

Timed-Arc Petri-Nets based Agent Communication for Real-Time Multi-Agent Systems

Awais Qasim¹, Sidra Kanwal², Adnan Khalid³, Syed Asad Raza Kazmi⁴, Jawad Hassan⁵
Department of Computer Science, Government College University, Lahore Pakistan

Abstract—This research focuses on Timed-Arc Petri-nets-based agent communication in real-time multi-agent systems. The Agent Communication Language is a standard language for the agents to communicate. The objective is to combine Timed-Arc Petri-nets and FIPA Performatives in real-time multi-agent systems. FIPA standards provide a richer framework for the interaction of agents and makes it easier to develop a well-defined system. It also ensures the management by precisely specifying the agent's interaction. Though FIPA protocol has already been described with the help of Petri-nets but this specification lacks the timing aspect that is a dire need for real-time multi-agent systems. The main objective of this research is to provide a method of modeling existing FIPA performatives by combining Timed-Arc Petri-nets in real-time multi-agent systems. We have used properties, such as liveness, deadlock and reachability for the formal verification of the proposed modeling technique.

Keywords—Formal verification; FIPA; multi-agent systems; timed-arc petri nets; real-time systems

I. INTRODUCTION

Modeling of agents for the transmission of messages is much needed as agents interact with one another to achieve goals. Cooperation in Multi-Agent Systems (MAS) is mainly achieved through interacting agents. For effective communication these interacting agents require some interaction protocol. The main purpose of the interaction protocol is to provide a set of well defined rules for the communication of agents. The primary function of an agent is to handle the dynamic situations. There is no agent that possesses information of the whole system rather its decision making is dependent upon limited view of the complete system [1]. Real-time agents can be depicted within the deadline. MAS's have been formally specified using petri-nets but no such work has been done for the formal modeling of agent communication protocols in real-time environment. By using the Real-Time Multi-Agent Systems (RTMAS), the interaction of agents is bounded within the proposed context. Protocols become the cause of message flow among these RTMAS's and specify the sequence of messages, number of messages and updation. Foundation for Intelligent Physical Agent (FIPA) performatives provide outline for the existence of agents, their actions and architecture. They also elaborate the authentication of the agents. Agent Communication Language (ACL) is a proposed standard language for agent communications like FIPA-ACL. FIPA has 20 performatives i-e request, inform, accept proposal, etc that describe the interpretation of messages. These messages are actually the actions (communicative actions or CAs). Interaction protocols are the standards that oversee the interaction between agents. It permits the description of expressly sequence of dialogue between agent's communication. Interaction Protocols are utilized to characterize set of messages transmitted between

agents and portray how collaborative agents response on messages. It is normal to make models from basic conventions into complicated protocol. These interaction protocols have been demonstrated in MAS but not presented in RTMAS. FIPA performatives provide an outline for the existence and action of agents. Existing FIPA protocols have been described through Petri-nets but the timing aspect was not specified which was a limitation. The main idea of this research is to formally model the existing FIPA performatives by Timed Arc Petri-Nets (TAPNs) in real-time multi-agent systems. RTMAS is formally determined and checked with the time limitations. In a real time system, there are some actions, which have specified deadlines and depicts how long agents will wait for reply or perform next action by using FIPA performatives.

Agent's communication has been modeled formally in the past but not for real-time environment. The communication between agents is modeled by using Agent Petri-nets (APN) and it is undeniable that integration between protocols and APN greatly facilitates the development of a system which leads to correct interaction between agents through appropriate specification of the exchange of messages. The time aspect has not been handled in [2] before performing any target. The work of [3], [4], [5] leads to formal specification and verification of interactive real-time software agents (RT Agents). Agents work independently and handle the uncertain scenarios. Visually expressive broader structure and modeling approach i.e. TAPN have been used for specification and representation of Stock Market System (SMS). It is based on RTMAS. The Model is verified by Timed Computational Tree Logic (TCTL) fragments AF, AG, EG and EF. In this paper, KQML register conversation and simple negotiation interaction conversation are modeled through CPN. The work done in [6] describes popular ACL like FIPA to formally model the organization of MASs and clarifies the analysis about FIPA ACL semantics. FIPA specification has been used for guidance. ACL specifications have been introduced with the example of online stock brokering to secure the interaction between agents. This new model of ACL has expressiveness and reusability. In [7] nested petri-nets have been used to model multi-agent systems. In [8] the overview of FIPA-ACL and protocols has been given. FIPA-ACL is based on speech act theory as communicative act. With the passage of time, a lot of improvements have been made but still, none of the protocols can be treated as the complete in itself. In [9] an agent-based framework in an unconstrained platform has been described. This research highlights agents and multi-agents system as the state-of-the-art and distributed environments respectively. The work of [10] emphasizes the modeling of the vehicle framework that can go through crossing points with no or less delay. This approach determines how the model

emphasizes the traffic movement (transitions) by reachability graph within time constraints. Time-based constraints for MAS are presented in [11]. It provides clear ways to accomplish the MAS compliance. Vehicle to vehicle communication is presented by using Petri-nets. Vehicle to vehicle communication makes an efficient exchange of messages between cars and also matches the ID of cars for verification. It also works for the modeling and stimulation of vehicles to vehicle communication of discrete event system, Petri-net is used as a powerful tool. In [12] a more far-reaching display and re-enactment approach is introduced that records for the MAS-related conventions as portrayed in the FIPA particular; together with a co-reproduction stage for the examination of MAS. The process of checking a smart workflow management framework and postpone forecast is presented [13], [14]. The work of [15] describes the formal specification approach for the presentation of communicative agents. It explores the internal state of agents and behavior of interactive agents. In [16] model checking techniques are discussed, which are utilized as a part of tool TAPAAL for reachability analysis using inhibitor arcs. Reachability is utilized for the dead state. It finds that there is a state which is not reachable from some other state. The research of [17] presents a liveness based analysis for Timed-Arc Petri-Nets with weights and arcs i.e. inhibitor arcs, transport arcs and age invariants. This research highlights agents and MAS as state of the art and distributed environments, respectively. Face detector and tracker agents interact through contract net protocol. This system tracks the agents which path has been chosen and time spent. Events triggered sample data consensus has been proposed in [18], [19].

