Development of a Remote Management System for Automatic Parking Towers through Mobile Devices

Jin-Shyan Lee and Ta-Cheng Chien
Department of Electrical Engineering
National Taipei University of Technology (Taipei Tech.)
Taipei, Taiwan
jslee@mail.ntut.edu.tw and t101318053@ntut.edu.tw

Yuan-Heng Sun
Information & Communications Research Labs
Industrial Technology Research Institute (ITRI)
Hsinchu, Taiwan
gilbertsun@itri.org.tw

Abstract—**Vehicle parking plays a key role in our modern life. Currently, automatic parking towers have attracted much research due to its high space-efficiency. Remote management of a parking tower provides the functions for human to perform monitoring and control via a specific network. From a system point of view, an automatic parking tower is inherently a discrete event system. For such systems, this paper has realized a Java-based management system to provide cross-platform remote access via various devices. In the present approach, Colored Petri Nets (CPNs) are used to model the operated behaviors so as to result in a more compact structure. A prototype of an eight-space parking tower is designed and implemented to show the feasibility of the developed approach. It is believed that the technique presented in this paper could be further applied to large-scale parking systems.**

Keywords—remote management systems; mobile devices; Colored Petri Nets (CPNs); automatic parking towers.

I. Introduction

Recently, there has been an increasing emphasis on developing consumer electronics, which combines in a synergistic way the classical engineering disciplines of mechanical and electrical engineering and computer science, leading to new kinds of research. In our modern days, car parking is a very crucial issue, and one of the topics in consumer electronics is the investigation of applying the electrical, automation, information, and communication technologies to deal with this problem. In particular, remote management of an automatic parking tower provides the functions for human to perform monitoring and control over a great distance. In real applications, a human operator may use the remote monitoring and control functions to further investigate the parking tower conditions if an alert is launched, or to maintain and repair the parking system if required. However, most of the remote management literature focuses on using a regular PC as the client to access the system. Due to the rapid evolution of mobile phones and tablet PC, the human operator would like to use these modern devices via different web browsers and operation systems to access the server platform at anytime from anywhere [1]-[4]. For such system requirements, this paper has realized a Java-based management system to provide cross-platform remote access via the smart phone, tablet PC, or regular PC.

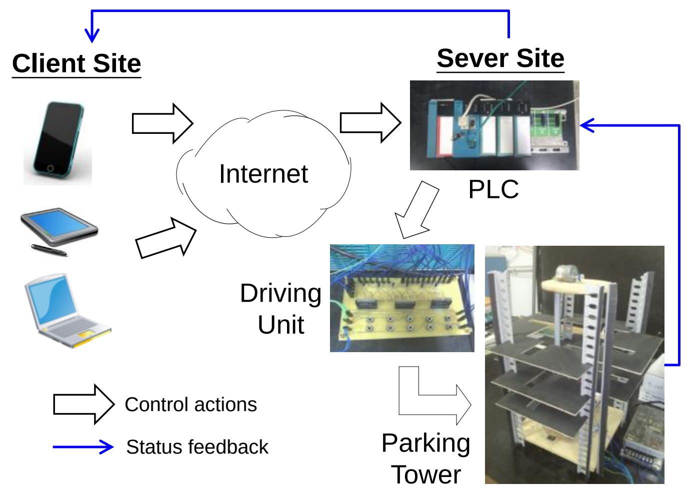

Figure 1. Prototype realization of a parking tower for remote monitoring and control via various devices.

On the other hand, a car parking system is inherently a Discrete Event System (DES), that is, a dynamic system with state changes driven by occurrences of individual events [5]. One way of modeling the DES is using the Petri Nets (PNs) [6]-[8], which have been applied in manufacturing, and more specifically in factory automation, for many years. Their major advantage is the evaluation of all system states before implementation. However, as the number of specifications increases, even simple PN models soon tend to become highly complex. Thus, the use of high-level Petri nets, i.e., Colored PNs (CPNs) has been proposed, leading to a more compact model [9]-[10]. For example, Dotoli and Fanti [11] proposed a Colored Timed Petri Net (CTPN) model to describe in a concise and efficient way the dynamics of an Automated Storage and Retrieval System (AS/RS) serviced by Rail Guided Vehicles (RGVs). Also, Lee and Lee [12] presented a CPN-based modeling and control framework for remotely operated conveyor systems. In their proposed approach, system behaviors were modeling based on the CPN so as to deal with the modeling complexity in large-scale systems with similar behaviors.

