'94 KACC (1994. 10. 17 ~ 20)

Hierarchical Petri Netting for Design and Supervision of an Automated Vehicle System

Y.Sakai,* M.Kitazawa,* M.Nakamura,** and N.Matsuda***

* Dept. of Mechanical Engineering, Yamaguchi University, Ube, Yamaguchi 755, Japan
FAX:+81-836-35-9411
** Industrial Technology Institute, Yamaguchi Prefectural Government, Yamaguchi 753, Japan
FAX:+81-839-22-6813
*** Graduate School, Yamaguchi University, Ube, Yamaguchi 755, Japan
FAX:+81-836-35-9411

Abstract

A hierarchical Petri net is utilized in supervising an automated vehicle system. The supervisory system is supported by computer networking in order to facilitate necessary processing, and consists of control flow level and computer allocation level so that a designer and an operator can easily build and/or access to each level. There are two modes of utilizing Petri net here in this paper. One is to employ it in designing the control system, in order to optimally allocate computers in every stage of processing. The other is for supervision of the system in operation, in order for the operator to be in a easy-to-comprehend environment of operation. The effect of these two modes of utilizing Petri net is examined.

1. Introduction

With the development of computers and network facilities, distributed information processing is prevailing. This fact has an effect on flexibility of building information processing systems. At the same time, it has another effect that it makes systems complicated, and hence design, control, and supervision need deep knowledge and higher techniques. Utilization of Petri net may partly be a solution. It provides visual understanding of the structure and the behavior of a distributed system.

In this paper, by parallel processing, designing and controlling a supervisory system for an automated vehicle system is discussed. In Section 2, a brief explanation is made on the structure of the present Petri net tool, in Sections 3 and 4, on an outline and the design of the supervisory system, and in Section 5, on how to use the present system.

2. Petri Net Tool

The Petri net tool[1][2][3] here builds and edits a Petri net model for the system dealt with. The Petri net tool here has process-place, status-place, transition, arc, and tokens as its elements. A process-place, an arc, and a token can be given an attribute. An attribute, as will be described later in, is differentiated with each other by coloring. As in Fig.1, the present tool is layered in two levels. The first level is for describing computers with their relations with each other, and will be called computer-allocation level.

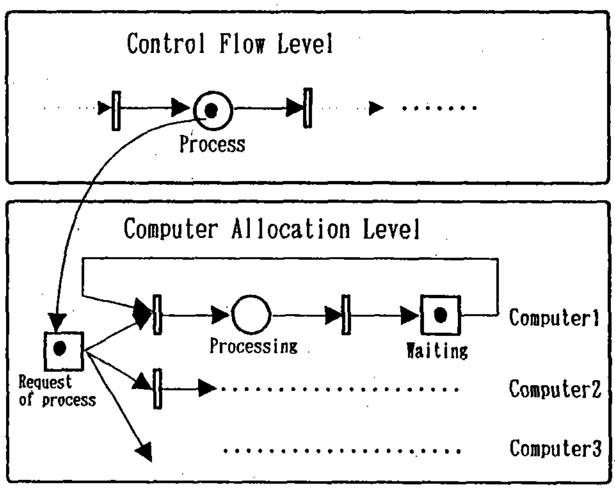

Fig .1 Schematic of hierarchical Petri net

The second level is used to model the system dealt with, and is called control-flow level. Layering the tool makes a human operator feel easy in comprehending what is going on in the system in operation. And it is also advantageous in designing a system, avoiding the risk of building a complicated model of the system on the tool.

a)Process-Place

Here in this paper, a place is classified into process-place or status-place according to its role in the network. In the level of control flow, a process-place indicates that the place is the starting of a processing. And the content of process depends on its attribute indicated by coloring. As in Fig.1, when a token comes to a process-place, then the status-place in the level of computer allocation, which represents the demand of that processing, obtains a token of the same attribute. A process-place in the level of computer-allocation represents a computer and the occurrence of a token at that place indicates the occurrence of the processing demand.

b)Status-Place

A status-place indicates the occurrence of demand of processing, and admission of starting processing.

c)Token

A token in the level of control flow indicates a flow of processing. It possesses an attribute to follow only the arcs corresponding to the particular attribute. In the level of computer-allocation, an attribute is in one to one correspondence with the content of processing.