Even triggering condition is intermittently examined after constant sampling instant for distributed MAS with directed graph. Consensus of distributed MAS can be transformed into stability of system with time. Then a sufficient condition on the consensus of Multi-Agent system is derived. The management of communication between agents is presented in [20]. This model is illustrated with the help of Petri-nets and result is validated with coordination between agents. In [21] demonstrates that response to the presence of discrete-time and continuous time strategies occur at the same time or getting input or wait with decision until next occasion is activated and creates algorithm for discrete Timed-arc Petri-net games. A work process based that concentrates on the foundational issues of soundness is strongly based on timed-constrained soundness. Through subclass of bounded nets, we can efficiently verify the design [22]. For the effective processing, agents search for more agents by using KQML. Contract net protocol is utilized on the interaction of agents. If an agents bounds with one contract, it is illegible to take new one until the completion of previous [23].

According to our knowledge, formal specification of MAS's interaction in a real-time environment is a novel approach. Although formal modeling of MASs have been done in the past but it is limited to domain functionality of the complete system either at the micro or at macro level. In MAS goals can be defined at any single agent level or at the complete system level but any single goal may contain several agent performatives. Formal modeling of agents interaction allows the designer to verify the system's correctness at the design time. In this research, FIPA performatives are

formally specified in TAPN. TAPN is a framework for visual representation of sequence of events in time. This is used to describe the modeling approach. Arcs are used to represent the time specification i.e inhibitor arc, transport arc and invariants. In the formal specification of RTMAS, agents interact with one another to achieve their goals within time constraints. This element is required for their correct functioning. Protocols define how much long period of time the agents would wait for the concerned interacting agents and also for updation. TAPAAL model checker is used for formal verification of FIPA performatives in RTMASs. TAPAAL is a graphical representation and verification tool of TAPN. It is also used for the verification of different queries specified to ensure the correct functioning of the system's model. The model is verified with the fragments of TCTL whose fragments are AF, AG, EG and EF.

II. PRELIMINARIES

A few terminologies and computational models have been described in this section that will be used in the rest of the discussion for the specification of the problem under analysis.

A. Timed-Arc Petri-Net (TAPN)

TAPN is an established technique for the formal modeling of multi-agents systems in real-time environments. The timing aspect considers unequivocal treatment of real-time. TAPN provides a mechanism to formally verify the properties of interest of the system to ensure its correctness. The TAPN definition has been proposed in [24]. A TAPN is a 6-tuple $(P, T, IA, OA, I, Type)$ where

P: is finite set of places.

T: is a finite set of transitions.

IA: $P * T$ is finite set of input arcs.

OA: $T * P$ is finite set of output arcs.

I: represents the age interval of places.

Type represents the type of arc normal, inhibitor, transport arc.

TAPNs are basically an extension of the standard petri-nets in which age of the tokens is utilized to incorporate timing aspect of the system. The time intervals defined on arcs are used to restrict the progress of the system by only allowing those tokens whose age falls in the interval. Arc defines a place to transition and transition to place. Transition firing and enabling depend on the age of token. Transition enabling and firing is not possible when the time interval is expired that is mentioned on a certain arc. There are three types of arcs such as transport arc, inhibitor arc and normal arc. Transport arc produces the same time which is consumed on input arc. Inhibitor arc restricts the certain age of token on the place. Normal arc produces the age zero token although consumed of any age token. TAPN is used for the verification of reachability, boundedness and liveness properties.

B. TAPAAL

TAPAAL is the graphical representation and powerful tool for modeling of Timed-Arc Petri-nets. TAPAAL is used for

simulation and verification of TAPN. It is graphical, modeling editor for Timed-Arc Petri-net. It provides its own engine for verification [25]. For verification different properties are used such as Reachability, boundedness, and liveness. These properties are defined under the Timed Computational Tree Logic (TCTL). TCTL is the actual logic which is utilized for the determination of properties about structure. The fragments of TCTL are AF, AG, FG, and EF.

C. FIPA Performatives

FIPA developed its standards in 1995 for agent’s communication. Speech act theory provides a base for FIPA ACL. Modeling of agents is useful for the transmission of messages. Agents interact with one another for the achievement of some goals. Agent communication language is a proposed standard language for agent communications like FIPA-ACL. In [26] FIPA provides a specification for agent Communication. FIPA-ACL suggests the parameters for effective communication among the agents within the scenario. All performatives of FIPA contain message structure i-e sender, receiver and content of the message. The message expresses the meaning of agents. It consists of the content and action of the communication. If an agent communicates with another agent, a suitable performative is used. If the agent does not understand some message and is unable to the process the sent message, it can reply with the “not understand” performative. In [26] FIPA has proposed 20 performatives that cover all the maximum range of aspect of agents expected communication. FIPA-ACL is now a basic standard for determining the encoding and exchange of messages among agents. In [27] a set of performatives is given for ACLs that specifies how these communicating actions should be executed in a concurrent and reactive way with respect to a given logical semantics. It has assumptions about how the recipient should react to the message.

III. PROPOSED TAPN BASED FIPA PERFORMATIVES

In our designed model of performatives, workflow shows within time highlights, enables transitions, sends time interval and manages task completion within deadlines. Our model is the combination of FIPA performatives and Timed-Arc Petri-Nets in which token sends the message with time constraints. We use places as agents which generate tokens. Tokens convey an age and the age is exactly equal to the time interval that is defined on arc. Arc defines the area and limit within which transition takes place. Transition firing and enabling depend on the age of the token. In our proposed solution, each performative must be sent within seven seconds. When we map these performatives to a real-time multi-agent system, each performative takes predefined time but the starting time and ending time of transition may change according to the RTMAS. To save space only two out of twenty performatives have been specified in this paper.

