International Journal of Control and Automation
Vol.7, No.2 (2014), pp.21-32
http://dx.doi.org/10.14257/ijca.2014.7.2.03

Modeling and Analysis of Traffic Guidance Systems Based on Multi-Agent

Haitao Zhang and Yanyan Li

Information Engineering College, Henan University of Science and Technology, Luoyang 471023, China

zhang_haitao@163.com

Abstract

Aimed at the deficiency of existing modeling method of traffic guidance system based on Multi-Agent, the advantage of Unified Modeling Language (UML) and Hierarchical Colored Petri Nets (HCPN) is combined to model the system. The modeling method using UML and HCPN is first put forward, and more UML diagrams are constructed so as to model the framework of traffic guidance systems. Moreover, the mapping rules from UML model to HCPN is set up, the formal model of traffic guidance system is gotten, and design error may be found by formal verification and validation of HCPN model. So UML model can be improved and correct UML model can be gotten.

Keywords: *Multi-Agent, Petri Nets, Traffic Guidance Systems, UML*

1. Introduction

With the rapid development of city transportation, it becomes more necessary to develop traffic guidance systems. Because of the complexity of city traffic, for a given destination the driver must identify all feasible routes, and select the most appropriate route in accordance with complicated traffic information.

Traffic guidance systems have been developed in many developed countries, such as United States, Germany and Japan. These systems play a very significant role in strengthening traffic management and control function [1]. In china, the traffic guidance systems have been lately paid more and more attention by the government, and set up the traffic guidance systems in some large city such as Beijing, shanghai and Shenzhen [2]. However, the existing traffic guidance systems are too complicated and costly, and are not fit for the need of many Chinese cities. So it is necessary to develop the traffic guidance systems on basic of the existing equipments and technology.

In recent years with the popularity of embedded processors and network technology, vehicles, intersection controllers, and control centers have made up a networked system. With higher performance requirements of traffic guidance control, the complexity of control algorithm and the cost of control device will increase rapidly. For example, in order to make traffic guidance control more intelligent, we must design the intelligent traffic intersection controllers and vehicular controller with higher cost. In addition, most embedded devices only have limited ability of computation and memory, and it is difficult for these devices to implement complex control algorithm, so traditional control theory including classical and modern control theory has great deficiencies. So some researchers introduce agent technology into the network systems, and put forward agent-based control method [3-4]. The control method is fit for the running and management of traffic systems and vehicle systems.

ISSN: 2005-4297 IJCA

The first step is to design formal model of agent-based control system on so as to analyze and verify whether the design method is correct, then we can get software agent from the model after correct model is gotten. Petri Nets has been used to model multi-agent system [5-6]. Petri Nets has strong dynamic analysis capabilities of the concurrency, asynchronous and uncertainty of the system. It is easily not only extended to satisfy modeling requirements of complicated traffic systems, but also has perfect math theory and simulation tool. So the hierarchical colored Petri Nets is applied to satisfy the requirements of agent-based traffic guidance systems.

However, Petri Nets is mainly used in the analysis stage of agents, and cares for the conversation and interaction of agents. But it isn't fit for the whole design and implementation stage of systems. In order to make up for the deficiencies of Petri Nets, UML is introduced into the whole modeling of traffic guidance systems so as to get perfect system model [7-8]. UML is very good at describing the static structure of systems, but don't fit to describe the dynamic behavior, and UML model is lack of strict and effective verification and analysis method, so UML and HCPN are combined to model traffic guidance systems.

The rest of the paper is organized as follows. In Section 2, we first present the modeling methodology of agent-based systems for traffic guidance systems. Section 3 gives the construction method of traffic guidance systems based on multi-agent technology, and corresponding simulation result. Finally, a brief summary are discussed in Section 4.

2. Modeling Method

2.1. Hierarchical colored Petri Nets

Basic Petri Nets is difficult to model and analyze the complicated systems, so Hierarchical Colored Petri Nets (HCPN) is used to model agent-based control systems. In the following, we first present the definition of HCPN.