Based on the CPN modeling technique, this paper has implemented a prototype of an 8-space parking tower which is automatically controlled by a Programmable Logic Controller (PLC). In our proposed remote management scheme, as shown

 ISBN: 978-1-61208-368-1

in Figure 1, the human operator uses the client devices (smart phone, tablet PC, or regular PC) to send control commands to the PLC-based sever through the internet. The commands could be decided according to the status feedback from the server site. Then, the PLC actuates the controlled components in the parking tower via the driving unit. The responses with the status will be fed back to the client site and hence the control loop is closed in this way.

The rest of this paper is organized as follows. Section II briefly introduces the CPN-based modeling scheme. Next, a PLC-based implementation of the remote management system is described in Section III. Then, in Section IV, an example of an eight-space parking tower is illustrated to show the feasibility. Finally, Section V concludes this paper.

II. MODELING VIA COLORED PETRI NETS

A. Ordinary Petri Nets

An ordinary PN is identified as a particular kind of bipartite directed graph populated by three types of objects. They are places, transitions, and directed arcs connecting places and transitions. Formally, an ordinary PN is defined as

$$PN = (P, T, I, O, M) \tag{1}$$

where,

$P = \{p_1, p_2, \ldots, p_m\}$ is a finite set of places, where $m > 0$;

$T = \{t_1, t_2, \ldots t_n\}$ is a finite set of transitions, where $n > 0$;

$I : P \times T \rightarrow N$ is an input function that defines a set of directed arcs from P to T, where $N = \{0, 1, 2, \ldots\}$;

$O : T \times P \rightarrow N$ is an output function that defines a set of directed arcs from T to P;

$M : P \rightarrow N$ is a marking. An initial marking is denoted by M_0.

A transition t is enabled if each input place p of t contains at least the number of tokens equal to the weight of the directed arc connecting p to t. When an enabled transition fires, it removes the tokens from its input places and deposits them on its output places. PN models are suitable to represent the systems that exhibit concurrency, conflict, and synchronization. Some important PN properties include boundness (no capacity overflow), liveness (freedom from deadlock), conservativeness (conservation of non-consumable resources), and reversibility (cyclic behavior). The concept of liveness is closely related to the complete absence of deadlocks. Validation methods of these properties include reachability analysis, invariant analysis, reduction method, siphons/traps-based approach, and simulation.

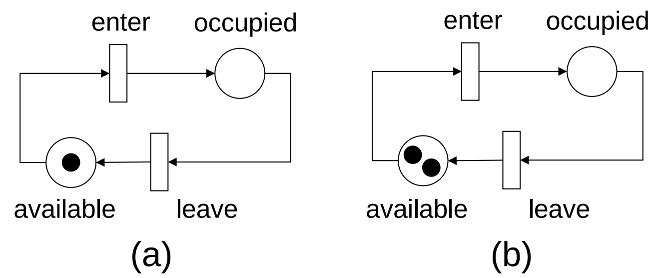

Figure 2. Ordinary Petri net model of the (a) one-space, and (b) two-space parking areas.

Figure 2 (a) shows a simple example of the ordinary PN model for car entering and leaving one parking space in the automatic parking tower. The initial state of parking space is available. Assume a parking tower has *k* parking space, it could be modeled by *k* tokens in the ordinary PN, as shown in Figure 2 (b), where *k*=2. However, using this modeling technique for multiple parking spaces, only the question of "how many" spaces available could be answered, rather than "which" parking space is available. Hence, in order to indicate "which" parking space is available, a more complex PN model, as shown in Figure 3, could be applied. The parking area is organized into two parking spaces, of which the Space-1 is occupied and the Space-2 is available, respectively. Obviously, the model becomes much more complicated as the number of parking spaces increases. Hence, the high-level Petri nets, i.e., CPNs are applied in our work, resulting in a more compact model.