A. Request

Using request performative, one agent requests another agent to take some action. For the simulation of this act, we have used two places and six transitions. The agent1 place has an initial marking 0.0 token that represents the age of the token. Agent1 requests to Agent2 as transition firing $\langle \text{Request}_{A1_A2} \rangle$ from Agent1 to $\langle \text{Request}_{A1_A2} \rangle$

and $\langle \text{Request}_{A1_A2} \rangle$ to A2. Time interval on arc restricts the token firing time that must be fired within [StRequest, EnRequest] including delay. Agent2 accepts the request by passing the message of Agree as transition firing $\langle \text{Agree}_{A2_A1} \rangle$ and can also refuse the request as transition firing $\langle \text{Refuse}_{A2_A1} \rangle$. A1 receives the response of Agent2. In case Agent2 agrees to the request of Agent1, token is fired on Agent1 place with the age 0 for the response of agree and another token is fired on Agent2 place to continue the process with same age that has been consumed at the fired transition. For this purpose, we have used transport arc.

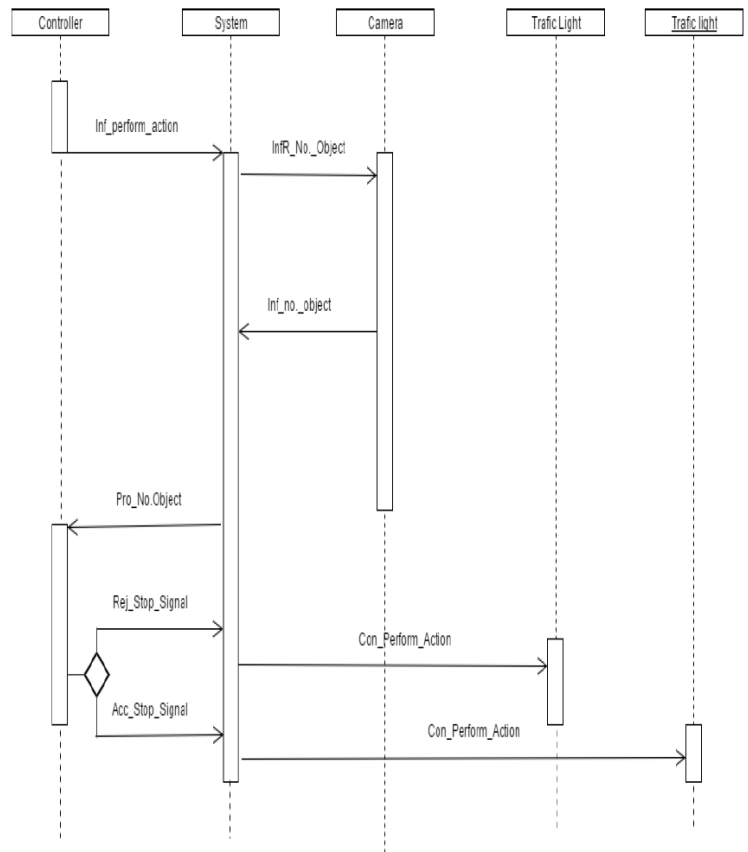


Fig. 3. Proposed Sequence of Traffic Light System.

In response of Agree, Agent2 replies in the form of $\langle \text{InformDone}_{A2_A1} \rangle$ or within [StInfDone , EnInfDone] if the task is executed within the deadline. Agent2 can reply in the form of $\langle \text{InfoRef}_{A2_A1} \rangle$ within [StInfoRef , EnInfoRef] in case of detailed reply. If the agent fails to fulfill the requirement within the specified deadline, reply $\langle \text{Failure}_{A2_A1} \rangle$ comes within [StFailure , EnFailure]. Finally, the task is executed within time limit with the options of EnFailure, EnInfoRef or EnInfDone. Simulation history of the two agents is represented in Fig. 1 and the results of temporal constraints are shown in Table I.

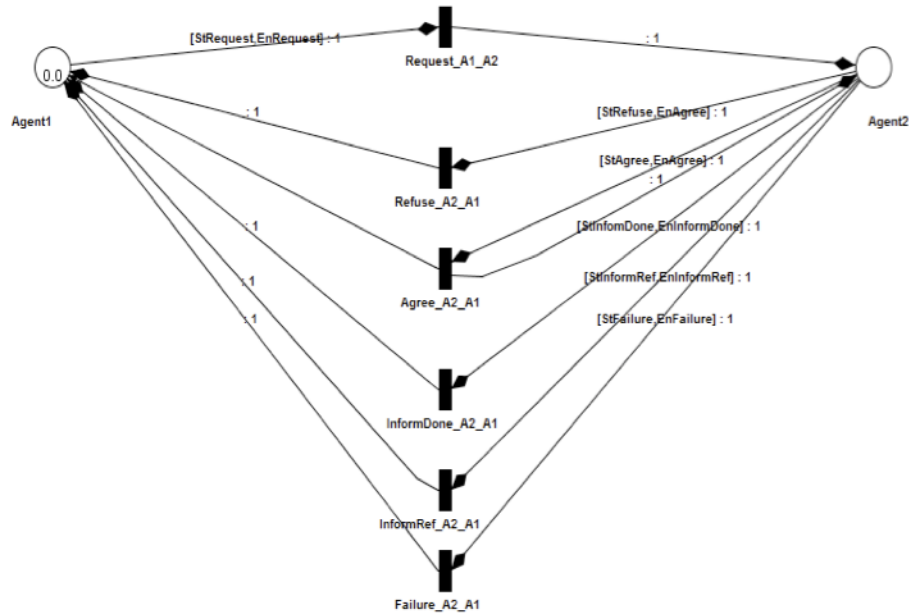


Fig. 1. TAPN based Request Performative for two agents.

