

A Data Synchronization Model and Approach Based on Petri Net for Soft PLC Systems

Guanci Yang^{1,*}, Shaobo Li^{1,2}, Yang Wang², Lin He¹ and Jie Hu¹

¹Key Laboratory of Advanced Manufacturing Technology of Ministry of Education, Guizhou University, Guiyang 550025, China

²Chengdu Institute of Computer Applications, Chinese Academy of Sciences, Chengdu 610041, China

*Corresponding Author, E-mail: guanci_yang@163.com

Abstract

The efficiency of data synchronization between different modules is a significant parameter in industry control systems. Data synchronization is widely used to eliminate inconsistency and conflicts in data among subsystems. This study analyzes the data transfer process of the soft programmable logic controller system and establishes a data flow model based on Petri net. A hierarchical data structure based on a hash table is also presented, and an improved observer pattern is proposed to integrate delegation and the agent model. A new type of data synchronization approach is proposed as well. Finally, energy-saving control logic is used in one station in line seven of the Shanghai Metro to test the performance of the proposed approach. Results show that the proposed approach effectively synchronizes data.

Keywords: Petri net, observer pattern, data synchronization, soft programmable logic controller, energy-saving control system

1. Introduction

In an industrial control system, data must be collected from different modules, and data collection should be rapid and doable in real time. The efficiency of data synchronization between different modules is a significant parameter in industry control systems, and the response time of data synchronization must be milliseconds or microseconds fast [1–3]. With their increasing complexity, programmable logic controllers (PLCs) are being utilized in complex automation systems. A soft PLC is control system based on the structure of an embedded system or industrial PC. Soft PLCs are widely used in distributed and automatic controlled industrial systems [4], especially in systems requiring near-instantaneous results.

An embedded controller [5] based on soft PLC technology was adopted in the operational monitoring subsystem for process control and optimization of mineral grinding. The developed system is characterized by high real-time performance and excellent portability and is in accordance with actual industrial processes. Results showed that the system performed satisfactorily for the mineral grinding process and provided a reliable and efficient verification environment for operational control methods. Several embedded control systems employ networks for data synchronization [6]. These networks must connect the client and server before synchronization. The systems are restricted by the reliability of the network, and achieving satisfactory results at the millisecond level is difficult. The synchronized method based on black box protocol can only implement synchronization of specific data, which limit the expansibility of the soft PLC system [7, 8]. Furthermore, studies have developed the soft PLC system [9–12], but most of them focused on visualization and compiling. Efficient data synchronization has been ignored.

In this research, we focus on the energy-saving control system and develops a soft PLC system to improve the efficiency of energy-saving products. We analyze the data transfer process of a soft PLC system and the data flow model based on Petri net, and then present the hierarchical data structure based on a hash table. We propose an observer pattern integrated with delegation and the agent model. The data synchronization model and approach based on Petri net for the soft PLC system are thus obtained.

Contributions of the paper are summarized as follows.

- The data flows model based on a Petri Net for the soft PLC system has been developed.
- One improved observer pattern and data synchronization strategy for the soft PLC have been proposed.
- The data synchronization model and approach based on Petri Net for Soft PLC Systems has been implemented, which is used to optimize the environment control system of the Metro.

The remainder of this paper is organized as follows. Section 2 introduces the soft PLC system and application background. Section 3 describes the modeling of data flows based on Petri net. Section 4 presents the structure of data synchronization. Section 5 shows the improved observer pattern and data synchronization strategy. Performance testing and analysis are conducted in Section 6. Section 7 concludes this paper and provides future research issues.

2. Soft PLC System

A typical soft PLC system has debugging and compiling functions. Its general features are as follows.

- 1) Standardizing the programming language, following the IEC61131-3 standard, and supporting different programming languages that can transform others quickly;
- 2) Possessing multitudinous controlling models and supporting various PID and control algorithms;
- 3) Opening the interface of controlled algorithms, and encouraging the embedding of the controlled algorithm models by itself;
- 4) Simulating and running, monitoring in real time and online, editing and compiling the code online; and
- 5) Network communication.