Table 1 Petri net Symbols

	NAME	Control flow level	Computer allocation level
		Definition of element	
○	Process-place	Process	Computer
□	Status-place	Status	Status Request of processing
▯	Transition	Event	Event
→	Arc	Direction of data flow	Direction of data flow
●	Token	Data	Process

d)Arc

An arc possesses an attribute in order for a token to determine which arc to follow.

Table 1 shows symbols of those elements.

3.Distributed Processing System by Petri Net

Fig.2 shows a schematic of the present system. In this system, local area network (LAN) is employed which runs on Windows. Computers mutually connected on the LAN execute data processing and serve as an interface between an operator and the vehicle system. Independent of those computers, a computer is set to concentrate on the Petri net tool and it enables the control of the overall distributed system. Data transference is done via the server of the LAN. This system is applied to the supervision of an automated vehicle system which will be abbreviated as SAVS[4]. The outline is described below.

The present system uses a bird's-eye view from a video camera set over the vehicle and an obstacle for detection of their locations. The data from this processing is used to determine an optimal course to take and necessary steering angle and speed at every position. Such decisions are sent to the vehicle wirelessly. The major functions of the present system are:

Function 1.Detection of an obstacle and the vehicle.[5]

Function 2.Setting an optimal course.

Function 3.Tracing a course determined by Function 2.

Among those, course-setting is done only at the starting and in the case of the change in the location of the obstacle. Fig.2 shows the schematic of the whole system, and the explanation will be given in detail in the next section. A route for the vehicle to follow is shown on CRT, and an optimal steering angle is determined at every moment.

4.Modeling the System by Petri net

In the control-flow level, Functions 1 to 3 are interpreted as follows, referring to Fig.2. A scene is inputted from a video camera. The location of the vehicle is identified by the image inputted, and the data is transferred to the file server. If there is found change in an obstacle's location, then the data is also transferred to the server. This is the first stage of controlling the vehicle's behavior. Next, in the case of occurrence of the change in an obstacle, a detour is set in order to avoid collision with the obstacle. This must be made before

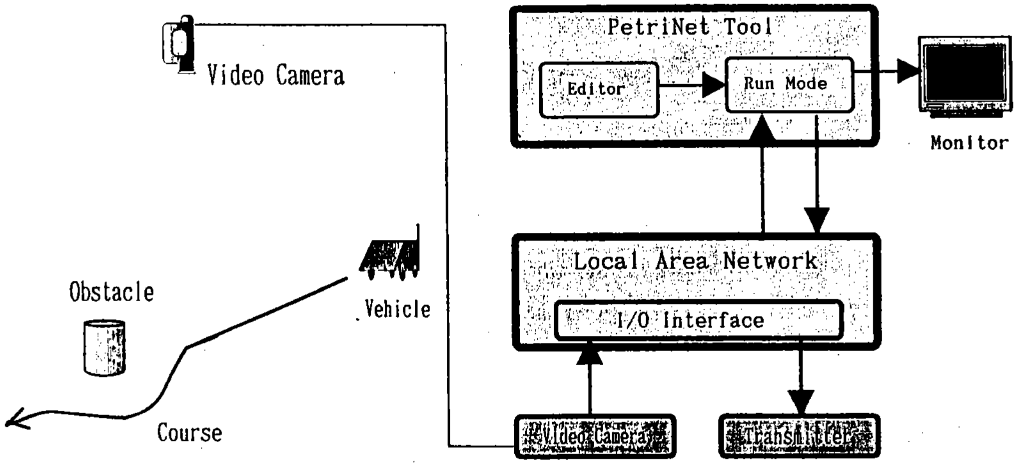

Fig.2 Schematic of an automated vehicle system

providing steering angle to the vehicle. The last to be processed is the determination and transference of necessary steering angle. The steering command is issued every second, and the distance which the vehicle proceeds in that time is identical to the vehicle's wheelbase. While the transference of the data for steering must be made regularly, the time required for image processing depends on the situation, since the existence of and/or the change in the location of the obstacle needs additional processing compared with the case in which only the vehicle is detected in a particular scene. Fig.3 shows the model of the system behavior as a colored Petri net. The procedures for system running are executed by one server and three clients. Each of those procedures must be assigned to any one of three clients as jobs. Management of job assignment is also done by a Petri net named the level of computer-allocation. This management level is run in accordance with the state of the system described by the model of the control-flow level in Fig.3, and is as follows: If a token comes to any process-place, then a token appears at a status-place which indicates the demand of that processing. Necessary jobs for this processing is allocated to clients which are free from job at that moment. Also, some clients can do a certain special job by coloring related arcs as its attribute. In fact, one of three clients specializes in data processing. And major jobs of other two are image processing and sending steering commands to the vehicle, respectively. Petri net model for those processes are shown schematically in Fig.4. An additional computer other than the four computers, i.e., three clients and a server, is required for running Petri net. So the number of computers necessary for the present system is five in all.