TABLE I. TRANSITION AND TIMING DURATION OF TWO AGENTS IN REQUEST PERFORMATIVE.

Place	Transition	Total Time Duration	Start Time	End Time
Agent1	Request_A1_A2	7	0	7
Agent2	Refuse_A2_A1	7	8	15
Agent2	Agree_A2_A1	7	8	15
Agent2	InformDone_A2_A1	7	16	23
Agent2	InfoRef_A2_A1	7	16	23
Agent2	Failure_A2_A1	7	16	23

TABLE II. TRANSITION AND TIMING DURATION OF ONE TO MANY AGENTS INTERACTION IN CFP PERFORMATIVE.

Place	Transition	Total Time Duration	Start Time	End Time
Agent1	CFP_A1_A2A3	7	0	7
Agent2	Proposal_A2_A1	7	8	15
Agent2	Refuse_A2_A1	7	8	15
Agent2	NotUnderstand_A2_A1	7	8	15
Agent3	Proposal_A3_A1	7	8	15
Agent3	Refuse_A3_A1	7	8	15
Agent3	NotUnderstand_A3_A1	7	8	15
Agent1	RejectProposal_A1_A2	7	16	23
Agent1	RejectProposal_A1_A2	7	16	23
Agent1	AcceptProposal_A1_A2	7	24	31
Agent1	AcceptProposal_A1_A3	7	24	31
Agent2	InfoDone_A2_A1	7	32	39
Agent2	InfoRef_A2_A1	7	32	39
Agent2	Failure_A2_A1	7	32	39
Agent3	InfoDone_A3_A1	7	32	39
Agent3	InfoRef_A3_A1	7	32	39
Agent3	Failure_A3_A1	7	32	39

B. Call for Proposal

Call for proposal (CFP) is used to start communication between agents. FIPA CFP performative augments approval and denial of preceding form of communicating agents. In

CFP performative, one agent acts as a manager, that requires a certain task to be accomplished effectively within a specified time. In CFP performative, the manager sends the call for proposal to the contractors. The contractors send the reply in one of the three forms that are refused, not understanding and proposal within the time limit. After getting the response from the contractors, the manager approves one of the proposals and sends rejection (reject proposal) to the remaining agents. The selected contractor then apprises the manager of task completion. In CFP performative, we have used three places and fifteen transitions. Agent1 place has 0.0 token that represents the age of the token. Agent1 CFP to Agent2 and Agent3 is represented as transition firing $\langle \text{CFP_A1_A2A3} \rangle$. Time interval on arc restricts the token firing time that must be fired within $[\text{StCFP, EnCFP}]$ including delay. Agent2 accepts the message by passing the proposal as transition firing $\langle \text{Proposal_A2_A1} \rangle$ and refuses transition firing $\langle \text{Refuse_A2_A1} \rangle$. Agent1 can accept the proposal as transition firing $\langle \text{AcceptProposal_A1_A2} \rangle$ within $[\text{StAccept, EnAccept}]$. For accept Proposal, we have used transport arc which means that the age of the token remains same as the age of token at the firing time. Agent1 can reject the proposal as transition firing $\langle \text{RejectProposal_A1_A2} \rangle$ with time constraints $[\text{StReject, EnReject}]$. In case of reject proposal, the token is transferred to Agent2 or Agent3. The agent replies further within time constraint if the proposal is accepted by Agent1. After Acceptance of proposal, Agent2 replies in the form of $\langle \text{InformDone_A2_A1} \rangle$ or $\langle \text{InformDone_A3_A1} \rangle$ within $[\text{StInfDone, EnInfDone}]$ if the task is completed within deadline. Agent2 or Agent3 can reply in the form of $\langle \text{InfoRef_A2_A1} \rangle$ $\langle \text{InfoRef_A3_A1} \rangle$ within $[\text{StInfoRef, EnInfoRef}]$ if detailed reply. If agent fails to fulfil the proposal requirements within fixed deadline then it replies as $\langle \text{Failure_A2_A1} \rangle$ or $\langle \text{Failure_A3_A1} \rangle$ within $[\text{StFailure, EnFailure}]$.

TABLE III. PROPERTIES FOR REQUEST PERFORMATIVE AND TCTL FRAGMENTS FOR TAPAAL.

Query	Formula	Result	Count	Transition	Place	Maximum Token
Are both agents reachable?	EF (Request1to1.Agent2 = 1)	Satisfied	8	CFP.InformDone_A1_A2A3	CFP.Agent1	2
Are all agents accessible?	EF (Request.Agent1 = 1 and Request.Agent2 = 1 and Request.Agent3 = 1)	Satisfied	1	CFP.Proposal_A3_A1	CFP.Agent3	1
			1	CFP.Refuse_A3_A1	CFP.Agent2	1
			1	CFP.Proposal_A2_A1		
			1	CFP.NotUnderstand_A3_A1		
			0	CFP.AcceptProposal_A3_A1		
			0	CFP.RejectProposal_A3_A1		
			0	CFP.InformDone_A3_A1		
			0	CFP.InformRef_A3_A1		
			0	CFP.Failure_A2_A1		
			0	CFP.AcceptProposal_A1_A2		
			0	CFP.RejectProposal_A1_A2		
			0	CFP.InformDone_A2_A1		
			0	CFP.InformRef_A2_A1		
			0	CFP.Failure_A2_A1		
			0	CFP.InformRef_A3_A1		
			0	CFP.NotUnderstand_A3_A1		

TABLE IV. K-BOUNDEDNESS OF REQUEST PERFORMATIVE "ARE BOTH AGENTS REACHABLE?".