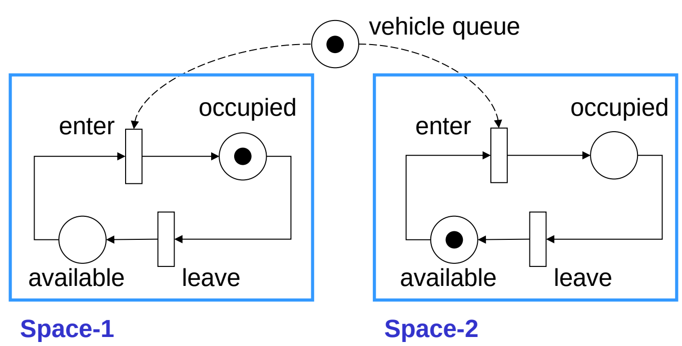

Figure 3. Ordinary Petri net model of a two-space parking area.

B. Colored Petri Nets

A colored PN comprises tokens to which colors are attributed [9]. CPN forms a category of nets whose intuitive perception is less clear than the ordinary PN, but has great value for the modeling of certain complex systems. Formally, a colored PN is defined as

$$CPN = (P, T, C, I, O, M) \tag{2}$$

where,

$P = \{p_1, p_2, \ldots, p_m\}$ is a finite set of places, where $m > 0$;

$T = \{t_1, t_2, \ldots t_n\}$ is a finite set of transitions, where $n > 0$;

C is the color-function; $C(p)$ and $C(t)$ denote the sets of colors associated with place $p \in P$ and transition $t \in T$.

$I(p,t): C(p) \times C(t) \to N$ is an input function that defines a set of directed arcs from *p* to *t*, where $N = \{0, 1, 2, \ldots\}$;

$O(t,p): C(t) \times C(p) \to N$ is an output function that defines a set of directed arcs from *t* to *p*;

$M: C(p) \to N$ is a marking.

Obviously, the CPN is the extension of an ordinary PN. In a CPN, there is a set of colors associated with each place and transition, and a transition can fire with respect to each of its colors. Considering the previous example of the two-space parking tower, the CPN model can be designed as shown in Figure 4 with the elements as follows:

$$P = \{\text{available, occupied}\}$$
$$T = \{\text{enter, leave}\}$$
$$C(\text{available}) = C(\text{occupied}) = \text{SPACE},$$
$$\text{where SPACE} = \{s1, s2\}$$
$$I(p,t) = O(t,p) = ID$$
$$M_0(\text{available}) = s2, M_0(\text{occupied}) = s1$$

Note the *ID* means the identity function, which selects all the items of a basic color domain. Here, the *ID* means the set of SPACE.

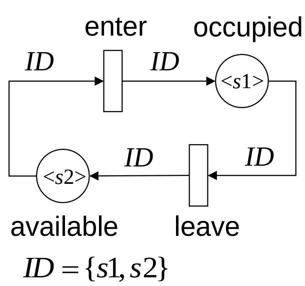

Figure 4. Colored Petri net model of a two-space parking area.

Obviously, the net structure of the CPN (Figure 4) is greatly simplified as compared with the previous ordinary PN model (Figure 3).

III. PLC-Based Realization

A. Client-Server Architecture

Figure 5 shows the client-server architecture for implementing the remote management system. On the client side, the remote manager uses a Java-capable web browser, such as Internet Explorer or Firefox, to connect to the web server through the internet. On the server side, an industrial PLC with a built-in Java-capable web server assigned to handle the client requests is employed. Within the PLC, a Java servlet handles user authentication, a Java applet provides a graphical Human-Machine Interface (HMI), and a Ladder Logic Diagram (LLD) performs the detailed operations of the requested tasks. Our choice of using LLD to implement the local operations due to its wide use in industry, while using Java to implement the remote functions because of its object-orientation, portability, safety, and built-in support for networking and concurrency.