In previous works [13, 14], we focused on the energy-saving control system of a central air-conditioning unit to develop one soft PLC system that made the intelligent control system easily updatable. We designed a visualization of the instructions and developed a user graphic programming interface for graphic programming of the control logics. The target system provided a function to add object entities, communicate with hardware, and obtain the keys, and so on. With this system, we configured specific aspects of the hardware, including configuring modules and ports and mapping ports between modules. Control logic, strategic programming functions, and data analysis were also provided. The following experiment is based on this system. Specifically, one previous work [13] was about creating the graphic symbol and realizing its functions based on a QT platform. This endeavor was fundamental to control logical programming. The other study [14] aimed at logic function implementation and data association with the interaction of control devices.

3. Modeling of Data Flows Based on Petri Net

3.1. Petri Net

A Petri net (PN) is a mathematical modeling language that describes distributed systems. It is a directed bipartite graph whose nodes describe transitions and places. The directed arcs represent places that are pre- and/or post-conditions for transitions (signified

by arrows). A PN focuses on organizational structure and the dynamic state of the target systems, as well as the potential relationship between each state. Furthermore, it is an excellent tool to describe asynchronous and subsequent phenomena. A PN can establish the equation of states, the equation of algebra, and other mathematic models that describe the behavior of systems [15].

Usually, a 3-triple $N=(S, T, F)$ is a PN based on the following [16]:

- 1) $S \cap T = \phi$;
- 2) $S \cup T \neq \phi$;
- 3) $F \subseteq S \times T \cup T \times S$; and
- 4) $dom(F) \cup cod(F) = S \cup T$,

where S is a finite set of places, T is a finite set of transitions, and S and T are disjoint. Specifically, no object can be both a place and a transition. Moreover, $dom(F) = \{x \in S \cup T \mid \exists y \in S \cup T, (x, y) \in F\}$, $cod(F) = \{x \in S \cup T \mid \exists y \in S \cup T, (y, x) \in F\}$.

In the diagram of a PN, the PN comprises places, transitions, and arcs. Arcs are one-way arrows that run from a place to a transition and never between places or between transitions. The places from which an arc runs to a transition are named the input places of the transition, and they are conventionally depicted with circles. The places to which arcs run from a transition are called the output places of the transition and are depicted by long narrow rectangles. If the diagram were of a basic net, then those places in a configuration would be conventionally depicted as circles, where each circle encompasses a single dot called a token. In a given diagram of a PN, the place circles may encompass more than one token to show the number of times a place appears in a configuration. The configuration of tokens distributed over an entire PN diagram is called a marking [17, 18]. When multiple transitions are enabled at the same time, any one of them may fire. Given that firing is non-deterministic and that multiple tokens may exist anywhere on the net, PNs are well suited for modeling the concurrent behavior of distributed systems.

3.2. Data Flows Based on PN

In practice, the processes of data flowing in soft PLC systems are as follows:

- 1) Collect the real-time foundation data on the scene and store the data in the object layer database.
- 2) Update data of the object layer to the global variables of the control system and the input variable of the instructions according to the scanning cycle of the soft PLC.
- 3) Apply the control strategies to refresh the data of the display layer, data view layer, and configuration files.
- 4) Refresh the latest data to the object layer database.
- 5) The object layer maps the data to the controlled hardware.

According to the theory of PNs and the analysis of data flow in the process of the system, we developed the mode of data flows based on a PN for the soft PLC system shown in Figure 1. The following are the representations of S and T :

- S_1 : status data of the controlled objects,
- S_2 : input variables of the data layer,
- S_3 : input variables in the global variable set,
- S_4 : hash table,
- S_5 : configuration files,
- S_6 : display layer,
- S_7 : view layer,
- S_8 : control strategies,
- S_9 : output variables in the global variable set,
- S_{10} : output variable in the object layer,
- S_{11} : input port of the hardware layer}

- T₁: collect the input data,
- T₂: scan and refresh the input data,
- T₃: synchronize hash table,
- T₄: update configuration files,
- T₅: update display layer
- T₆: synchronize the data of view layer,
- T₇: analysis and calculate the data,
- T₈: synchronize the output global variables
- T₉: refresh the input variables of the object layer, and
- T₁₀: update the input data of the controlled hardware to control its status }