Count	Transition
1	Request1to1.Request_A1_A2
0	Request1to1.Refuse_A2_A1
1	Request1to1.Agree_A2_A1
1	Request1to1.InformDone_A2_A1
1	Request1to1.InformRef_A2_A1
1	Request1to1.Failure_A2_A

TABLE V. K-BOUNDEDNESS OF REQUEST PERFORMATIVE "ARE ALL AGENTS ACCESSIBLE?".

Count	Transition
4	Request.Request_A1_A3
4	Request.Request_A1_A2
3	Request.Refuse_A3_A1
3	Request.Agree_A3_A1
3	Request.InformDone_A3_A1
3	Request.InformRef_A3_A1
3	Request.Failure_A3_A1
1	Request.Refuse_A2_A1
1	Request.InformDone_A2_A1
1	Request.InformRef_A2_A1
1	Request.Failure_A2_A1
1	Request.Agree_A2_A1

TABLE VI. PROPERTIES FOR CFP PERFORMATIVE AND TCTL FORMULA.

Query	Formula	Result
Is agent2 globally reachable?	EG (CFP.Agent2 = 1)	Not Satisfied
Are both agent2 and agent3 reachable?	EF ((CFP.Agent1 = 2 and CFP.Agent2 = 0) or (((!(CFP.Agent1=2) and CFP.Agent4=1) and CFP.Agent=1 and !(CFP.Agent1 = 2)) and !(CFP.Agent3 = 2))	Satisfied
All these states reachable?	EF ((CFP.Agent1 = 2 and CFP.Agent2 = 0) or (((!(CFP.Agent1=2) and CFP.Agent4=1) and CFP.Agent=1 and !(CFP.Agent1 = 2)) and !(CFP.Agent3 = 2)))	Satisfied

TABLE VII. K-BOUNDEDNESS OF CFP PERFORMATIVE "ALL THESE STATES REACHABLE?"

Count	Transition	Place	Maximum Token
1	CFP.CFP_A1_A2A3	CFP.Agent1	1
0	CFP.Proposal_A3_A1	CFP.Agent3	1
0	CFP.Refuse_A3_A1	CFP.Agent2	1
0	CFP.AcceptProposal_A1_A3		
0	CFP.RejectProposal_A1_A3		
0	CFP.InformDone_A3_A1		
0	CFP.InformRef_A3_A1		
0	CFP.Failure_A3_A1		
0	CFP.Proposal_A2_A1		
0	CFP.Refuse_A2_A1		
0	CFP.AcceptProposal_A1_A2		
0	CFP.RejectProposal_A1_A2		
0	CFP.InformDone_A2_A1		
0	CFP.Failure_A2_A1		
0	CFP.InformRef_A2_A1		
0	CFP.Failure_A2_A1		
0	CFP.InformRef_A2_A1		
0	CFP.Accept		
0	CFP.Accept		
0	CFP.NotUnderstand_A3_A1		
0	CFP.NotUnderstand_A2_A1		

TABLE VIII. K-BOUNDEDNESS OF CFP PERFORMATIVE "ARE BOTH AGENT2 AND AGENT3 REACHABLE?"

Count	Transition	Place	Maximum Token
1	CFP.CFP_A1_A2A3	CFP.Agent1	1
0	CFP.Proposal_A3_A1	CFP.Agent3	1
0	CFP.Refuse_A3_A1	CFP.Agent2	1
0	CFP.AcceptProposal_A1_A3		
0	CFP.RejectProposal_A1_A3		
0	CFP.InformDone_A3_A1		
0	CFP.InformRef_A3_A1		
0	CFP.Failure_A3_A1		
0	CFP.Proposal_A2_A1		
0	CFP.Refuse_A2_A1		
0	CFP.AcceptProposal_A1_A2		
0	CFP.RejectProposal_A1_A2		
0	CFP.InformDone_A2_A1		
0	CFP.Failure_A2_A1		
0	CFP.InformRef_A2_A1		
0	CFP.Accept		
0	CFP.Accept		
0	CFP.NotUnderstand_A3_A1		
0	CFP.NotUnderstand_A2_A1		

TABLE IX. PROPERTIES FOR TLS AND TCTL FRAGMENTS FOR TAPAAL.

Query	Formula	Result
Are all systems reachable?	EF (TLS.System1 = 1 and TLS.System2 = 1 and TLS.System3 = 1 and TLS.System4 = 1)	Satisfied
Are all cameras work properly?	EF (TLS.Camera1 = 1 and TLS.Camera2 = 1 and TLS.Camera4 = 1 and TLS.Camera3 = 1)	Satisfied

TABLE X. EXECUTION COUNT FOR ALL THE SIGNALS IN TAPAAL.

Count	Transition
31	TLS.Acc_Op_Sig1
31	TLS.Acc_Op_Sig4
31	TLS.Acc_Op_Sig3
31	TLS.Acc_Op_Sig2
26	TLS.Rej_St_Sig2
26	TLS.Rej_St_Sig1
26	TLS.Rej_St_Sig
26	TLS.Rej_St_Sig3
21	TLS.Inf_Per_Act1
21	TLS.Inf_Per_Act3
21	TLS.Inf_Per_Act4
21	TLS.Inf_Per_Act2
14	TLS.Con_Per_ActG1
13	TLS.Con_Per_ActR1
12	TLS.Pro_N_Obj1
12	TLS.InfR_N_Obj1
8	TLS.Pro_N_Sig3
8	TLS.Con_Per_ActG3
8	TLS.InfR_N_Obj3
8	TLS.Con_Per_ActR3
7	TLS.Pro_N_Obj4
7	TLS.Con_Per_ActR4
7	TLS.Con_Per_ActG4
7	TLS.InfR_N_Obj4
6	TLS.Pro_N_Obj2
6	TLS.InfR_N_Obj2
6	TLS.Con_Per_ActG2
6	TLS.Con_Per_ActR2
2	TLS.Go1
2	TLS.Stop1
2	TLS.Inf_N_Obj1
1	TLS.Inf_N_Obj2
1	TLS.Go2
1	TLS.Stop2
1	TLS.Stop4
1	TLS.Go4
1	TLS.Inf_N_Obj4
1	TLS.Go3
1	TLS.Inf_N_Obj3
1	TLS.Stop3

TABLE XI. EXECUTION COUNT FOR TRAFFIC LIGHT SYSTEMS IN TAPAAL.