B. Interactive Modeling

A sequence diagram of the Unified Modeling Language (UML) [13] is applied to model the client-server interaction in the remote management system. As shown in Figure 6, at the first stage, the *Remote Client* sends a HyperText Transfer Protocol (HTTP) request to the *Web Server*. Next, the *Web Server* replies 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. After that, the *Remote Client* can issue an action command from the control page to actuate the remote system through the constructed socket. The responses with the status will be continuously fed back to the *Remote Client* and thus the control loop is closed.

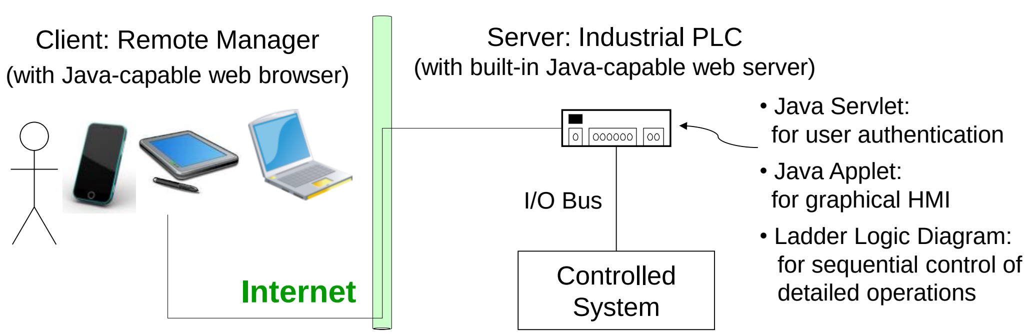

Figure 5. Implementation architecture of the remote management system.

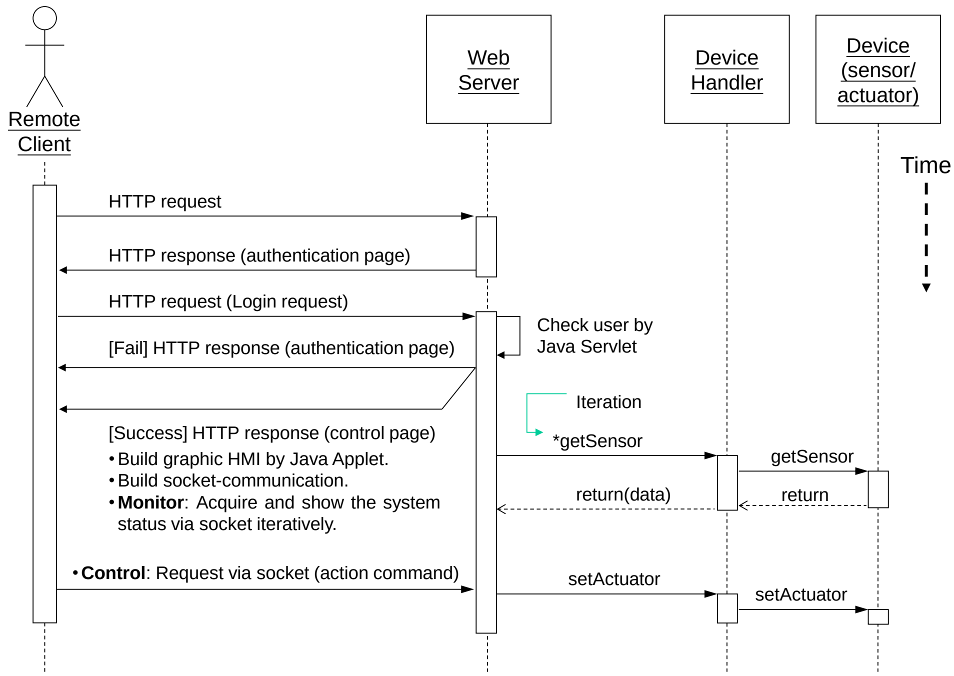

Figure 6. Interactive modeling with sequence diagram for the remote management system.

IV. An Application Prototype

A. System Description

In a parking tower, the area is organized into a lot of parking spaces. For simplicity, the prototype of an eight-space parking tower is designed and implemented as shown in Fig 7. In the developed tower system, three motors (corresponding to three degrees of freedom) and four limited switch sensors are employed to place the vehicle on a desired parking space. Also, ten control buttons on the driving unit are used to provide local control functions.

