

Figure 4. State space for (MaxParts = 1 et d = 1).

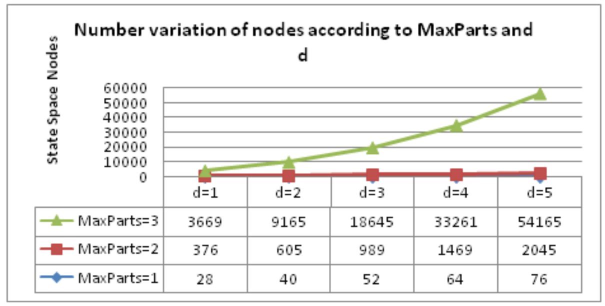

Figure 5. Number variation of the reachability graph nodes according to Maxparts and the deadline.

EXIT_C with respect to P_i (and EXIT_NC with respect to the rest of the participants) changes to END_SUCCESS or END_FAILURE with regard to P_i. All the other participants P_j (j≠i) are in the state exit_nc. Thus, we can deduce that at t=d and for any value of MaxPArts we have:

NumberDeadmarkings = (2*MaxParts +1).

The rest of the dead markings is calculated at t>d which correspond to scenarios after the fire of the timeout where at least one participant is not in the due time. Two cases can be distinguished: a particular case of a single participant (MaxParts=1) and a general case of several participants (MaxParts $>$ 1):

t $>$ d and MaxParts = 1: this is particular because the single participant may miss the deadline and, consequently, changes state to exit_nc because of the canceling of its late response. The manager is in the state EXIT_NC with respect to this participant. This corresponds to the end of negotiation without a contract caused by the deadline overrun. This dead marking is reached for any discrete value of t where d<t>=2*d-1, ie, (d-1) times and thus we deduce:

$$NumberDeadmarkings = 2 * MaxParts + d \qquad (1)$$

which is equal in this case to (2+d).

t $>$ d and MaxParts $>$ 1: all the dead markings calculated after the timeout and for each discrete value in the interval (d..2*d-1) are similar to those obtained at t=d. The only difference is that the global clock values and the timestamps of the tokens differ. Thus, these are equivalent timed markings. Consequently, we obtain (d-1) times the same number of dead markings , ie, (d-1)* (2*MaxParts +1) and, therefore, we deduce:

$$NumberDeadmarkings = (2 * MaxParts + 1) * d \quad (2)$$

All these dead markings are desired terminal states of the protocol. This discussion justifies that the protocol works correctly and the beliefs between the manager and the participants are consistent. Also, it should be noted that if for a given marking two or more transitions are enabled, then the choice of the transition to fire is non-determinist. This means that our system satisfies concurrency and non-determinism which are key characteristics. About the communication channel, we note that at the end of negotiations, the places PART2INIT and INIT2PART are empty, signifying that there is no unprocessed messages in the network, proving, hence, that the property of cleaning the network from late messages is satisfied.

Absence of livelocks and correct termination. Table III shows that the size of the state space increases exponentially with the number of participants and the value of the deadline. This is illustrated by the graph of the Figure 5. The large number of nodes and particularly of dead markings is essentially caused by the increasing value of the deadline. The reason for this is that the timing information makes more markings distinguishable and contributes to the presence of more nodes in the state space leading to several equivalent timed markings. To verify that all the dead markings for all the values of MaxParts specified in Table III form a home space, we have used the CPN ML function HomeSpace (ListDeadMarkings()) which evaluates to true. This confirms that there is no livelocks and the system will always terminate correctly. Table III also shows that, for all values of MaxParts examined, the number of nodes and arcs in the SCC graph always remains the same as that of the state space, this implies that there is no cyclic behavior in the system, which is expected. From Table III, we conclude that there is no live transitions because of the presence of dead markings.

Absence of dead code. A dead code corresponds to a dead transition. According to table III, there is no dead transitions in the system for all values of MaxParts examined, this implies that all the specified actions are executed.

Channel bound. Table III shows that the communication channel is bounded by the MaxParts value examined, this confirms that the manager issues a single message to each of the participants and then MaxParts messages. Similarly, each participant issues, at a given moment, one message to the manager justifying the limit of MaxParts responses.

VI. Conclusion and Perspectives

In this paper, we have proposed a TCPN model of the contract net protocol with temporal aspects. We have used

Table III
STATE SPACE ANALYSIS RESULTS AS A FUNCTION OF THE PARAMETERS MAXPARTS AND DEADLINE (D).

Properties	**MaxParts=1**					**MaxParts=2**					**MaxParts=3**					**MaxParts=4**	
	d=1	d=2	d=3	d=4	d=5	d=1	d=2	d=3	d=4	d=5	d=1	d=2	d=3	d=4	d=5	d=1	d=2
State Space Nodes	28	40	52	64	76	317	605	989	1469	2045	3669	9165	18645	33216	54164	42337	140513
State Space Arcs	38	53	68	83	98	801	1357	2081	2973	4033	14113	30143	55863	93817	146549	221393	619193
Time (seconde)	00	00	00	00	00	00	00	00	01	02	07	33	161	404	831	1298	16119
SCC nodes	28	40	52	64	76	317	605	989	1469	2045	3669	9165	18645	33216	54164	42337	140513
SCC Arcs	38	53	68	83	98	801	1357	2081	2973	4033	14113	30143	55863	93817	146549	221393	619193
Dead Markings	3	4	5	6	7	5	10	15	20	25	7	14	21	28	35	9	18
HomeSpace	true	true	true	true	true	true	true	true	true	true	true	true	true	true	true	true	true
Dead Transition Instances	None	None	None	None	None	None	None	None	None	None	None	None	None	None	None	None	None
Live Transition Instances	None	None	None	None	None	None	None	None	None	None	None	None	None	None	None	None	None
Channel bound	1	1	1	1	1	2	2	2	2	2	3	3	3	3	3	4	4

