Remote Supervisory Control of the Human-in-the-Loop System by Using Petri Nets and Java

Jin-Shyan Lee and Pau-Lo Hsu, *Member, IEEE*

***Abstract*—For Internet-based remote control systems, certain human operations may violate desired safety requirements and result in catastrophic failure. For such human-in-the-loop systems, this paper proposes a systematic approach to developing supervisory agents which guarantee that remote manual operations meet safety specifications. By applying the mutual exclusion concept, the Petri net approach is used to model, design, and verify a supervisory system which prevents human errors. Then, Java technology is adopted to implement the supervisor as an intelligent agent for online supervision of the remote control system. To demonstrate the feasibility and practicability of the developed supervisory approach, we apply it to a rapid thermal processor controlled over the Internet.**

***Index Terms*—Human in the loop, Java, Petri nets (PNs), rapid thermal processor, remote monitoring and control, supervisory control.**

I. Introduction

RECENTLY, due to the rapid development of Internet technology, system monitoring and control no longer needs to be conducted within a local area. Several remote approaches have been proposed which allow people to monitor the processes of manufacturing systems from great distances and to perform maintenance functions in hazardous environments without exposure to dangers [1]–[6]. Typically, an Internet-based control system (remote access using IP-based networks) is a "human-in-the-loop" system since people use a general Web browser or specific software to monitor and control remotely located systems [7]. As shown in Fig. 1(a), the human operator is involved in the loop and sends control commands according to the observed status displayed by the state or image feedback. Research results indicate that approximately 80% of industrial accidents are attributed to human errors, such as omitting a step, falling asleep, and improper control of the system [8]. However, the Internet-based control literature provides few solutions for reducing or eliminating the possibility of human errors.

An Internet-based control system is inherently a discrete-event system (DES), that is, a dynamic system with state changes driven by occurrences of individual events. Supervisory control theory provides a suitable framework for analyzing DES [9]–[11] and, in this paper, we propose applying a supervisory design to the present remotely controlled, human-in-the-loop system so as to prevent abnormal operations from being carried out. Fig. 1(b) shows the proposed supervisory control scheme for a remotely located system with the human in the loop. First, the supervisory agent acquires the system status and makes the decision to enable/disable associated events to meet the required specifications, typically safety requirements. The human operator is then only allowed to perform the enabled events to control the system. The role of a supervisory agent is to interact with the human operator and the controlled system so that the closed human-in-the-loop system meets the required specifications and to guarantee that undesirable executions do not occur.

Most existing methods for supervisory system design are based on automata models. However, these methods often involve exhaustive searches of overall system behavior and result in state-space explosion problems. One way of dealing with these problems is to model the DES with Petri nets (PNs) [12], [13]. PN modeling is normally more compact than the automata approach and is better suited for modeling systems with parallel and concurrent activities. In addition, PN has an appealing graphical representation with a powerful algebraic formulation and, thus, it has generated intense interest on the part of many researchers [14]–[18]. In this paper, PNs are used in designing the supervisory system, yielding a compact and graphical model for the remote control system.

The supervisory control literature provides few indications as to how to implement the generated abstract supervisory control model in real applications. However, this paper demonstrates the feasibility and practicability of the proposed supervisory control by applying it to a rapid thermal processor (RTP) in semiconductor manufacturing, where the supervisory agent is implemented using Java technology on an industrial programmable logic controller (PLC) [19], [20]. Our approach ensures that remote control operations via the Internet meet the given safety requirements.

II. PN-Based Supervisor Design

This section first shows the required control modes and specification types for remote supervisory control. Then, the PN-based procedure for designing the supervisor is described with a simple door–valve system for an RTP.

Manuscript received June 30, 2002; revised November 15, 2002. Abstract published on the Internet March 4, 2003. This work was supported in part by the National Science Council, R.O.C., under Grant NSC 90-2213-E-009-101, and in part by the MOE Program for Promoting Academic Excellence of Universities under Grant 91-E-FA06-4-4.

The authors are with the Department of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu 300, Taiwan, R.O.C. (e-mail: plhsu@cc.nctu.edu.tw).

Digital Object Identifier 10.1109/TIE.2003.812283

0278-0046/03$17.00 © 2003 IEEE

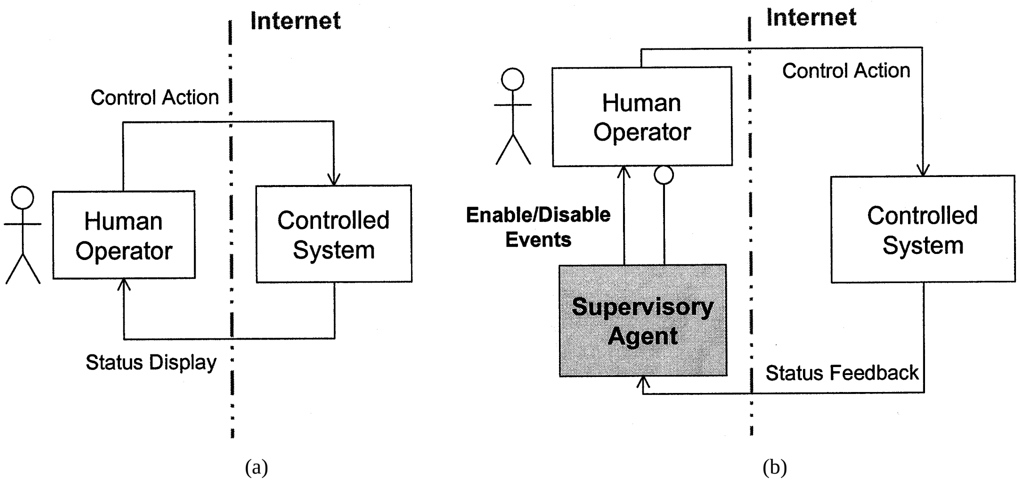


Fig. 1. (a) Typical remote control system with the human in the loop. (b) Proposed remote supervisory control scheme.