Definition 1: Hierarchical Colored Petri nets is a 10-tuple HCPN=(S, SN, SA, PN, PT, PA, FS, FT, PP) where S is a finite set of pages. $\forall s \in S$, s is a non-hierarchical CPN. If $\forall s_1, s_2 \in S$ and $s_1 \neq s_2$, then they don't include common elements;

$SN \subset T$ is a set of substitution nodes;

SN is a page assignment function. It maps each substitution node to a page.

$PN \subseteq P$ is a set of port nodes.

PT is a function of port type. PT : PN $\rightarrow$ {in, out, i/o, general}.

PA is a function of port allocation that makes Socket nodes associate with port nodes.

$FS \subseteq PS$ is a limited set of fusion sets. $\forall$ fs $\in$ FS : p1p2 $\in$ fs : [C (p1) =C (p2) $\wedge$ I (p1) =I (p2)]

FT is a function of fusion type. FT : FS $\rightarrow$ {global, page, instance} ;

PP $\in$ SMS is a set of prime page.

2.2. UML

UML is a graphical modeling language, and uses diagrams to describe static structures as well as dynamic behavior systems. Each diagram is composed of a set of figures that includes the important information on one aspect of the system. UML describes the static structure of the system by class diagrams and object diagrams, and describes the dynamic behaviors by state diagrams, collaboration diagrams, sequence diagrams and activity diagrams.

Of course, when we model a system, we don't need design all diagrams, but select the necessary diagrams according to real requirements.

Once UML model is designed by the software “rational rose”, source code framework is easy to be gotten from the software. The following UML diagrams are all designed in the software.

2.3. Modeling Framework

In the following, we first present modeling framework combining the merits of UML and HCPN.

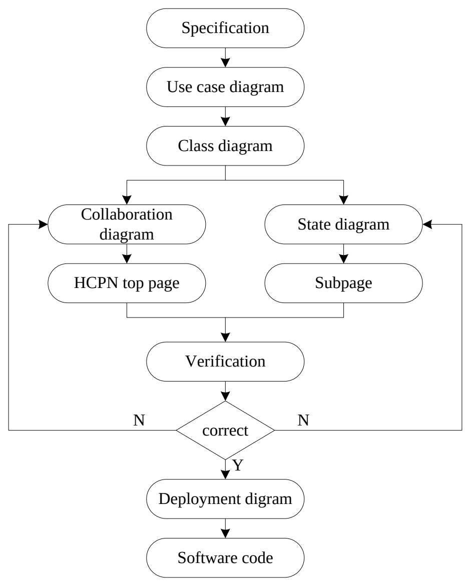

Figure 1. Modeling framework

Shown as Figure 1, we first constitute the system specification according the requirements of the system. Then the use case diagram of UML model is designed to capture the traffic guidance requirements corresponding to system specification, and the class diagram is designed according to the existing objects and their functions.

After that the collaboration diagram and state diagram are designed on the basic of class diagram, we map the two diagrams to HCPN model. Using the simulation tool and math theory we may get some characteristics of the model so as to decide whether the model satisfy the system requirements, such as deadlock, reachability, boundedness. According to the simulation result of HCPN model we may improve the corresponding UML model. Then we repeat the process of “map, verification and improve” until the verification satisfy the requirements.

This approach integrates the merits of UML and Petri Nets. Petri Nets is used to do quantitative and qualitative analysis, then software code can be gotten from UML model.

3. Construction of System Model

The traffic guidance system based multi-agent is composed of a city center agent, some region center agents, many traffic intersection controller agents and many vehicle agents. All vehicle agents are dynamically divided into one or a number of region groups by city command center agent; each group has a region center agent as a leader. Vehicle agents need identify and select the best path by all information from environments.

Region center agent must communicate with intersection controller agents so as to get all traffic information, and transmit the information to city center agent.

Region center agent can communicate with city center agent and all vehicle agents of the region at any time, and realize the information exchange. In traffic guidance system, each vehicle agent must register their information, including name, address, interfaces and relative services to the city and region center agent. When a vehicle agent enters or exits a region, region command center must add or remove the registered information, thus it only keep one dynamic information table that will not cause confusion. City center agent should have the highest priority to control and communicate with every region center agent.

In our systems, route control algorithm is discomposed into the design of region route control agents, and different control algorithms are assembled by different region route control agents. Each region center agent doesn't hold all route control agents, but "control on demands", and it only possesses the region route control agents which are currently needed.