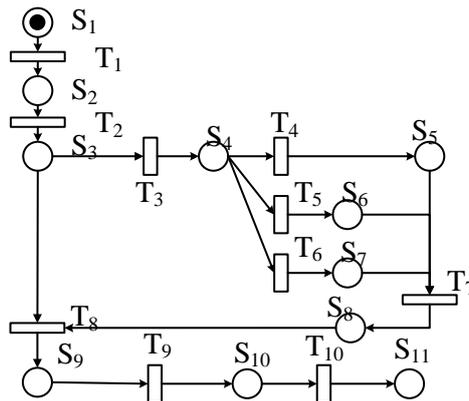


Figure 1. Mode of Data Flows Based on PN for Soft PLC System

4. Structure of Data Synchronism

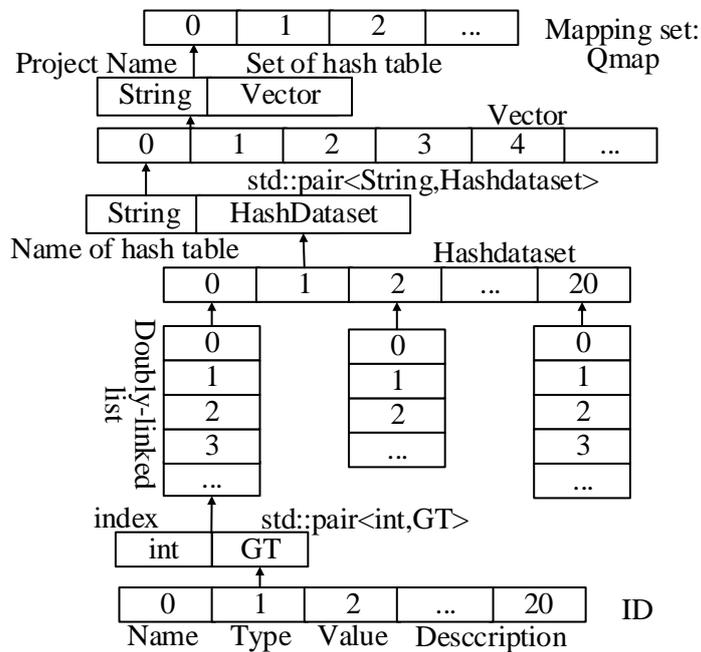


Figure 2. Data Structure of Designed Hash Table

Data exist in the object layer, and collecting and refreshing the data from the object layer frequently during the execution of control strategies are essential. To improve the operation efficiency of the soft PLC system, we try to design a reasonable data index mechanism by employing a block-based storage heap hash table. The block is a binary

string with fixed length k and an assumed set of the basic block G_k . The primary data structure $G_{d \times k}$ consists of the information block and its index, which divides the hash table into d sub-heap. Then $\forall x_0, x_1, x_2, \dots, x_{d-1} \in G_k, hash(x_0, x_1, x_2, \dots, x_{d-1})$ is defined as a hash function under the $G_{d \times k}$. The hash table can be established as follows:

$$hash(x_i) = \left(\sum_{i=0}^{tempSize} sName[i]_{ASCII} - 1 \right) \% d \quad (1)$$

where $sName$ is the attribute name of the data block, $tempSize$ is the string length of the attribute name, i is the index of one character of the attribute name, and $ASCII$ represents the ASCII values of the character.

Figure 2 illustrates the data structure of the designed hash table. In a particular control logical project, the hash tables of different subsystems belong to the same mapping set, which uses the name of subsystems to map the project and hash table.