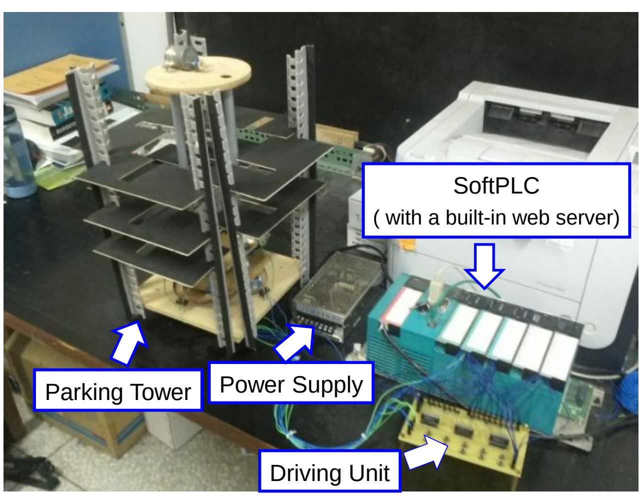

Figure 7. The hardware setup during prototype development.

B. Modeling and Implementation for Local Control

For the operator-issued commands, the CPN model of the eight-space parking system is constructed as shown in Figure 8. The model consists of 4 places, 4 transitions, and 8 color elements (corresponding to eight parking spaces), respectively. Corresponding notation of the PN model is also described in the figure. Also, the related LLD is designed for local control, as shown in Figure 9.

C. Java-Based Realization for Remote Management

To implement the remote monitoring and control functions, we use Java due to its object-orientation, portability, safety, and built-in support for networking and concurrency. The developed server program is located on an advanced PLC (80486-100 CPU) with built-in web server and Java virtual machine so that it can process both HTTP requests and Java programs.

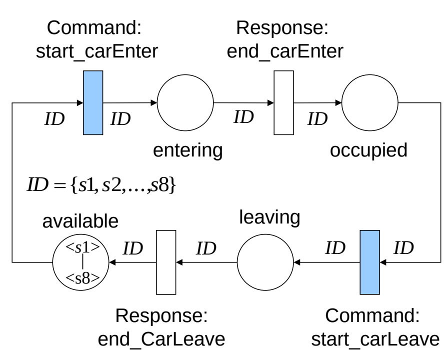

Figure 8. Colored PN model of the eight-space parking system.

 ISBN: 978-1-61208-368-1

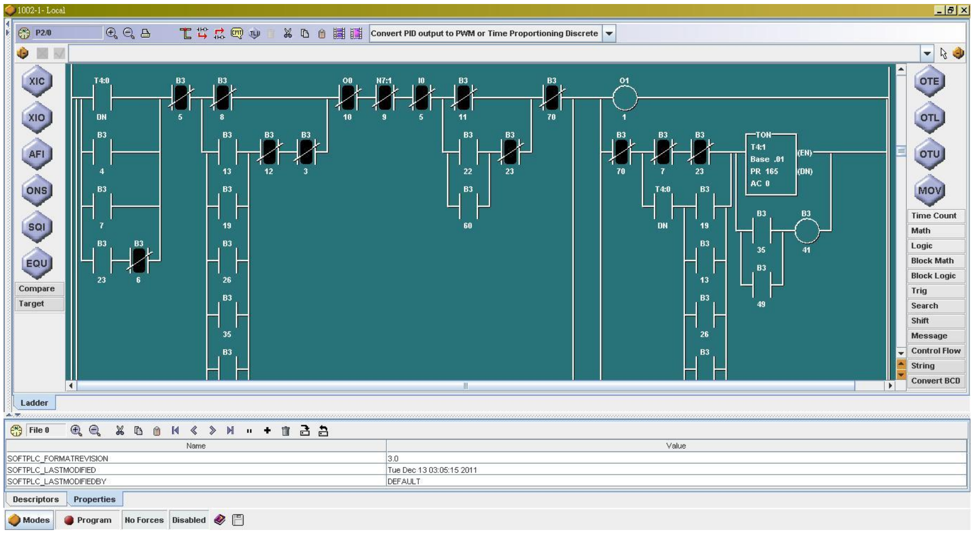

Figure 9. Snapshot of ladder logic diagrams during the PLC implementation.