In the following, all kinds of diagrams of UML model are presented.

3.1 The use case diagram

Shown as Figure 2, it is the use case diagram of traffic guidance systems, and includes a vehicle agent, a city command center agent, a region center agent and an intersection controller agent. The use cases which are drawn as eclipses represent the scenarios of the systems.

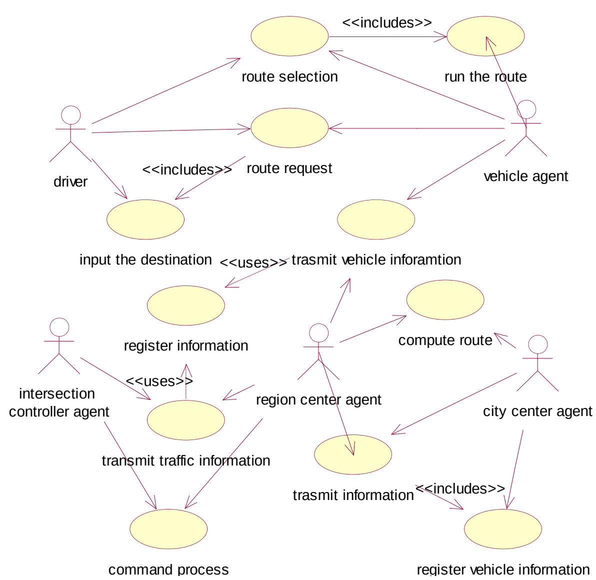

Figure 2. The use case diagrams of the system

In Figure 2, the request from the driver "input the destination" causes the vehicle agent to carry out "route request", then cause region and city center agent to carry out "route compute". After getting feasible routes, the drive carries out "select the route", and the vehicle carries out "receive the route" and "running the route".

3.2 The Static diagrams

The class diagram is the most important static diagram in UML model. It describes all kinds of static relationships among objects, and the attributes and operations of a class.

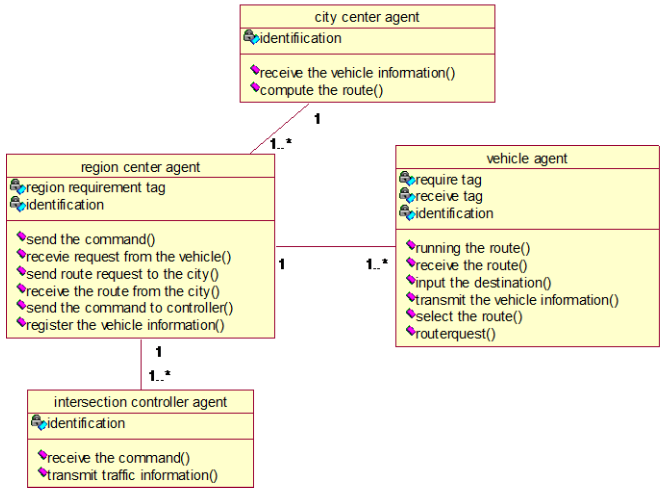

Figure 3. The class diagrams of the system

Figure 3 is a class diagrams of the traffic guidance systems based on Multi-Agent. Figure 3 includes the following four classes: city command center agent, region center agent, vehicle agent, and intersection agent. The "vehicle agent" class includes six operations corresponding to the six use cases of the actor "driver" and "vehicle agent" in Figure 2. The "city center agent" and "intersection controller agent" class are both simple. However, the region center agent add two internal functions "receive route from the city" and "send route to the city" on the basic of Figure 2 so as to make the relation of city and region center agent more clear.

3.3 The dynamic diagrams

Figure 4 is the collaboration diagram of requiring the route. The collaboration diagram represents the dynamic structure among the class instances. Figure 4 only represents the case in which the driver requests a route whose destination is not in local region.