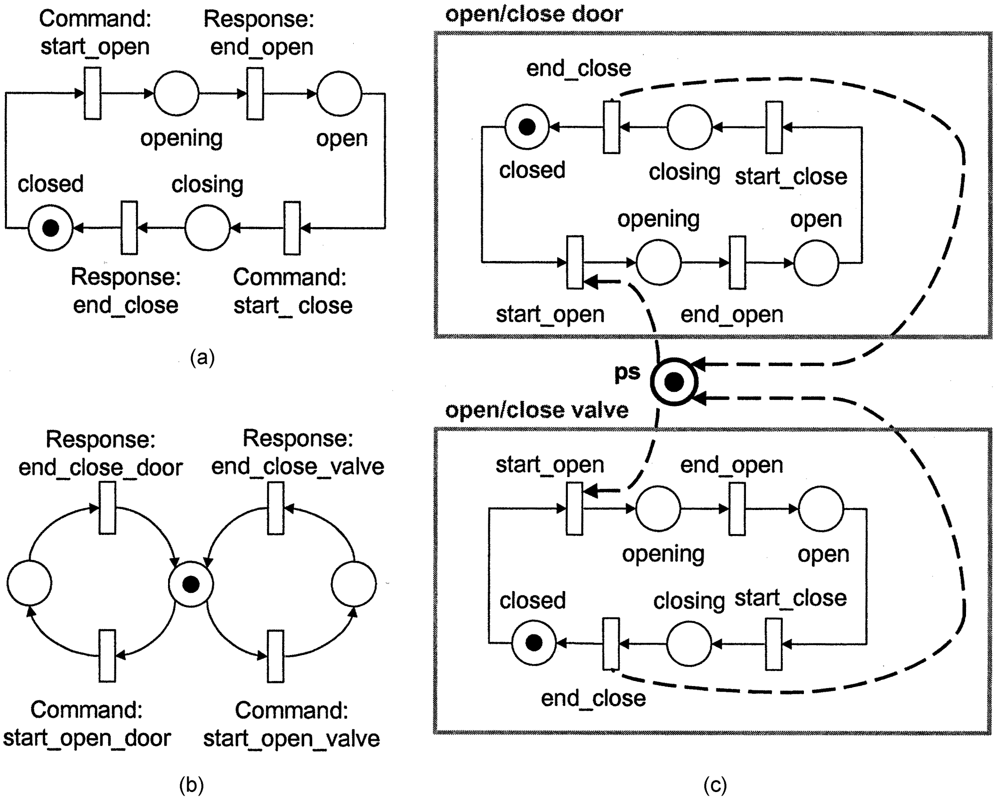


Fig. 2. (a) General model for door and valve components. (b) Mutual exclusion specification model. (c) Composed supervisor for the door–valve system.

A. Control Modes

For remote control via the Internet, we are interested in the following two control modes.

1) *Automatic control mode:* When the system is in automatic control mode, the automatic controller autonomously controls the manufacturing process without user intervention (the human operator only needs to push a button to start the control cycle). Generally, an active sequence controller is used to automatically complete several operations in a certain order.
2) *Manual control mode:* A system often must be open to manual control for various purposes, such as for test runs and fault diagnosis. Here, we examine the case in which the user can directly perform each operation. To ensure that safety constraints are not violated, the supervisory agent is online executed to acquire the system status and decide to either enable or disable specific operations.

B. Specification Types

The objective of the supervisor is to restrict the behavior of the system so that it is contained within the set of admissible states, called the specification. Two types of specification are classified as follows.

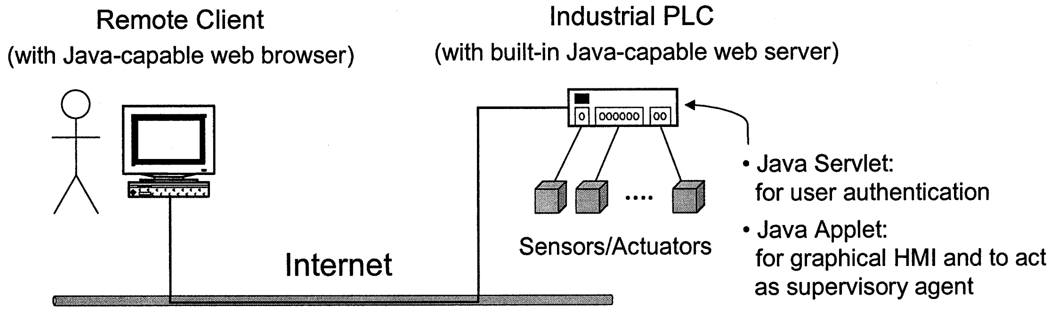


Fig. 3. Implementation architecture of the remote supervisory control system.