5. Improved Observer Pattern and Data Synchronization Strategy

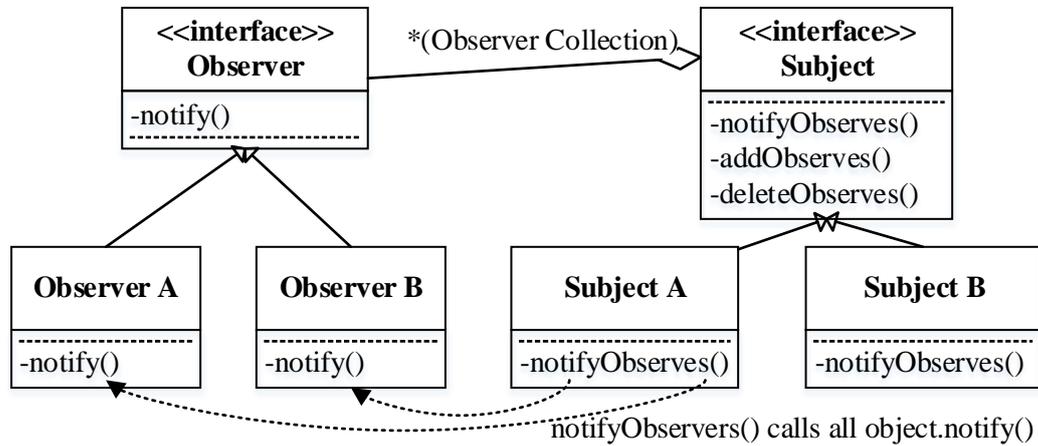
A type of two-way synchronization (bi-directional synchronization or both-ways synchronization) [19] is used to implement the synchronization between two nodes (Figure 3). This synchronization process transmits data in both directions to reconcile changes as needed. Data are expected to change in both locations/nodes. The two sites are considered equivalent. However, this type of data transmission cannot occur simultaneously, that is, the data flow is asynchronous in both directions. This prescription satisfies the modeling requirements of the PN. For example, if a set of data in node A is new, then it will be copied to node B; however, if the data are newer in node B, these will be copied to node A. Similarly, if the data are deleted from one node, then these will be removed from another node.



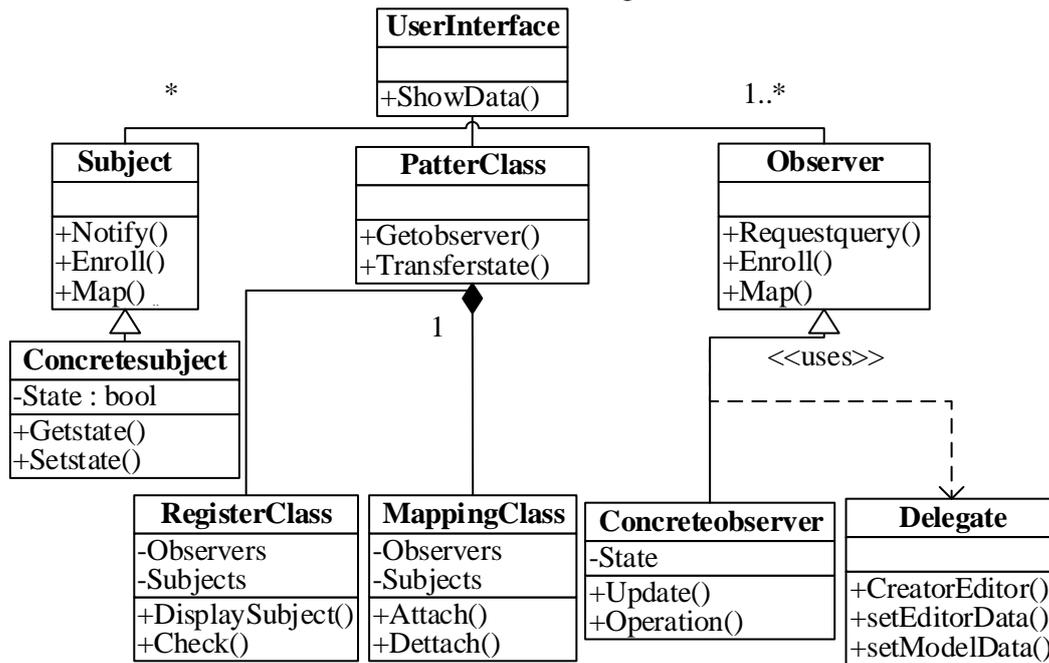
Figure 3. Illustration of Two-way Synchronization