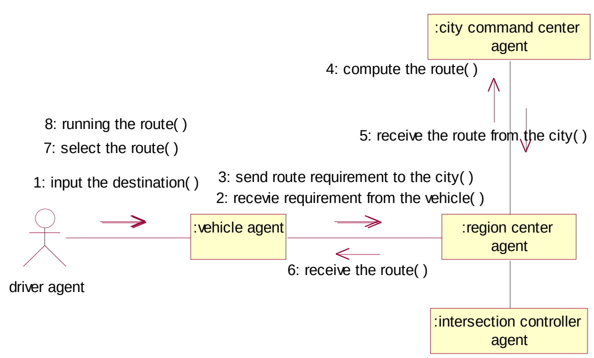

Figure 4. The collaboration diagrams of the vehicle

In Figure 4, at first, after the driver inputs a destination, the vehicle receives the request and sends it to the region center agent. Then city center agent disposes all possible city routes into region routes using its algorithm, and region center agents receive, compute and get the region routes. Moreover, the vehicle agent receives all region routes, assembles them. Finally, the driver selects a route, and makes the vehicle agent run according to the route.

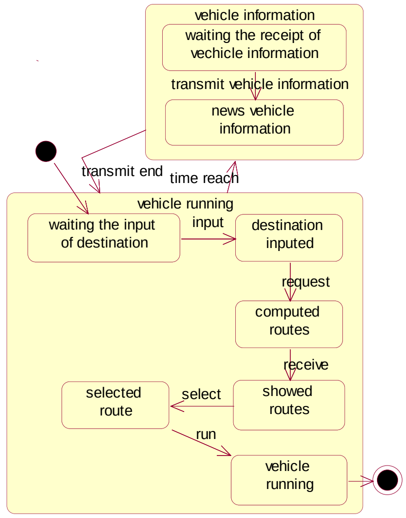

Figure 5. The state diagrams of the vehicle agent

By the collaboration diagram, we may get the relationship among objects. However, in order to get the conversion relationship of various state of an object, the state diagram must be constructed. Figure 5 is the state diagram of the vehicle agent. It has two parallel states including "vehicle information", and "vehicle running". In the state "vehicle running", it includes six sub-states, and the sub-state converts from "waiting the input of the destination" to "vehicle running" by continuous trigger of events.

Figure 6 is the state diagram of region center agent. It has three states: "city route", "command send" and "information transmission".

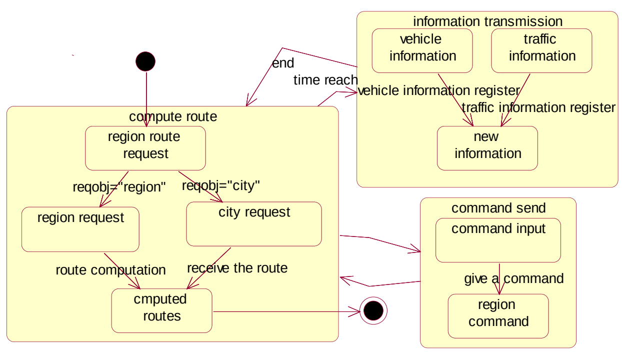

Figure 6. The state diagrams of the region center agent

After the destination request arrives, the region center agent first decides whether the route is city route or region route, then the diagram converts from "region route request" to the corresponding sub-state.

During the driving, because of the changes of traffic information, in some road segments of the route there may exist some unexpected incidents, such as traffic jams, traffic control, region command center agent can receive the warning message that this road has some abnormity from intersection agent on this road segments. The region center agent must adjust the original route based on the message and set down a new route so as to complete the mission successfully.

However, the aim of the paper is to research the modeling of traffic guidance systems, for simplicity, so we only consider one traffic and vehicle information transmission during one route request.

3.4. Mapping methods

In the following we give the steps by which HCPN model can be constructed from UML model.

1) Obtain the substitution transitions of top page of HCPN model from UML class diagram.

In the class diagram the class that contains several operation functions is mapped to a replacement transition in HCPN model, and the name of the replacement transition may be same with the class name. Each operation in the class that is mapped to a substitution transition of the HCPN is replaced by a transition of the sub-page of this substitution transition. In addition, the class that only contains one operation function is mapped to an ordinary transition with same name [9].

The class that only has the static property in class diagrams is mapped to the place whose color sets is corresponding to the color sets of the class.

In HCPN the direction of the arc is decided by the association direction of the class diagram.

2) Map the state diagrams to a sub-page.

State diagram displays the details of the dynamic behavior of objects, so it can easily be mapped to sub-page of HCPN.