Count	Transition
1122	TLS.Con_Per_ActG1
967	TLS.Con_Per_ActG3
891	TLS.Con_Per_ActG4
846	TLS.Con_Per_ActG2
725	TLS.Acc_Op_Sig1
725	TLS.Acc_Op_Sig4
725	TLS.Acc_Op_Sig3
725	TLS.Acc_Op_Sig2
673	TLS.Go1
528	TLS.Go3
465	TLS.Go4
431	TLS.Go2
421	TLS.Con_Per_ActR1
388	TLS.Con_Per_ActR3
372	TLS.Con_Per_ActR4
367	TLS.Stop1
364	TLS.Con_Per_ActR
299	TLS.Stop3
272	TLS.Stop4
256	TLS.Stop2
165	TLS.Rej_St_Sig2
165	TLS.Rej_St_Sig1
165	TLS.Rej_St_Sig
165	TLS.Rej_ST_Sig3
159	TLS.Pro_N_Obj1
159	TLS.InfR_N_Obj1
154	TLS.Pro_N_Sig3
154	TLS.InfR_N_Obj3
151	TLS.Pro_N_Obj4
151	TLS.InfR_N_Obj4
150	TLS.Pro_N_Obj2
150	TLS.InfR_N_Obj2
135	TLS.Inf_N_Obj1
122	TLS.Inf_N_Obj3
118	TLS.Inf_N_Obj4
115	TLS.Inf_N_Obj2
35	TLS.Inf_Per_Act1
35	TLS.Inf_Per_Act3
35	TLS.Inf_Per_Act4
35	TLS.Inf_Per_Act2

B. Call for Proposal

The interaction between two or many agents for the CFP protocol has been verified here. In Table VI, verification results of the properties of interest of the system specified in TCTL fragments are shown. Table VII and Table VIII shows the result for K-boundedness of the system.

V. APPLICATION OF THE PROPOSED APPROACH

In this section, we have formally modeled FIPA performatives in RTMAS using TAPN to demonstrate the application of the proposed approach to real-time applications. We have formally specified and verified the Traffic Light System (TLS) based on RTMAS using TAPN. In TLS, the traffic actions are governed by TAPN in to control the lights intersections. The proposed TAPN based application is an eight line traffic system. The signals are used in a cyclic arrangement of on and off. The presented technique can be used to manage the rush of traffic, balance traffic flow, and traffic safety. The application has been explained through a sequence transactions and TAPN.

A. Traffic Light System

The proposed application is based on eight lines. A signal gets on and off in a cyclic pattern. Cameras are fixed to detect vehicles and inform to the system about the number of vehicles. We use the following agents which are controller,

System, Camera, <YG_Traffic_Light> <G_Traffic_Light>, <YR_Traffic_Light>, <R_Traffic_Light>. Complete working of the system is shown in Fig. 3.

A large number of vehicles are managed by traffic lights so their efficient and smooth working is very important in maintaining traffic flow and to prevent accidents. In our proposed application, we have attempted to map FIPA performatives for traffic light system with timing aspect. We have used an eight lines traffic system comprising the following agents; controller, system, camera, <YR_T_Light>, <YG_T_Light>. TAPN based specification of the system is shown in Fig. 4. The controller agent manages the system and gives instructions to other systems to perform the task. System manages the cameras that are fixed above the lines with the traffic signal. The controller gives instructions to the system to perform action. For this scenario, we have used ‘inform’ performative that works in the controller to <Inf_Per_Act> and <Inf_Per_Act> to system. <Inf_Per_Act> is the same for all systems that is <Inf_Per_Act1>, <Inf_Per_Act2>, <Inf_Per_Act3> <Inf_Per_Act4>. After getting instructions from the controller, the system gives instructions to its own camera that is fixed along with traffic lights. For sending the instruction from the system to the camera <InfR_N_obj1>, <InfR_N_obj2>, <InfR_N_obj3> and <InfR_N_obj4> are used for camera1, camera2, camera3, and camera4, respectively. Sensor