5. Simulation

In preceding sections, a methodology is described, for constructing Petri net model of the system. Here in this section, simulation of the system operation is discussed. How the simulation is made is described first.

In simulating SAVS, the time necessary for inputting image and the time interval of sending steering commands are estimated 1.2s and 1.0s, respectively. An obstacle is assumed to exist without changing its location. Hence, a routing of the vehicle is done just at the beginning of a run of the vehicle.
In the SAVS, the processes of image processing and determining steering angle are independently processed, but the difference in timing of demand between those processings may affect early-finding of an obstacle and quick-processing of necessary response. From this standpoint, evaluation of the system performance is made by checking transition cycles of T2, which denotes firing sending steering data, and T9, which denotes firing finishing the vehicle's location data. Two schedules, Schedule 1 and Schedule 2, are compared with each other:

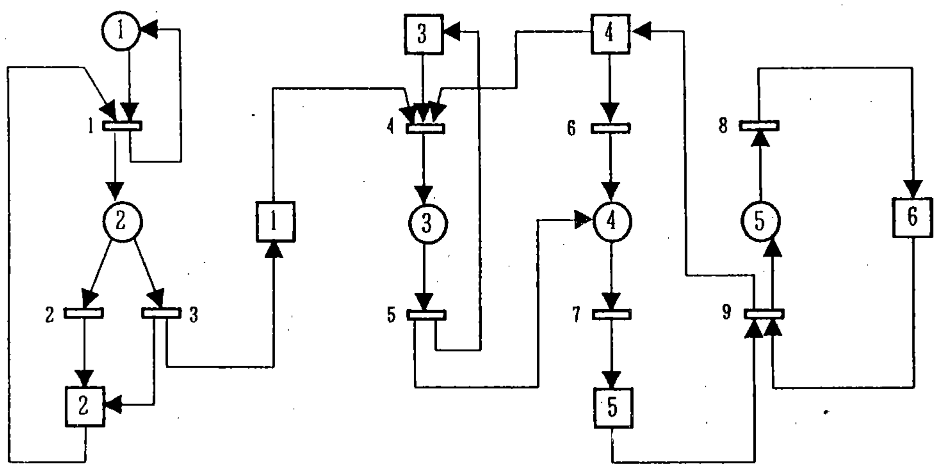

Fig.3 Control flow level for SAVS

Nomenclature

CONTROL-FLOW LEVEL

Process-places

PP1=Inputting a scene
PP2=Acquiring the locations of the vehicle and an obstacle
PP3=Routing
PP4=Determining steering angle
PP5=Sending steering data

Status-places

SP1=An obstacle's location is already acquired
SP2=Starting processing PP2 is admitted
SP3=Starting processing PP3 is admitted
SP4=The vehicle's location is already acquired
SP5=Steering angle is already determined
SP6=Acceptance of steering data is admitted

Transitions

T1=Starting inputting a scene and detection of the locations of the vehicle and an obstacle.
T2=Finishing detection of the locations of the vehicle and an obstacle (Obstacle is stopping)
T3=Finishing detection of the locations of the vehicle and an obstacle (Obstacle is moving)
T4=Starting detouring
T5=Finishing detouring
T6=Starting determination of steering angle
T7=finishing determination of steering angle
T8=Finishing steering data transference
T9=Starting steering data transference

(1)**Schedule 1**

Computer 1 specializes in processing inputted scenes. Also, sending steering data is exclusively done by Computer 2. Remaining jobs are done by Computer 3.

(2)Schedule 2

Part of image processing is assigned to Computer 3, which is time-consuming. Table 2 shows those schedules and their evaluations. Fig.5 shows a simulated behavior of the vehicle on the CRT.

In Table 2, the cycles of firing T2 and T9 are set as their averages, respectively.