According to the characteristics of the state diagram and the HCPN, a conversion event in UML state machine diagram is mapped to a transition of the Petri Nets, the source state is mapped into a pre-set place of the transition, the target state is mapped to a post-set place of the transition, and the conversion element is mapped to the two arcs which are from the pre-set place to the transition and from the transition to post-set place. The state diagram reflects states of the objects and external events which make states changed in UML. In HCPN the place and transition of sub-page reflects these relationships.

3) The whole HCPN of the system is constructed by UML collaboration diagrams and many sub-pages.

The step is to connect all the sub-pages so as to form the HCPN model of the whole system.

3.5. HCPN model

In Figure 3, there are four classes, according to the mapping rules of the UML to HCPN, HCPN model of traffic guidance should includes four substitution transitions: vehicle agent, intersection agent, region center agent and city center agent.

For the substitution transition “vehicle agent”, the sub-page is shown as Figure.7. The place “vehicle destination” represents the route request of driver, and it makes the transition “input” and “request” enabled. The place “input_ tag” models the execution sequence of the transition “input” and “select”. Once region or city center agent finished the computation of requested destination, the place “computed routes” will have a token whose type is route list. Then the transition “receive”, “select” and “run” will be fired sequentially. After the vehicle is run according to selected route, the place “vehicle running” will get a token.

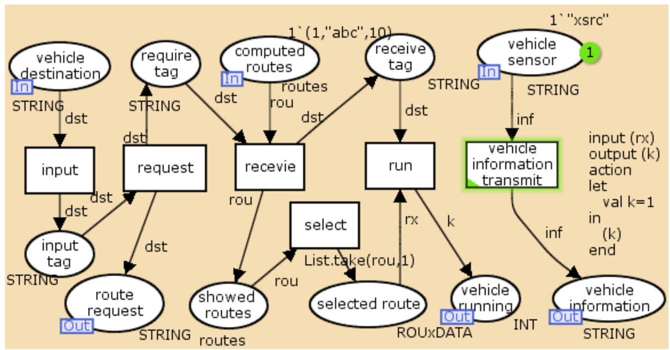

Figure 7. The sub-page of vehicle agent

Figure 8 is the sub-page of the substitution transition “region center agent”. Once the vehicle requests a route, the place “route request” will get a token. If the token of the place “region city tag” is “region”, the token will make the transition “request compute” enabled, then the place “computed routes” gets a token which represents that route request has been finished.

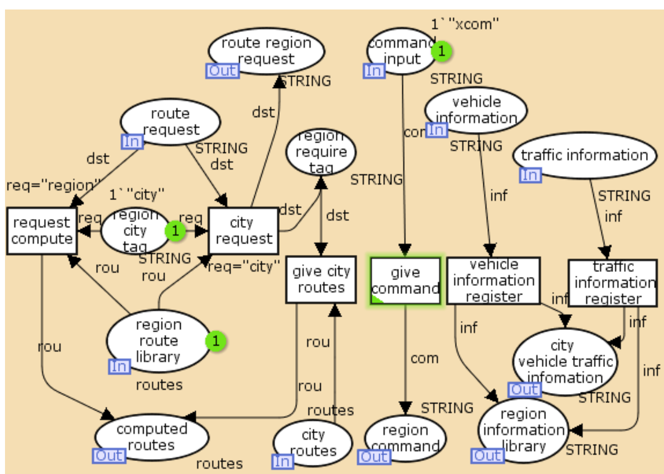

Figure 8. The sub-page of region center agent

If the token of the place “region city tag” is “city”, it makes the transition “city request” enabled. It makes region center agent request the routes to city center agent. After the city center agent transmits the routes to the place “city routes”, the place “give city routes” is fired, and the place “computed routes” gets a token.

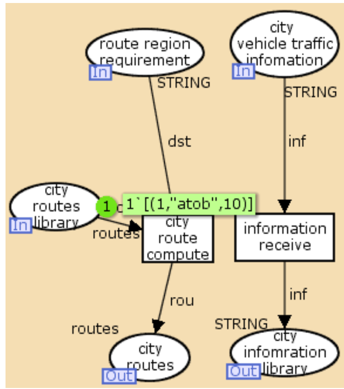

Figure 9. The sub-page of city center agent

Figure 9 is the sub-page of the substitution transition "city center agent". Once it gets the request from the region center agent, the transition "city route compute" is fired, and transmits the computation result to all relative region center agents. The transition "information receive" models the collection of vehicle and traffic information.