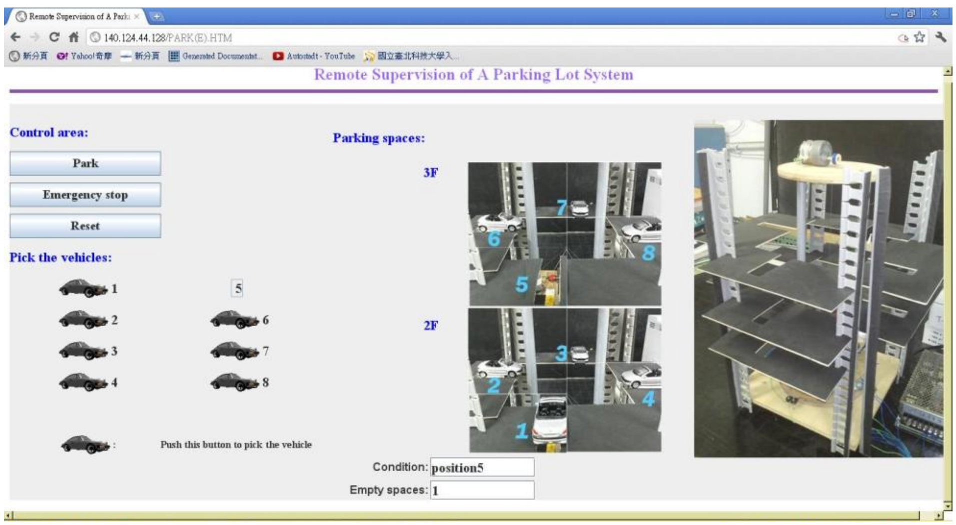

Figure 10. Interactive web page for the remote management of the parking tower.

The developed HMI, shown in Figure 10, is carefully designed to make its web pages more user-friendly. The button control area is placed on the left, and the current status and system message is on the right. The human operator can push the buttons to park a car into a parking space or to take a specific car from the tower.

Figure 11. Remote management of the parking tower via a smart phone.

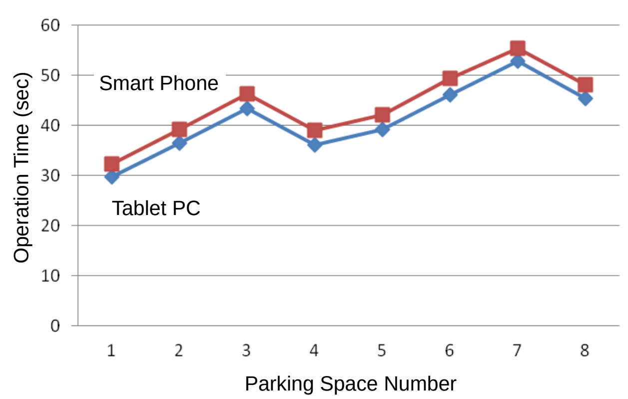

Figure 12. Comparison of using a smart phone and a tablet PC to remotely perform parking (or taking) a car.

Also, the operator can use a smart phone to manage the parking system, as shown in Figure 11. However, due to limitations of computing power and communication bandwidth on smart phones, the consumed time of using the phone to park (or take) a car is a little longer than the time of using a tablet PC. Figure 12 shows the comparison of realistic operation time of parking (or taking) a car between using a smart phone and a tablet PC. Moreover, the operation time of parking (or taking) a car from parking space 7 is the longest, since the space 7 is on the innermost part of the circle as shown in Figure 10. On the other hand, using the space 1 will take the shortest time since it is near the entry.