The observer pattern [20] is a software design pattern in which an object maintains a list of its dependents, called observers, and notifies the dependents automatically of any state changes, usually by calling one of their methods. The observer pattern is a significant part of the model-view-controller (MVC) architectural pattern, as seen in Figure 4(a). Although the observer pattern is employed in many programming libraries and systems, the standard observer pattern also has several problems, such as memory leaks, because it requires both explicit registration and explicit deregistration in basic implementation, and because the subject holds strong references to the observers to sustain them in the disposal pattern. To prevent weak references to the observers, we design a mapping between subject and observer by adding a register class *RegisterClass* and a mapping class *MappingClass*. An illustration is provided in Figure 4(b). The *RegisterClass* responds to the registered subjects and observers, and the *MappingClass* takes charge of the mapping between the hash table and multiple observers to guarantee consistency. To assign proper permission to the observers with a different authority, the *DisplaySubject* method belonging to the register class provides the content list for all the observers explicitly registered to the hash table. The *Delegate* class based on the MVC mode is used to render and edit the variables of the view layer, which is capable of self-synchronous data refreshing.

Meanwhile, data synchronization also responds to the mapping between the hash table and different data objects. Star topology [21] is employed to handle this mapping (see Figure 5). In this type of structure, multiple transitions are enabled at the same time, and any of them may fire. The hash table, which is connected to all the other layers, is the center of this structure. The other layers cannot communicate with one another directly. When data need to update to another node, the original data subject sends the latest data to the hash table. Then the hash table notifies the reference layers to synchronize the data.



(a) Classic class diagram



(b) Improved class diagram

Figure 4. Class Diagram of Observer Pattern

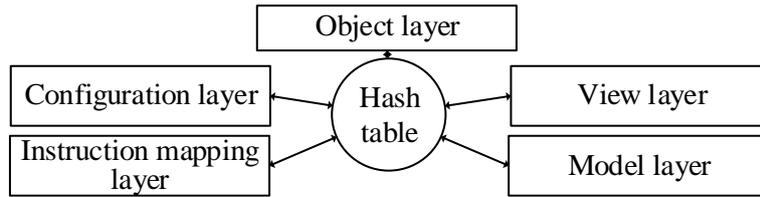


Figure 5. Synchronization Structure Based on Star Topology for Soft PLC System

6. Performance Testing and Analysis

6.1. Experimental Setup

C++ was used to implement the proposed data structure, model of data flow, and data synchronization strategy in the QT platform. These components were integrated into our soft PLC system for energy-saving control.

We employed a simulation and debugging platform for a central air-conditioning system (Figure 6(a)), embedded controller (Figure 6(b)), and efficient frequency control cabinet MSC-400 (Figure 6(c)). These components were produced by our research cooperation partner, Guizhou Huitong Huacheng Co., Ltd.



(a) Simulation and debugging platform of central air-conditioning system



(b) Embedded controller



(c) MSC-400: efficient frequency control cabinet

Figure 6. Equipment Used for the Test

The simulation and debugging platform of the central air-conditioning system simulates the status data and load characteristic of the controlled systems/terminals/devices as well as the required environmental parameter sampling procedures, such as delivery and return air temperature, concentrations of carbon dioxide, temperature, and humidity of the monitoring area.

The embedded controller running the developed soft PLC system used as configuration software programs the control logic; it provides the optimal control parameters for efficient frequency control cabinet MSC-400. The parameters are obtained by running the optimal control algorithm according to the simulated data submitted by the simulation and debugging platform of the central air-conditioning system.

The efficient frequency control cabinet MSC-400 controls the controlled systems/terminals/devices and provides them feedback on the dynamic information from the simulation and debugging platform of the central air-conditioning system.

Figure 7 is the block diagram for the test, which is simplified from the environment control system used at one station of Shanghai Metro. The test system involves 80 data maps and includes the water temperature, flux, valve opening of branches, temperature and humidity of the monitoring area, running status of the central air-conditioning unit, and so on.

Meanwhile, focusing on checking the performance of dealing large-scale data mapping, we set the systems to have 400, 800, 2400, and 8000 data mapping by paralleling the basic 80 data mapping control logic. That is, five sets of data mapping are used to check the performance of the proposed approach. Otherwise, the scanning period is set to 500 ms.