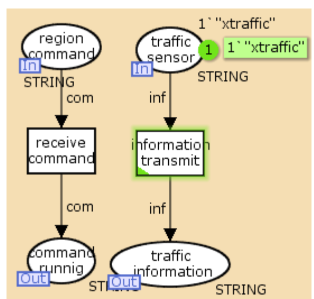

Figure 10. The sub-page of intersection controller agent

Figure 10 is the sub-page of the substitution transition "intersection controller agent". It models the receipt of command from the region center agent, and the collection of traffic information.

Of course, we may further get more HCPN models of traffic guidance system. However, the above models have made us implement formal verification and gotten reliable UML model.

After getting all the sub-pages of each substitution transition of HCPN model, the top page of HCPN model is gotten from the collaboration diagram of UML model.

In the following, the top page which is gotten by the collaboration diagram of UML model is shown as Figure 11. In Figure 11, the name of substitution transition is same with that of the class in Figure 3. The relation among the substitution can be gotten from the collaboration diagram of UML model.

In Figure 11, four substitution transitions are corresponding to the above four sub-pages.

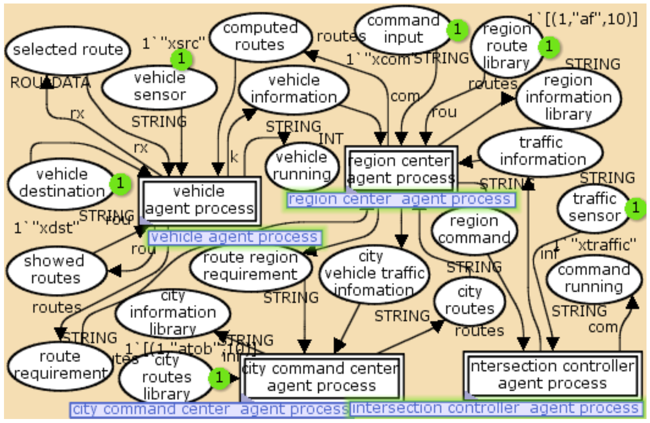

Figure 11. The top page of the system

3.6 The verification of Petri Nets model

Various analysis methods of Petri Nets can be used to do the strict qualitative analysis and validate the Petri Nets model that we get.

CPN Tool is the simulation software of HCPN model, and provides two analysis tools. One is to simulate the running of the model, and observe the changes of the mark of each place after a transition is fired. So we may get the properties of the model by single step or continuous running of transitions. The other is the State analysis tool which is called "Statespace". It can generate all states of HCPN model so that we can get all the properties by state graph. In the following, we use the two analysis method to verify the HCPN model which is gotten from UML model.

Shown as Figure 12, it is the running result after executing 17 steps of transitions continuously. The place vehicle running gets a token "1" which represents that the vehicle has selected a route and is running.

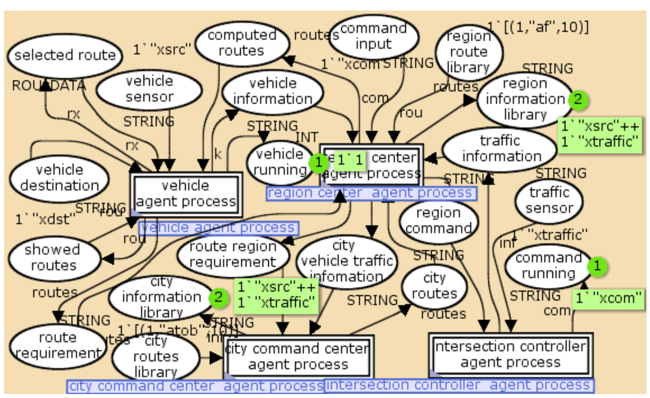

Figure 12. The running result of completing a route request

Shown as Figure 13, it is the part result after executing the analysis of calculating the state space and SCC graph. It represents that the state space and SCC graph have both 432 nodes and 1320 arcs. The upper boundedness of the place vehicle running is "1" which represents that the vehicle has selected a route and is running. The dead and home marking nodes are both 432. Since the state space has only 432 nodes it must be end node. The existing of dead transition “request compute” is that the token of the place “region city tag” is “city” so that the transition is not executed.