V. CONCLUSION AND FUTURE WORK

This paper was motivated by the requirement of remotely access for automatic parking systems. For such systems, this paper has realized a Java-based management system to provide cross-platform remote access. Moreover, in order to cope with the complexity and realization issues in large-scale parking systems, a systematic development approach is proposed in this paper. In the present approach, the system behaviors are modeling based on colored Petri nets, and then remote management functions are implemented via a Java-based PLC. To demonstrate the practicability of the proposed approach, an application to the eight-space parking tower is illustrated with realization. It is believed that the presented technique could be further extended to large-scale parking towers. Future work will attempt to improve the access-control policy for multiple operators. Also, the issue on how to optimize given problems of parking systems (e.g., how to route cars to the shortest fitting free parking slot) would be addressed in the future.

ACKNOWLEDGMENT

This paper was supported by the Ministry Of Science and Technology (MOST) of Taiwan under the grant 102-2511-S-027-004, and in part by the Ministry Of Economic Affairs (MOEA), Taiwan. Moreover, the authors would like to thank Y-J. Zhuang and Z-N. Zhang, and L-Y. Chen from the Department of Electrical Engineering, National Taipei University of Technology (Taipei Tech.), for their assistance in the prototype implementation.

REFERENCES

[1] H. S. Ahn, I. K. Sa, and J. Y. Choi, “PDA-based mobile robot system with remote monitoring for home environment,” IEEE Trans. Consumer Electron., vol. 55, no. 3, pp. 1487-1495, Aug. 2009.

[2] G. Paravati, C. Celozzi, A. Sanna, and F. Lamberti, “A feedback-based control technique for interactive live streaming systems to mobile devices,” IEEE Trans. Consumer Electron., vol. 56, no. 1, pp. 190-197, Feb. 2010.

[3] J. S. Lee, “A Petri net design of command filters for semiautonomous mobile sensor networks,” IEEE Trans. Ind. Electron., vol. 55, no. 4, pp. 1835-1841, April 2008.

[4] J. S. Lee and P. L. Hsu, “Implementation of a remote hierarchical supervision system using Petri nets and agent technology,” IEEE Trans. Syst., Man, Cybern. C, Appl. Rev., vol. 37, no. 1, pp. 77-85, Jan. 2007.

[5] B. Hruz and M. C. Zhou. Modeling and Control of Discrete Event Dynamic Systems. Springer: London, 2007.

[6] J. S. Lee, M. C. Zhou, and P. L. Hsu, “An application of Petri nets to supervisory control for human-computer interactive systems,” IEEE Trans. Ind. Electron., vol. 52, no. 5, pp. 1220-1226, Oct. 2005.

[7] H. Zheng, Y. G. Niu, and G. Ciardo, “Modelling and analysis of UPnP AV media player system based on Petri nets,” Int. J. Syst. Sci., vol. 42, no. 9, pp. 1573-1580, Sept. 2011.

[8] J. S. Lee and P. L. Hsu, “Design and implementation of the SNMP agents for remote monitoring and control via UML and Petri nets,” IEEE Trans. Contr. Syst. Technol., vol. 12, no. 2, pp. 293-302, March 2004.

[9] K. Jensen, “Coloured Petri nets,” Lecture Notes Compu. Sci., vol. 254, pp. 248-299, 1987.

[10] K. Feldmann, A. W. Colombo, C. Schnur, and T. Stockel, “Specification, design, and implementation of logic controllers based on colored Petri net models and the standard IEC 1131, Part I: Specification and design,” IEEE Trans. Contr. Syst. Technol., vol. 7, no. 6, pp. 657-665, Nov. 1999.

[11] M. Dotoli and M. P. Fanti, “A coloured Petri net model for automated storage and retrieval systems serviced by rail-guided vehicles: a control perspective,” Int. J. Computer Integrated Manufacturing, vol. 18, no. 2-3, pp. 122-136, March-May, 2005.

[12] J. S. Lee and Y. F. Lee, “Behavior modeling and remote control of industrial conveyor systems via internet,” Proc. IEEE Conf. Industrial Electronics and Applications (ICIEA), Melbourne, Australia, June 2013, pp. 387-392.

[13] G. Booch, J. Rumbaugh, and I. Jacobson. Unified Modeling Language User Guide. Addison-Wesley, 2005.