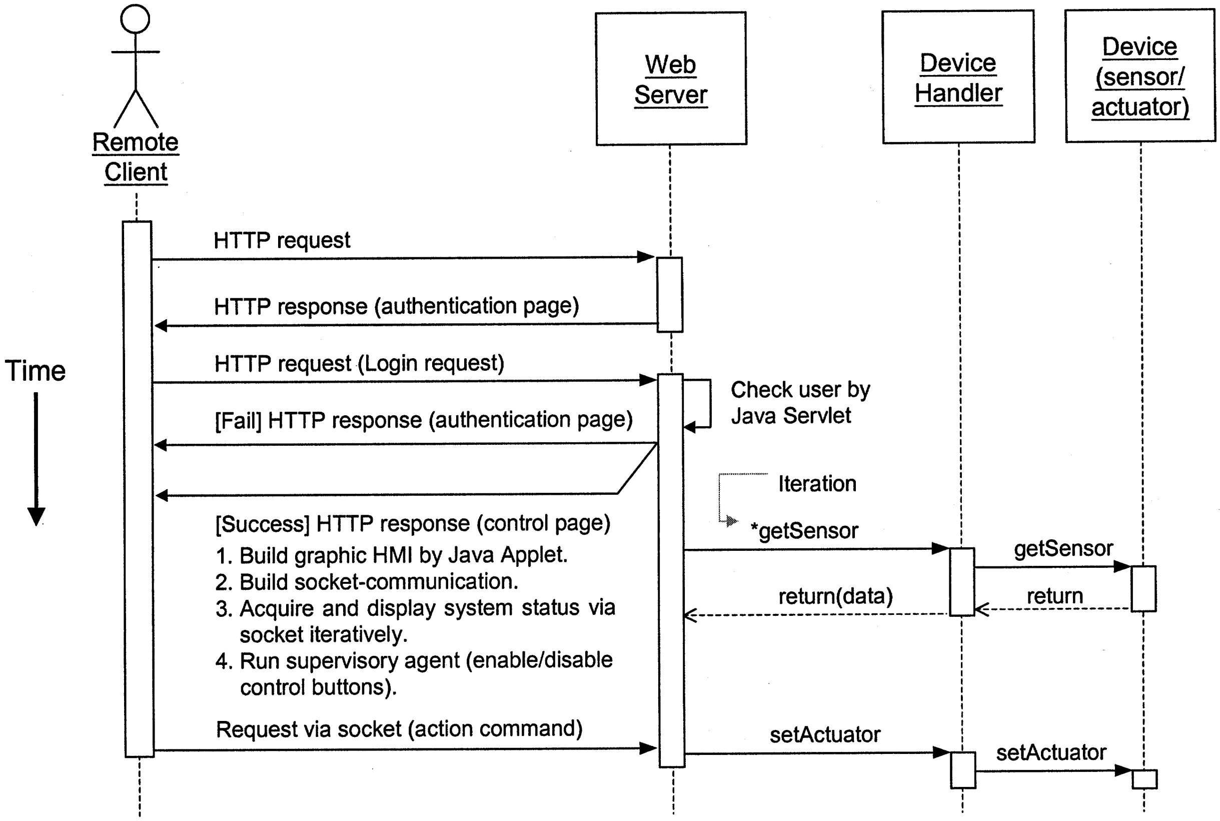

Remote Client
Web Server
Device Handler
Device (sensor/ actuator)
Time
HTTP request
HTTP response (authentication page)
HTTP request (Login request)
Check user by
Java Servlet
[Fail] HTTP response (authentication page)
Iteration
*getSensor
[Success] HTTP response (control page)
1. Build graphic HMI by Java Applet.
2. Build socket-communication.
3. Acquire and display system status via
socket iteratively.
4. Run supervisory agent (enable/disable
control buttons).
getSensor
return(data)
return
Request via socket (action command)
setActuator
setActuator

Fig. 4. Interactive modeling with sequence diagram for the remote supervisory control system.

1) *Explicit specifications for control sequences:* Generally, these specifications are “recipe-dependent”. They are enforced by a sequence controller in automatic mode or by a human operator in manual mode so as to accomplish certain tasks in a desired logical order.
2) *Implicit specifications for safety requirements:* These specifications are “recipe-independent” and, thus, must always be obeyed throughout operation of the system. Basically, these specifications are required to satisfy safety and liveness constraints. The safety specification prevents the system from performing undesirable actions, while the liveness specification ensures that a given behavior is repeatable. In automatic mode, these specifications can be effectively dealt with by the sequence controller. In manual mode, the supervisor enforces these specifications by restricting the commands available to human operators.

C. PN-Based Design for the Supervisor

PNs have been used to model, analyze, and synthesize control laws for DES. Zhou and DiCesare [21], moreover, addressing the shared resource problem recognized that mutual exclusion theory plays a key role in synthesizing a bounded, live, and reversible PN. In mutual exclusion theory, parallel mutual exclusion consists of a place marked initially with one token to model a single shared resource, and a set of pairs of transitions. Each pair of transitions models a unique operation which requires the use of the shared resource.

In this paper, we adopt mutual exclusion theory [21] to build the PN specification model and then compose it with the

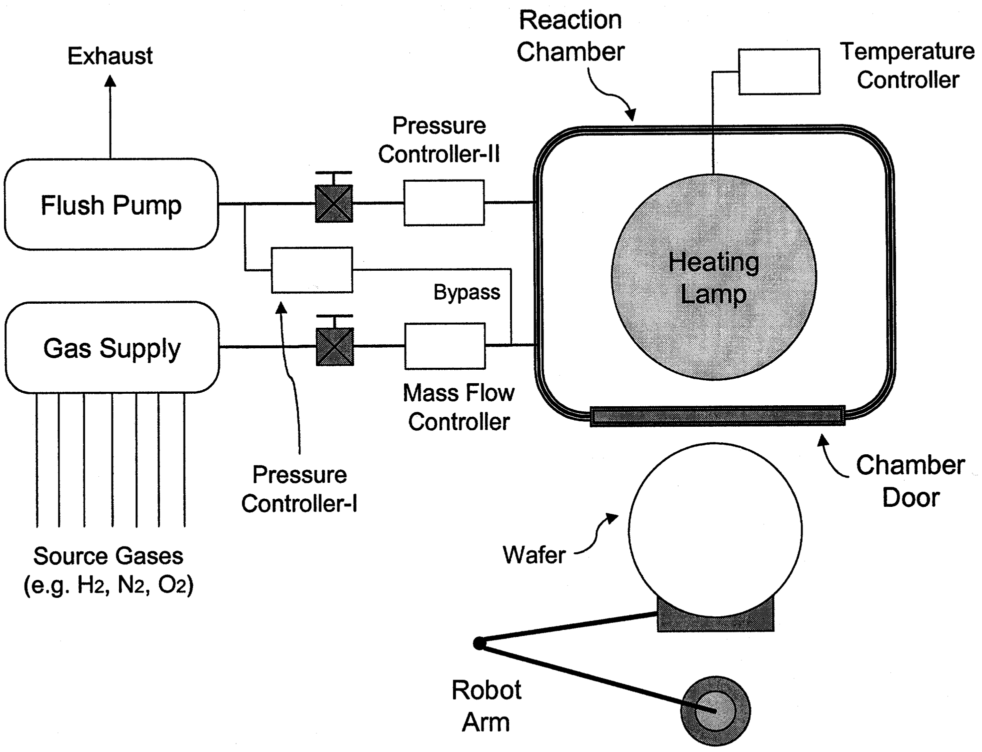


Fig. 5. Schematic diagram of the RTP system.