Statistics

State Space
Nodes: 432
Arcs: 1320
Secs: 0
Status: Full
Scc Graph
Nodes: 432
Arcs: 1320
Secs: 0

Boundedness Properties

Best Integer Bounds	Upper	Lower
Top'city_information_library 1	2	0
Top'city_routes 1	1	0
Top'city_routes_library 1	1	0
Top'city_vehicle_traffic_infomation 1	2	0
Top'command_input 1	1	0
Top'command_running 1	1	0
Top'computed_routes 1	1	0
Top'region_command 1	1	0
Top'region_information_library 1	2	0
Top'region_route_library 1	1	0
Top'route_region_requirement 1	1	0
Top'route_requirement 1	1	0
Top'selected_route 1	1	0
Top'showed_routes 1	1	0
Top'traffic_information 1	1	0
Top'traffic_sensor 1	1	0
Top'vehicle_destination 1	1	0
Top'vehicle_information 1	1	0
Top'vehicle_running 1	1	0
Top'vehicle_sensor 1	1	0
region_center_agent_process'region_city_tag 1	1	0
region_center_agent_process'region_require_tag 1	1	0
vehicle_agent_process'input_tag 1	1	0
vehicle_agent_process'receive_tag 1	1	0
vehicle_agent_process'require_tag 1	1	0

Home Properties

Home Markings [432]

Liveness Properties

Dead Markings [432]
Dead Transition Instances
region_center_agent_process'request_compute 1
Live Transition Instances None

Fairness Properties

No infinite occurrence sequences.

Figure 13. The part analysis result of state space

We know that the simulation result is correct by the above analysis. So we think that UML model is also correct. The correct source code framework may be gotten from UML model.

4. Conclusion

In the traffic guidance systems based Multi-Agent, we have presented a methodology to support formal validation of UML model. The main idea is to map a HCPN model from UML diagrams so as to utilize the analysis techniques of HCPN. We construct key UML components of traffic guidance systems, and discussed the mapping activities: 1) Generation of sub-page of each class instance, 2) Generation of top page of the model, and 3) Combine top page and sub-pages to create a system-level model. The proposed methodology helps us to model and analyze the traffic guidance systems based Multi-Agent.

References

[1] F. Yu, “The Application of Agent Technology in Intelligent Traffic”, Henan University of Science and Technology, Luoyang, **(2009).**

[2] X. F. Zha, “Vehicle guidance system and key technology research”, jiangsu university, Zhenjiang, **(2007).**

[3] F. Y. Wang, C. H. Wang, “Agent-Based Control Systems for Operation and Management of Intelligent Network-Enabled Device”, Proceedings of IEEE International Conference on Systems, Man and Cybernetics, Washington DC, United States**, (2003)** October 5-8.

[4] H. T. Zhang, F. Y. Wang and Y. F. Ai, “An OSGi and Agent Based Control System Architecture for Smart Home, Proceedings of IEEE International Conference on Networking, Sensing and Control, Tucson, AZ, United states US, **(2005)** March 19-22.

[5] Y. L. Cai, Z. H. Yu and X. M. Zhang, “Formal Modeling Methodology for Multi-Agent Systems”, Journal of System Simulation, vol. 19, no. 14, **(2007),** pp. 3151-3157.

[6] D. X. Xu, R. Volz, T. Ioerger, *et al.*, "Modeling and Verifying Multi-Agent Behaviors Using Predicate/Transition Nets", Proceedings of International Conference on Software Engineering and Knowledge Engineering, Ischia, Italy**, (2002)** July 15-19.
[7] J. S. Lee and P. L. Hsu, IEEE Transaction on Control Systems Technology, no. 12, **(2004),** pp. 293.
[8] F. Basile, P. Chiachio and D. D. Grosso, "Computer Standards & Interfaces", no. 31, **(2009),** pp. 528.
[9] J. A. Aaldhana, A. M. Shatz, "UML Diagrams to Object Petri Net Models: An approach for Modeling and Analysis", Proceedings of International Conference on Software Engineering and Knowledge Engineering, Kaiserslautern, Germany, **(2000)** June 16-19.