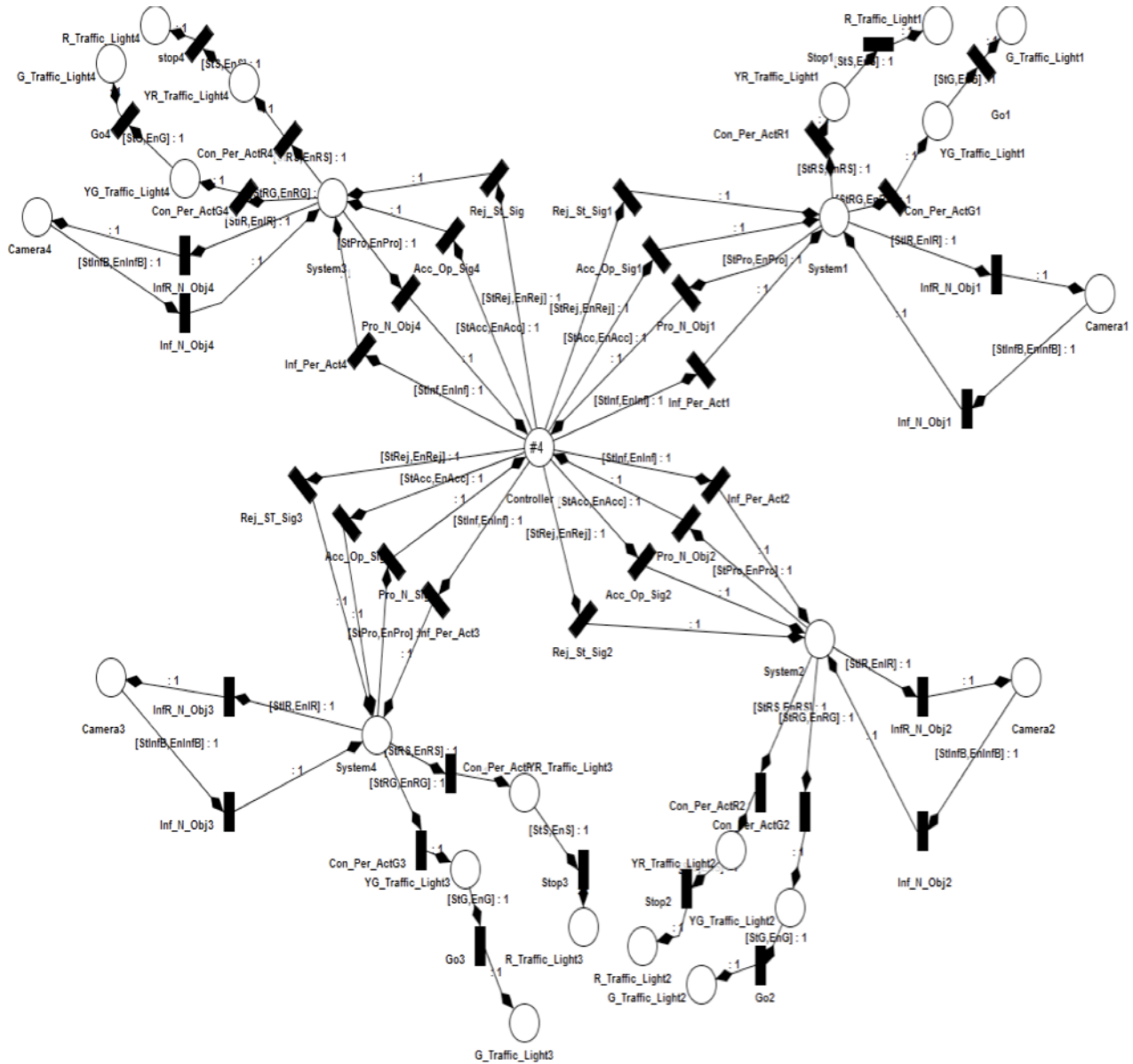


Fig. 4. TAPN based Traffic Light System.

senses the presence of vehicles and gives information to the system using $\langle \text{Inf_N_Obj} \rangle$ transition that is completed in $[\text{StInfO, EnInfO}]$. Now each system tells the controller about the number of vehicles using the proposal interaction protocol. Transition varies from system to $\langle \text{Pro_N_Obj} \rangle$ and $\langle \text{Pro_N_Obj} \rangle$ to the controller. The controller checks each system proposal and finds the maximum number of vehicles. It rejects the proposal of system that does not have the maximum number of vehicles. Next, the system sends message to yellow traffic light agent that is $\langle \text{YR_Traffic_Light} \rangle$ to get ready and the $\langle \text{YR_Traffic_Light} \rangle$ is turned to $\langle \text{R_Traffic_light} \rangle$ to perform the action of stop. The controller sends the accep-

tance message to the system that consists of the maximum number of vehicles and the System sends a message to $\langle \text{YG_Traffic_Light} \rangle$ to get ready. Then $\langle \text{YG_Traffic_Light} \rangle$ is shifted to $\langle \text{G_Traffic_light} \rangle$ using go transition within $[\text{StG, EnG}]$. In Table IX, verification results of the properties of interest of the TLS specified in TCTL fragments are shown. Table X and Table XI shows the transition count of different arcs during simulation. The higher number of counts represents more traffic at that signal.

TAPN has been used for the verification of boundedness, reachability and liveness properties. Formal verification gives

the correctness of the system. Boundedness ensures the maximum and the minimum number of tokens that each place holds. Reachability determines the sequence from the first node of marking to the second node. In our proposed application, all the states are reachable. Liveness determines that the application is executable. All places can contain any number of tokens throughout the life cycle. Our system is deadlock free because there is no such place where deadlock can occur. The dead-lock free system implies that the system is live and all places are working properly. All these properties ensure the correctness of our system. We have also achieved the inter operability and a well-defined process by using FIPA-ACL standards in our application.

VI. CONCLUSION

In this research, we have formally modeled FIPA performatives in RTMAS using TAPN for agent communication. Communication of agents is a significant characteristic of RTMAS and is useful for message interaction in real-time multi-agent systems. Previous work has focused on formal modeling of domain functionality of multi-agent systems and not on the agent's interaction level. The formal specification and verification of these multi-agent systems in which the agents interact with one another to accomplish their objectives with time constraints ensures their reliability. TAPAAL has been used for the verification of the properties of interest of the system specified through AF, AG, EG and EF fragments of TCTL. The research provides future directions for formal modeling of standardized agent's interaction with timing constraints. The approach ensures that the system is deadlock free and live. For the future, we will work on FIPA performatives with Timed Colored Petri-nets in RTMAS.