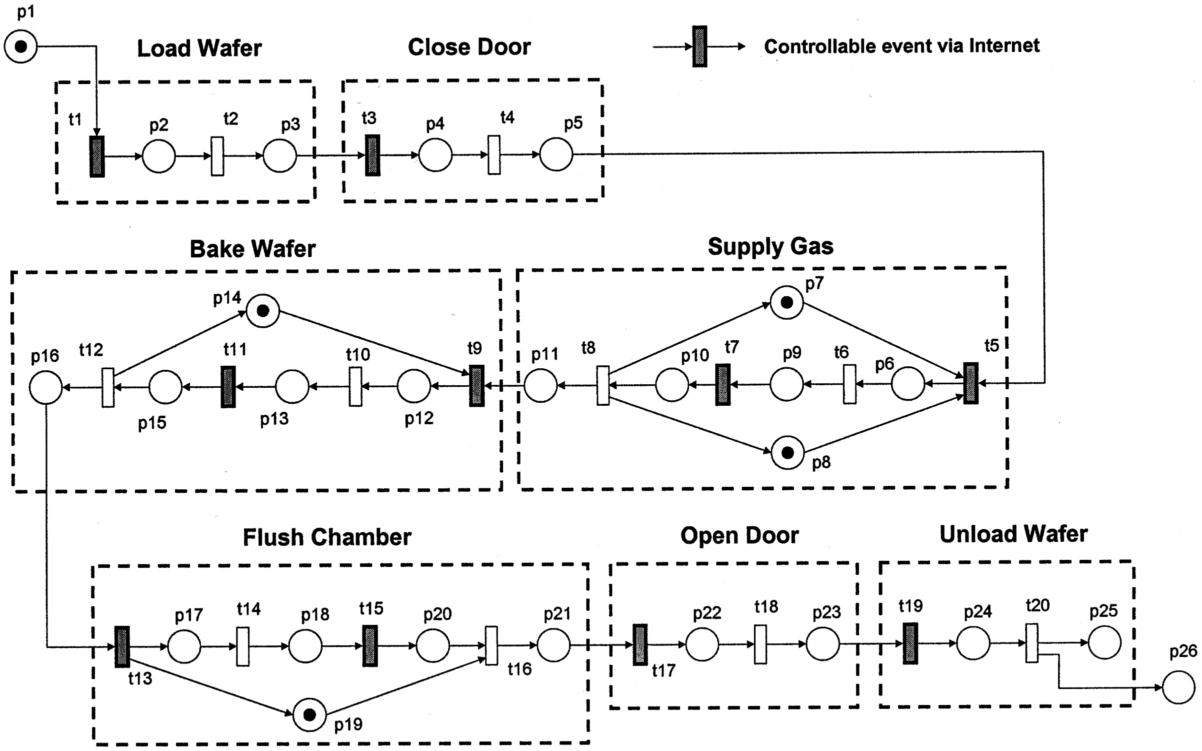


Fig. 6. PN model for automatic control of the RTP system.

plant model to design the supervisor. Moreover, the PN plant model is constructed using the task-oriented concept. Each operation is modeled as a task with a start transition, an end transition, a progressive place and a completed place. Note that the start transition is a controllable event as "command" input, while the end transition is an uncontrollable event as "response" output. The supervisor design procedure consists of the following steps.

Step 1) Construct the PN model of the plant using the task-oriented approach.

Step 2) Construct the PN model of the specifications using the mutual exclusion concept.

TABLE I
NOTATIONS FOR THE PETRI NET OF THE RTP SYSTEM IN FIG. 6

Place	Description	Transition	Description
p1	Raw wafer buffer	t1	Cmd: start loading wafer
p2	Loading wafer	t2	Re: end loading wafer
p3	Loading wafer completed	t3	Cmd: start closing chamber door
p4	Closing chamber door	t4	Re: end closing chamber door
p5	Closing chamber door completed	t5	Cmd: start opening gas valve
p6	Opening gas valve	t6	Re: end opening gas valve
p7	Mass flow controller ready	t7	Cmd: start closing gas valve
p8	Pressure controller-I ready	t8	Re: end closing gas valve
p9	Opening gas valve completed	t9	Cmd: start turning on heating lamp
p10	Closing gas valve	t10	Re: end turning on heating lamp
p11	Closing gas valve completed	t11	Cmd: start turning off heating lamp
p12	Turning on heating lamp	t12	Re: end turning off heating lamp
p13	Turning on heating lamp completed	t13	Cmd: start turning on flush pump
p14	Temperature controller ready	t14	Re: end turning on flush pump
p15	Turning off heating lamp	t15	Cmd: start turning off flush pump
p16	Turning off heating lamp completed	t16	Re: end turning off flush pump
p17	Turning on flush pump	t17	Cmd: start opening chamber door
p18	Turning on flush pump completed	t18	Re: end opening chamber door
p19	Pressure controller-II ready	t19	Cmd: start unloading wafer
p20	Turning off flush pump	t20	Re: end unloading wafer
p21	Turning off flush pump completed		
p22	Opening chamber door		
p23	Opening chamber door completed		
p24	Unloading wafer		
p25	Unloading wafer completed		
p26	Processed wafer buffer		

Step 3) Compose the plant and specification models to yield the supervisor model.

Step 4) Verify and refine the supervisor model to obtain a live, bounded, and reversible model.

D. Example: A Door–Valve System

Consider a simple example of the interaction for the chamber door-gas valve system in an RTP. The general PN model, shown in Fig. 2(a), can be used to describe the open/close tasks for both the door and valve. The initial states of the door and valve are both closed. Assume that one basic safety specification is that the door and valve must not be open at the same time. A PN model for this specification constructed using the mutual exclusion concept is shown in Fig. 2(b). In this model, the start_open_door and start_open_valve commands are mutually exclusive. Intuitively, performance of the start_open_valve command is only allowed if the door is closed and the start_open_door event has not yet been fired. If the start_open_door command has been fired, the start_open_valve command cannot be executed until the end_close_door response is given to signal that the door has been closed. The composed PN model of the door–valve system with the safety specification is shown in Fig. 2(c). The supervisor arcs are shown with dashed lines and the place **ps** showing the supervisor position is drawn thicker than those showing the task positions. In this approach, the supervisor consists only of places and arcs, and its size is proportional to the number of specifications that must be satisfied.

III. Implementation of Supervisors Using Java

This section describes the implementation architecture and interactive modeling of the remote supervisory control system.

A. Client/Server Architecture

Fig. 3 shows the client/server architecture for implementing the remote supervisory control system. On the remote client, the human operator uses a Java-capable Web browser, such as Netscape Navigator or Microsoft Internet Explorer, to connect to the Web server through the Internet. On the Web server side, a Java servlet handles user authentication, while a Java applet provides a graphical human/machine interface (HMI) and invokes the supervisory agent. In this paper, we use Java technology to implement the supervisory agent on an industrial PLC, with a built-in Java-capable Web server assigned to handle the client requests.

B. Interactive Modeling

A sequence diagram of the unified modeling language (UML) [22] is applied to model client/server interaction in the remote control system. Within a sequence diagram, an object is shown as a box at the top of a vertical dashed line, called the object's lifeline and representing the life of the object during the interaction. Messages are represented by horizontal arrows and are drawn chronologically from the top of the diagram to the bottom.