the simulation and the state space analysis techniques to verify some key properties of the proposed model for different values of both parameters MaxParts and deadline. In addition to have proved that the deadline is always respected, we have also proved the beliefs consistency between the manager and the participants and that the protocol works and ends correctly. The properties namely concurrency, absence of livelocks and absence of dead code were verified too. Furthermore, we have shown how the number of dead markings (terminal states) is related to both MaxParts and deadline parameters. The channel bound is, however, related to only the MaxParts parameter. As perspectives, we would like to use advanced state space reduction methods [15] like equivalence classes [5], [7] to alleviate the impact of the state explosion problem which is most accentuated for timed models. In doing so, we would verify the model for wider values of MaxParts and deadline. We would also like to model real time contract net [6], [16], [18] where, besides interaction aspects, time constraints related to task execution would be considered. These extensions would concern more complex versions of CNP. On the other hand, we would like to model a fault tolerant CNP so that the manager provides a fault tolerant behavior if ever the contractor crashes during task performing.

REFERENCES

[1] J. Billington and A. Gupta, *Effectiveness of Coloured Petri Nets for Modelling and Analysing the Contract Net Protocol*, Proc. Eighth Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, Aarhus, Denmark, 2007, pp. 49-65 (ISSN 0105 8517).

[2] *FIPA*, Foundation for intelligent physical agents 2003. *FIPA Modeling Area: Temporal Constraints*. Retreived May 10, 2012, from *http://www.fipa.org*

[3] *Agent Unified modeling language, AUML*. Retreived May 15, 2012, from *http://www.AUML.org*

[4] S. Aknine, S. Pinson, and M. F. Shakun, *An Extended Multi-Agent Negotiation Protocol, Autonomous Agents and Multi-Agent Systems* 8(1), pp. 5-45 (2004).

[5] W.M.P van der Aalst, *Interval timed coloured petri nets and their analysis*, In Application and Theory of Petri Nets 1993, Proc. 14th International Conference, volume 691, pp. 453-472, Chicago, (USA), 1993. Springer-Verlag, Lecture Notes in Computer Science.

[6] L. Qiaoyun , L. Jiandong, D. Dawei, and K. Lishan, *An extension of contract net protocol with real time constraints*. Wuhan University Journal of Natural Sciences. Wuhan University Journals Press. ISSN:1007-1202 (Print) 1993-4998, Volume 1, Number 2 / juin 1996, pp. 156-162.

[7] B. Berthomieu, *La méthode des Classes d' Etats pour l'Analyse des Réseaux Temporels - Mise en Oeuvre, Extension à la multi-sensibilisation, Modélisation des Systèmes Réactifs*, In Proc. of MSR'2001, pp. 254-263, Toulouse, France, 2001. Hermes Sciences.

[8] R. G. Smith, *The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver*, IEEE Trans. Computers 29(12): 1104-1113 (1980)

[9] *CPN tools homepage*. Retreived May 10, 2012, from *cpn-tools.org/*.

[10] F.D.J. Bowden, *A brief survey and synthesis of the roles of time in Petri nets*, Mathematical and Computer Modelling 31 (2000) pp. 55-68.

[11] K. Jensen and L. M. Kristensen, *Coloured Petri Nets - Modelling and Validation of Concurrent Systems*, Springer, July 2009.

[12] J. Shujuan , Q. Tian and Y. Liang, *A Petri-Net-Based Modeling Framework for Automated Negotiation Protocols in Electronic Commerce*, Lecture Notes in Computer Science, 2009, Volume 4078/2009, pp. 324-336.

[13] F. S. Hsieh, *Automated Negotiation Based on Contract Net and Petri Net*, Lecture Notes in Computer Science, vol. 3590, pp. 148-157, 2005. (SCI).

[14] J. Billington, A. K. Gupta, and G. E. Gallasch, *Modelling and Analysing the Contract Net Protocol - Extension Using Coloured Petri Nets*, Lecture notes in computer science, 2008, NUMB 5048, pp. 169-184, Springer-Verlag.

[15] M. A. Piera and G. Music, *Coloured Petri net scheduling models: Timed state space exploration shortages*, Mathematics and computer in simulation 82, (2011), pp. 428-441, Elsevier.

[16] N. Dragoni, M. Gaspari, and D. Guidi, *An ACL for specifying Fault-Tolerant protocols*, AI*IA 2005: Advances in Artificial Intelligence 9th Congress of the Italian Association for Artificial Intelligence, Italy 2005 pp. 237-248.

[17] L. Changyou and W. Haiyan, *An Improved Contract Net Protocol Based on Concurrent Trading Mechanism*, iscid, vol. 2, pp. 318-321, 2011 Fourth International Symposium on Computational Intelligence and Design, 2011.

[18] W. L. Yeung, *Behavioral modeling and verification of multi-agent systems for manufacturing control*, Expert Systems with Applications, 38(11):13555-13562, 2011.

 ISBN: 978-1-61208-202-8