REFERENCES

- [1] N. R. Jennings, K. Sycara, and M. Wooldridge, "A roadmap of agent research and development," *Autonomous agents and multi-agent systems*, vol. 1, no. 1, pp. 7–38, 1998.
- [2] B. Marzougui and K. Barkaoui, "Interaction protocols in multi-agent systems based on agent petri nets model," *Int J Adv Comput Sci Appl*, vol. 4, no. 7, 2013.
- [3] A. Qasim, S. A. R. Kazmi, and I. Fakhir, "Formal specification and verification of real-time multi-agent systems using timed-arc petri nets," *Adv. Elect. Comput. Eng.*, vol. 15, no. 3, pp. 73–78, 2015.
- [4] A. Qasim and S. A. R. Kazmi, "Mape-k interfaces for formal modeling of real-time self-adaptive multi-agent systems," *IEEE Access*, vol. 4, pp. 4946–4958, 2016.
- [5] I. Obaid, S. A. R. Kazmi, and A. Qasim, "Modeling and verification of payment system in e-banking," *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS*, vol. 8, no. 8, pp. 195–201, 2017.
- [6] J. Pitt and A. Mamdani, "Some remarks on the semantics of fipa's agent communication language," *Autonomous Agents and Multi-Agent Systems*, vol. 2, no. 4, pp. 333–356, 1999.
- [7] L. Chang, X. He, and S. M. Shatz, "A methodology for modeling multi-agent systems using nested petri nets," *International Journal of Software Engineering and Knowledge Engineering*, vol. 22, no. 07, pp. 891–925, 2012.
- [8] D. Juneja, A. Jagga, and A. Singh, "A review of fipa standardized agent communication language and interaction protocols," *Journal of Network Communications and Emerging Technologies*, vol. 5, no. 2, pp. 179–191, 2015.
- [9] C. Zaghetto, L. H. M. Aguiar, A. Zaghetto, C. G. Ralha, and F. de Barros Vidal, "Agent-based framework to individual tracking in unconstrained environments," *Expert Systems with Applications*, vol. 87, pp. 118–128, 2017.
- [10] Y.-S. Huang, Y.-S. Weng, and M. Zhou, "Design of traffic safety control systems for emergency vehicle preemption using timed petri nets," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2113–2120, 2015.
- [11] D. Calvaresi, M. Marinoni, A. Sturm, M. Schumacher, and G. Buttazzo, "The challenge of real-time multi-agent systems for enabling iot and cps," in *Proceedings of the international conference on web intelligence*. ACM, 2017, pp. 356–364.
- [12] C. Shum, W. H. Lau, T. Wong, T. Mao, S. Chung, C. Tse, K. F. Tsang, and L. L. Lai, "Modeling and simulating communications of multiagent systems in smart grid," in *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 2016, pp. 405–410.
- [13] A. Pla, P. Gay, J. Meléndez, and B. López, "Petri net-based process monitoring: a workflow management system for process modelling and monitoring," *Journal of Intelligent Manufacturing*, vol. 25, no. 3, pp. 539–554, 2014.
- [14] S. Khosravifar, "Modeling multi agent communication activities with petri nets," *International Journal of Information and Education Technology*, vol. 3, no. 3, p. 310, 2013.
- [15] B. Marzougui, K. Hassine, and K. Barkaoui, "A new formalism for modeling a multi agent systems: Agent petri nets," *Journal of Software Engineering and Applications*, vol. 3, no. 12, p. 1118, 2010.
- [16] J. F. Jensen, T. Nielsen, L. K. Oestergaard, and J. Srba, "Tapaal and reachability analysis of p/t nets," in *Transactions on Petri Nets and Other Models of Concurrency XI*. Springer, 2016, pp. 307–318.
- [17] M. Andersen, H. G. Larsen, J. Srba, M. G. Sørensen, and J. H. Taankvist, "Verification of liveness properties on closed timed-arc petri nets," in *International Doctoral Workshop on Mathematical and Engineering Methods in Computer Science*. Springer, 2012, pp. 69–81.
- [18] G. Guo, L. Ding, and Q.-L. Han, "A distributed event-triggered transmission strategy for sampled-data consensus of multi-agent systems," *Automatica*, vol. 50, no. 5, pp. 1489–1496, 2014.
- [19] G. S. Seyboth, D. V. Dimarogonas, and K. H. Johansson, "Event-based broadcasting for multi-agent average consensus," *Automatica*, vol. 49, no. 1, pp. 245–252, 2013.
- [20] W. Louhichi, B. Marzougui, and K. Hassine, "Formal model for coordination in multi-agents system based petri net agent," in *2017 International Conference on Smart, Monitored and Controlled Cities (SM2C)*. IEEE, 2017, pp. 134–137.
- [21] P. G. Jensen, K. G. Larsen, and J. Srba, "Discrete and continuous strategies for timed-arc petri net games," *International Journal on Software Tools for Technology Transfer*, vol. 20, no. 5, pp. 529–546, 2018.
- [22] J. A. Mateo, J. Srba, and M. G. Sørensen, "Soundness of timed-arc workflow nets," in *International Conference on Applications and Theory of Petri Nets and Concurrency*. Springer, 2014, pp. 51–70.
- [23] A. Kaur and S. Jain, "Kqml-from scenario to technology," *International Journal of Advanced Studies in Computers, Science and Engineering*, vol. 7, no. 3, pp. 30–34, 2018.
- [24] W. M. Zuberek, "Timed petri nets definitions, properties, and applications," *Microelectronics Reliability*, vol. 31, no. 4, pp. 627–644, 1991.
- [25] J. Byg, K. Y. Jørgensen, and J. Srba, "Tapaal: Editor, simulator and verifier of timed-arc petri nets," in *International Symposium on Automated Technology for Verification and Analysis*. Springer, 2009, pp. 84–89.
- [26] P. D. O'Brien and R. C. Nicol, "Fipa—towards a standard for software agents," *BT Technology Journal*, vol. 16, no. 3, pp. 51–59, 1998.
- [27] N. Dragoni and M. Gaspari, "Performative patterns for designing verifiable acls," in *International Workshop on Cooperative Information Agents*. Springer, 2006, pp. 375–387.