Fig. 4 shows the sequence diagram of the implemented remote supervisory control system. At the first stage, the *Remote*

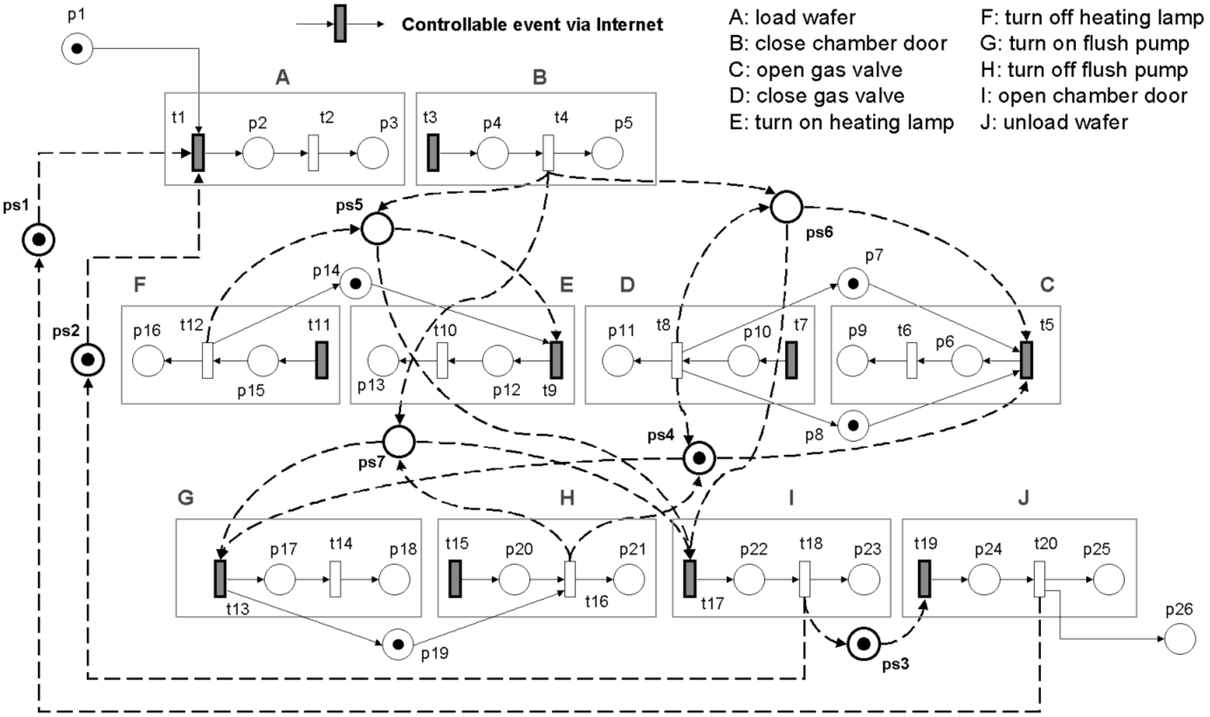


Fig. 7. Composed PN model for manual control of the RTP system.

Client sends a hypertext transfer protocol (HTTP) request to the *Web Server*. Next, the *Web Server* sends an HTTP response with an authentication web page, on which the *Remote Client* can login to the system by sending a request with user/password. The *Web Server* then invokes a Java servlet to authenticate the user. If the authentication fails, the Java servlet will respond with the authentication Web page again. On the other hand, if the authentication succeeds, the Java servlet's response will be a control Web page with a Java applet. The Java applet first builds a graphical HMI and constructs a socket on the specified port to maintain continuous communication with the server. Then, the Java applet acquires the system status through the constructed socket and displays it on the control Web page iteratively by invoking the *Device Handler* to fetch the sensor states of *Device* objects. Finally, the supervisory agent is called by the Java applet and run to enable/disable associated control buttons on the HMI according to the current system status so as to meet the required specifications. Thus, the *Remote Client* can send an action command by pushing an enabled button to control the remote system through the constructed socket.

IV. Remote Monitoring and Control of an RTP

This section demonstrates a practical application of the remote monitoring and supervisory control to an RTP via the Internet.

A. Description of the RTP System

An RTP is a relatively new semiconductor manufacturing device [23]. A schematic diagram of the RTP system is shown in Fig. 5, which is composed of: 1) a reaction chamber with a door; 2) a robot arm for wafer loading/unloading; 3) a gas supply module with a mass flow controller and pressure controller-I; 4) a heating lamp module with a temperature controller; and 5) a flush pumping system with a pressure controller-II.

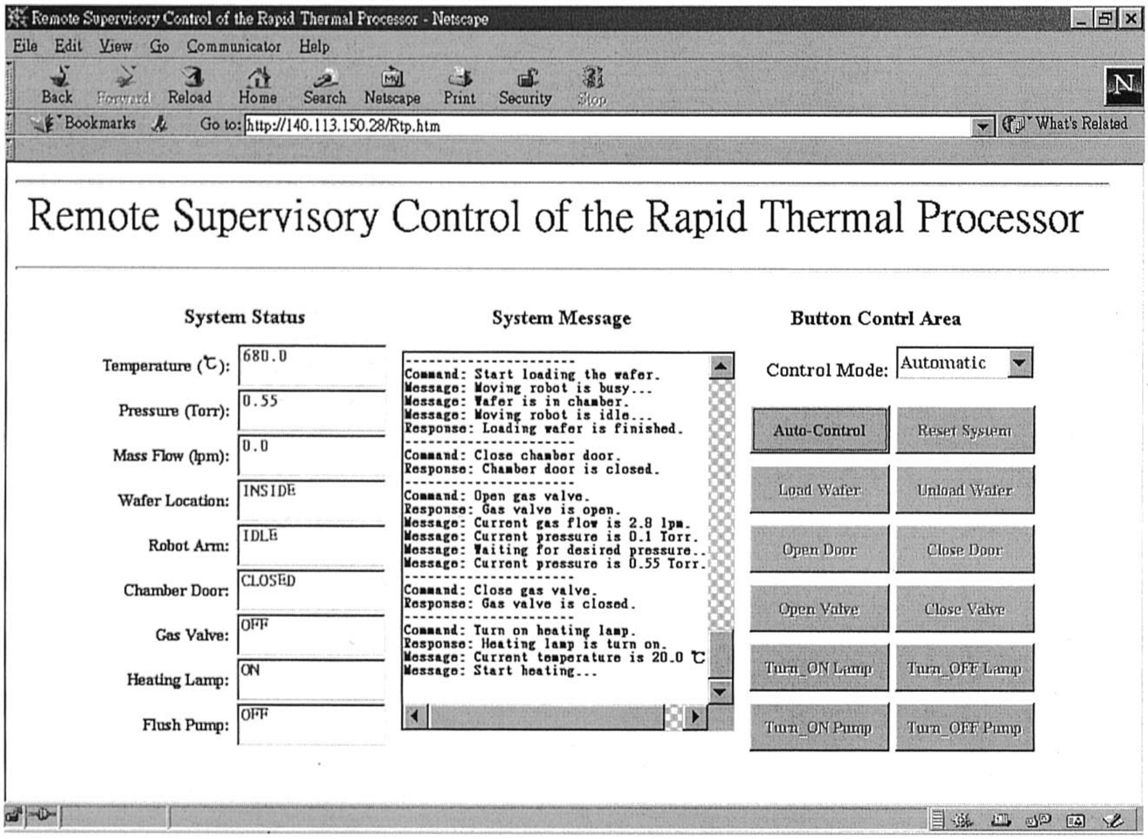
TABLE II
Notations for the Supervisory Places of the Petri Net in Fig. 7