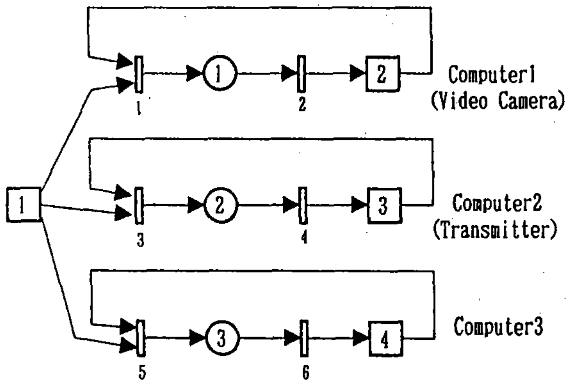

Fig.4 Computer allocation level for SAVS

Nomenclature

COMPUTER-ALLOCATION LEVEL

Process-places
PP1=Computer 1 is occupied
PP2=Computer 2 is occupied
PP3=Computer 3 is occupied

Status-places
SP1=Processing is demanded
SP2=Computer 1 is ready to accept demand
SP3=Computer 2 is ready to accept demand
SP4=Computer 3 is ready to accept demand

Transitions
T1=Starting processing on Computer 1
T2=Finishing processing on Computer 1
T3=Starting processing on Computer 2
T4=Finishing processing on Computer 2
T5=Starting processing on Computer 3
T6=Finishing processing on Computer 3

Table 2 Two schedules and their parformances

	Schedule 1	Schedule 2
Computer	Schedule(Number indicates P-Place)	
Computer1	1, 2	1, 2
Computer2	5	5
Computer3	3, 4	3, 4, 2
Transition	Cycles of firing (sec)	
No2	2. 325	1. 890
No9	1. 148	1. 222

In Schedule 1, the time interval of steering data transference has a delay of 0.148s from the prescribed 1.0s. This comes from the existence of data transference and execution of the Petri net. So, the vehicle has a blackout time of 0.148s. Schedule 2 needs longer time interval for data transference. This comes from the fact that Computer 3 shares the load of image processing with Computer 1. Referring to the row of firing No.2, Schedule 2 surpasses Schedule 1. This is advantageous in early-finding of an obstacle and quick-response for avoiding collision. In this respect, Schedule 2 is better than Schedule 1. The comparison of these two schedules are only a part of seeking for an optimal scheduling. What schedule is optimal depends on the purpose and the environment of the system.

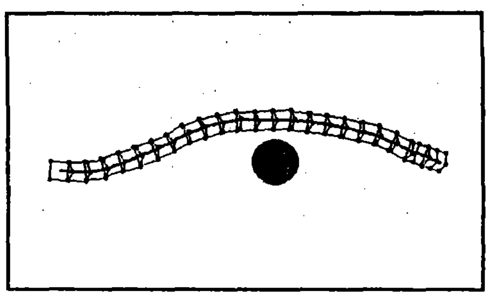

Fig.5 A simulated routing

6. Conclusions

(1)Modeling the system

Using a multi-layered Petri net model makes it easy to understand visually what is going on in the system.

(2)Scheduling processes

Simulating the system's behavior evaluates its performance. So, an efficient scheduling can be set.

(3)Monitoring the cycle of the process

Monitoring the cycle of the process enables system management without the operator's knowing unnecessarily detailed structure of the system.

References

1.H.Shiizuka, Practical Petri Nets, Korona-sha, 1992, (in Japanese)

2.J.L.Peterson (Translated into Japanese by A.Ichikawa and S.Kobayashi), Petri net theory and the modeling of system, Kyoritsusyuppan Ltd., 1984.

3.Y.Nagao, et al., "Application of Petri Net to Automated Guided Vehicle Control System," System, control and information Vol.5, No11, 1992, pp.469-479.

4.Y.Sakai, M.Kitazawa, Y.Okuno, and M.Nakamura, "Experience-Based Vehicle Operation," Proc. of ICC&IE-94, Ashikaga, Japan, March 7-9, 1993, pp.150-155.

5.Y.Sakai, M.Kitazawa, and Y.Okuno, "Object's Orientation and Motion for Scene Understanding," Proc. of '93 Korean Automatic Control Conference, Seoul, Korea, Oct.20-22, 1993, pp.265-270.