Place	Description
ps1	Spec-1: chamber is empty
ps2	Spec-2: chamber door is open
ps3	Spec-2: chamber door is open
ps4	Spec-3: gas is closed/pump is off
ps5	Spec-4: door is closed/lamp is off
ps6	Spec-4: door is closed/gas is closed
ps7	Spec-4: door is closed/pump is off

A realistic "recipe" of the hydrogen bake process, i.e., the explicit specification as mentioned in Section II-B, is as follows.

Step 1) Load the raw wafer.
Step 2) Close the chamber door.
Step 3) Open the gas valve to supply gases with a desired gas flow rate and pressure of 2.8L/min and 0.5 torr, respectively.
Step 4) Close the gas valve.
Step 5) Turn on the heating lamp to bake the wafer with a desired baking temperature and duration of 1000 °C and 4 s, respectively.
Step 6) Turn off the heating lamp.
Step 7) Turn on the flush pump with a desired pressure of less than 0.05 torr.
Step 8) Turn off the flush pump.
Step 9) Open the chamber door.
Step 10) Unload the processed wafer.

The initial state of the components in the RTP is either closed or off, except that the door is open. The following safety


Remote Supervisory Control of the Rapid Thermal Processor - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Go to: http://140.113.150.28/Rtp.htm
What's Related
Auto-Control
Reset System
Load Wafer
Unload Wafer
Open Door
Close Door
Open Valve
Close Valve
Turn_ON Lamp
Turn_OFF Lamp
Turn_ON Pump
Turn_OFF Pump

Remote Supervisory Control of the Rapid Thermal Processor

System Status

Temperature (℃): 680.0
Pressure (Torr): 0.55
Mass Flow (lpm): 0.0
Wafer Location: INSIDE
Robot Arm: IDLE
Chamber Door: CLOSED
Gas Valve: OFF
Heating Lamp: ON
Flush Pump: OFF

System Message

```
---------------------
Command: Start loading the wafer.
Message: Moving robot is busy...
Message: Wafer is in chamber.
Message: Moving robot is idle...
Response: Loading wafer is finished.
---------------------
Command: Close chamber door.
Response: Chamber door is closed.
---------------------
Command: Open gas valve.
Response: Gas valve is open.
Message: Current gas flow is 2.8 lpm.
Message: Current pressure is 0.1 Torr.
Message: Waiting for desired pressure..
Message: Current pressure is 0.55 Torr.
---------------------
Command: Close gas valve.
Response: Gas valve is closed.
---------------------
Command: Turn on heating lamp.
Response: Heating lamp is turn on.
Message: Current temperature is 20.0 ℃
Message: Start heating...
```

Button Contrl Area

Control Mode: Automatic

Fig. 8. Interactive web page for remote control of the RTP system by a Java applet (only **Auto-Control** button is admissible in the automatic control mode).

specifications, i.e., the implicit specification mentioned in Section II-B, must be enforced throughout system operation.

Spec-1: Wafer loading is allowed only when no wafer is in the chamber.

Spec-2: Wafer loading/unloading is allowed only when the door is open.

Spec-3: The gas valve must be closed when the flush pump is applied to the chamber.

Spec-4: The gas valve, heating lamp, and flush pump cannot be started when the door is open.

B. Sequence Controller Design for Automatic Control Mode

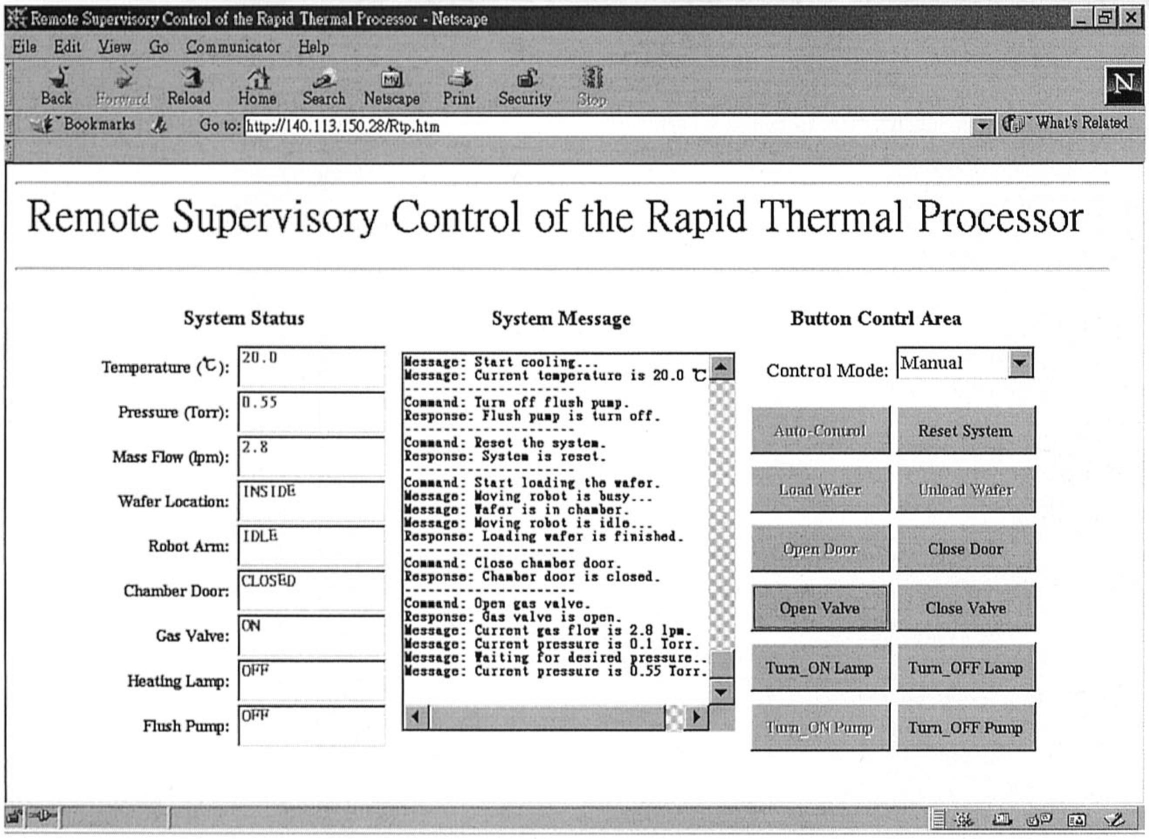
As mentioned in Section II-B, the specifications can be satisfied and involved in the sequence controller in automatic control mode. By applying the task-oriented concept, the PN model for the automatic control mode of the RTP is constructed as shown in Fig. 6, which consists of 26 places and 20 transitions, respectively. Corresponding notations are described in Table I. Transitions drawn with dark symbols are events that are controllable by remote clients via the Internet.

C. Design of the Supervisor for Manual Control Mode

For manual control mode, the plant model is formed by unconnecting each pair of transitions for the tasks in Fig. 6. In the specification model, Spec-1 and Spec-2 are modeled as the pre-conditions of the associated operations, while Spec-3 and Spec-4 are built by using the mutual exclusion concept. The composed PN model of both the plant and specifications is shown in Fig. 7, where A–J represent ten remote-controllable tasks for the RTP system. The supervisory places **ps1-7** (**ps1** for Spec-1, **ps2-3** for Spec-2, **ps4** for Spec-3, **ps5-7** for Spec-4) are used to prevent undesired and unsafe operations on the part of the human operator. Corresponding notations for the supervisory places are described in Table II. At this stage, the software package *PETRI Maker* [24] is chosen to verify the behavioral properties of the composed PN model due to its graphical representation, ease of manipulation, and ability to perform structural and performance analyses. Validation results reveal that the present PN model is live and bounded. The liveness property means that the system can be executed properly without deadlocks, while the boundedness property means that the system can be executed with limited resources (e.g., limited buffer sizes).

D. Implementation With Java Technology

The system modeling and design developed in previous stages provide supervisory control models for implementation of the present remote monitoring and control technology. To implement the supervisory control, we use Java due to its object orientation, portability, safety, and built-in support for networking and concurrency [1], [25], [26]. The developed supervisory agent is implemented on the Mirle SoftPLC (80486-100 CPU), an advanced industrial PLC with built-in web server and Java virtual machine so that it can process both HTTP requests and Java programs [19], [20].


Remote Supervisory Control of the Rapid Thermal Processor - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks
Go to: http://140.113.150.28/Rtp.htm
What's Related
Auto-Control
Reset System
Load Wafer
Unload Wafer
Open Door
Close Door
Open Valve
Close Valve
Turn_ON Lamp
Turn_OFF Lamp
Turn_ON Pump
Turn_OFF Pump

Remote Supervisory Control of the Rapid Thermal Processor

System Status

Temperature (℃): 20.0

Pressure (Torr): 0.55

Mass Flow (lpm): 2.8

Wafer Location: INSIDE

Robot Arm: IDLE

Chamber Door: CLOSED

Gas Valve: ON

Heating Lamp: OFF

Flush Pump: OFF

System Message

```
Message: Start cooling...
Message: Current temperature is 20.0 ℃
----------------------
Command: Turn off flush pump.
Response: Flush pump is turn off.
----------------------
Command: Reset the system.
Response: System is reset.
----------------------
Command: Start loading the wafer.
Message: Moving robot is busy...
Message: Wafer is in chamber.
Message: Moving robot is idle...
Response: Loading wafer is finished.
----------------------
Command: Close chamber door.
Response: Chamber door is closed.
----------------------
Command: Open gas valve.
Response: Gas valve is open.
Message: Current gas flow is 2.8 lpm.
Message: Current pressure is 0.1 Torr.
Message: Waiting for desired pressure..
Message: Current pressure is 0.55 Torr.
```

Button Contrl Area

Control Mode: Manual

Fig. 9. Interactive Web page in manual control mode at Step 3 of RTP processing (seven buttons are enabled).

The developed HMI, shown in Fig. 8, is carefully designed to make its web pages more user friendly and also to increase download speed by avoiding unnecessary images. Since the client users will be mainly operators and engineers, they will want effective information delivery and will not be interested in flashy graphics [27]. The current system status is placed on the left, the system message is in the center, and the button control area is on the right. Fig. 8 also shows that the system is in automatic control mode and, thus, only the **Auto-Control** button has been enabled by the supervisory agent. The human operator can only push this button which starts automatic process control by the sequence controller.

Fig. 9 shows the Web pages for manual control mode after the **Open Valve** button has just been pushed (Step 3 in Section IV-A). In this situation, since one wafer is already in the chamber and the door is closed, the **Load Wafer** and **Unload Wafer** buttons are both disabled by the supervisory agent to meet Spec-1 and Spec-2. Moreover, the **Turn_On Pump** and **Open Door** buttons are disabled to meet Spec-3 and Spec-4, respectively. Thus, the safety requirements of the RTP processing are guaranteed as human operations are conducted. Fig. 10 shows the hardware setup during prototype development.

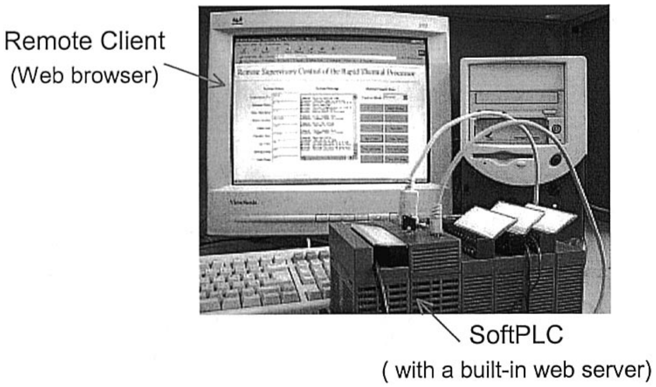


Fig. 10. Hardware setup during prototype development.

V. Conclusion

This paper has presented a framework for designing and implementing a PN-based supervisor for Internet-based control systems with the human in the loop. The supervisor is systematically designed by applying the mutual exclusion concept and is then implemented using Java technology. To demonstrate the practicability of the proposed remote supervisory approach, an application is provided in which an RTP system with an industrial PLC is controlled over the Internet. According to the feedback status of the remotely located system, the developed supervisor provides allowable commands for human operators while disabling operations which violate safety specifications.

Since the original supervisory control framework [9]–[11] is restricted to purely logical system models, for applications with time-based requirements, it is necessary to extend the present models with time specifications. Moreover, new specifications for the multiple-operator environment and the recovery of partially completed sessions due to the unreliable HTTP communication channels should be investigated in the future.

REFERENCES

[1] A. Weaver, J. Luo, and X. Zhang, "Monitoring and control using the Internet and Java," in *Proc. IEEE IECON'99*, San Jose, CA, 1999, pp. 1152–1158.

[2] R. L. Kress, W. R. Hamel, P. Murray, and K. Bills, "Control strategies for teleoperated Internet assembly," *IEEE/ASME Trans. Mechatron.*, vol. 6, pp. 410–416, Dec. 2001.

[3] G. Q. Huang and K. L. Mak, "Web-integrated manufacturing: Recent developments and emerging issues," *Int. J. Comput. Integrated Manuf.*, vol. 14, no. 1, pp. 3–13, 2001.

[4] J. S. Lee and P. L. Hsu, "Design of remote environmental monitoring systems," in *Proc. IEEE Int. Conf. Control Applications*, Glasgow, U.K., 2002, pp. 856–861.

[5] R. Oboe, "Web-interfaced, force-reflecting teleoperation systems," *IEEE Trans. Ind. Electron.*, vol. 48, pp. 1257–1265, Dec. 2001.

[6] C. Batur, Q. Ma, K. Larson, and N. Kettenbauer, "Remote tuning of a PID position controller via Internet," in *Proc. American Control Conf.*, 2000, pp. 4403–4406.

[7] H. McBreen and M. Jack, "Evaluating humanoid synthetic agents in e-retail applications," *IEEE Trans Syst., Man, Cybern. A*, vol. 31, pp. 394–405, Sept. 2001.

[8] J. Rasmussen, A. M. Pejtersen, and L. P. Goodstein, *Cognitive Systems Engineering*. New York: Wiley, 1994.

[9] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM J. Control Optim.*, vol. 25, no. 1, pp. 206–230, 1987.

[10] ——, "The control of discrete event systems," *Proc. IEEE*, vol. 77, pp. 81–98, Jan. 1989.

[11] S. Balemi, G. J. Hoffmann, P. Gyugyi, H. Wong-Toi, and G. F. Franklin, "Supervisory control of a rapid thermal multiprocessor," *IEEE Trans. Automat. Contr.*, vol. 38, pp. 1040–1059, July 1993.

[12] J. O. Moody and P. J. Antsaklis, *Supervisory Control of Discrete Event Systems Using Petri Nets*. Boston, MA: Kluwer, 1998.

[13] A. Giua and F. DiCesare, "Supervisory design using Petri nets," in *Proc IEEE Int. Conf. Decision and Control*, Brighton, U.K., 1991, pp. 92–97.

[14] R. Zurawski and M.C. Zhou, "Petri nets and industrial applications: a tutorial," *IEEE Trans. Ind. Electron.*, vol. 41, pp. 567–583, Dec. 1994.

[15] S. Srinivasan, "Modeling and performance analysis of cluster tools using Petri nets," *IEEE Trans. Semiconduct. Manufact.*, vol. 11, pp. 394–403, Aug. 1998.

[16] M. Uzam, A. H. Jones, and I. Yücel, "Using a Petri-net-based approach for the real-time supervisory control of an experimental manufacturing system," *Int. J. Adv. Manuf. Technol.*, vol. 16, no. 7, pp. 498–515, 2000.

[17] J. S. Lee and P. L. Hsu, "A PLC-based design for the sequence controller in discrete event systems," in *Proc. IEEE Int. Conf. Control Applications*, Anchorage, AK, 2000, pp. 929–934.

[18] ——, "A new approach to evaluate ladder logic diagrams and Petri nets via the IF-THEN transformation," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, Tucson, AZ, 2001, pp. 2711–2716.

[19] *SoftPLC Controller User's Manual Version 1.2.*, Mirle Automation Corp., Hsinchu, Taiwan, R.O.C., 1999.

[20] *SoftPLC-Java Programmer's Toolkit*, SoftPLC Corp., Spicewood, TX, 1999.

[21] M. C. Zhou and F. DiCesare, "Parallel and sequential mutual exclusions for Petri net modeling for manufacturing systems," *IEEE Trans. Robot. Automat.*, vol. 7, pp. 515–527, Aug. 1991.

[22] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*. Reading, MA: Addison-Wesley, 1999.

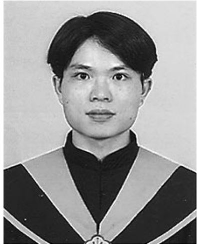
[23] R. B. Fair, *Rapid Thermal Processing: Science and Technology*. New York: Academic, 1993.

[24] A. Godon and F. Bousseau, *High-Level PN Simulator: PETRI Maker 3.1*. Angers, France: ISTIA, Univ. Angers, 1996.

[25] T. Hoshi, "Current and future Java technology for manufacturing industry," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, Tokyo, Japan, Oct. 1999, pp. 404–409.

[26] E. Bertolissi and C. Preece, "Java in real-time applications," *IEEE Trans. Nucl. Sci.*, vol. 45, pp. 1965–1972, Aug. 1998.

[27] P. Shikli, "Designing winning web sites for engineers," *Mach. Des.*, vol. 69, no. 21, pp. 30–40, 1997.



Jin-Shyan Lee received the B.S. degree in mechanical engineering in 1997 from National Taiwan University of Science and Technology, Taipei, Taiwan, R.O.C., and the M.S. degree in electrical and control engineering in 1999 from National Chiao Tung University, Hsinchu, Taiwan, R.O.C., where he is currently working toward the Ph.D. degree in electrical and control engineering.

His current research interests include discrete-event systems, supervisory control, hybrid systems, Internet-based monitoring and control, factory automation, and intelligent transportation systems.

Pau-Lo Hsu (M'92) received the B.S. degree from National Cheng Kung University, Tainan, Taiwan, R.O.C., in 1978, the M.S. degree from the University of Delaware, Newark, in 1984, and the Ph.D. degree from the University of Wisconsin, Madison, in 1987, all in mechanical engineering.

Following two years of military service in King-Men, he was with San-Yang (Honda) Industry during 1980–1981 and Sandvik (Taiwan) during 1981–1982. In 1988, he joined the Department of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu, Taiwan, R.O.C., as an Associate Professor. He became a Professor in 1995. During 1998–2000, he served as the Chairman of the Department of Electrical and Control Engineering. His research interests include mechatronics, CNC motion control, servo systems, network-based control systems, and diagnostic systems.