All data used to check the performance were historical data collected by the sensors from one station of Shanghai Metro.

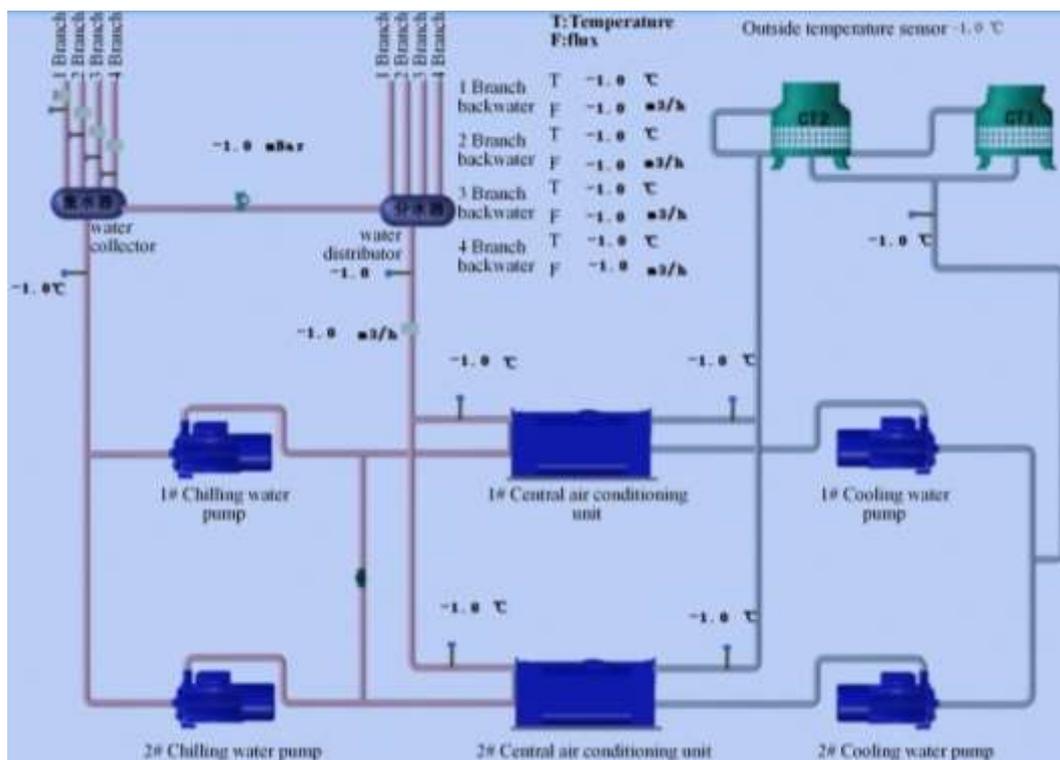


Figure 7. Environment Control System Used on One Station of Shanghai Metro

6.2. Analysis

The time consumption of data synchronization in 50 scanning periods are sampled at a scanning period 500 ms. The results are shown in Table 1. Figure 8 is the scatter plot of the data synchronization time in different sampling stages. Its horizontal ordinate is the sequence number of the 50 sampling stages, and the vertical ordinate represents the time

of data synchronization of the system, which indicates that the interval of time increases along with the increase of data scale. This increase is not the same as the data size.

Table 1 shows that the minimum time to finish the data synchronization is 12 ms and the maximum is 17 ms. When the variable scale of data mapping increases, the average time consumption also increases. However, the increase ranges of data synchronization time are 10.00%, 10.00%, 18.35%, and 126.79% as the data scale increases 5, 10, 30, and 100 times, respectively. This trend indicates that the proposed approach synchronizes data effectively because it has slower time consumption compared to the rapid increase of data.

Table 1. Time Consumption of Data Synchronization with Different Data Scales

No	Data scale	Sample size	Min/ms	Max/ms	Average/ms	Variance
Group 1	80	50	12.0	17.0	14.7	1.99
Group 2	400	50	13.0	18.9	16.1	2.63
Group 3	800	50	14.1	21.0	17.4	3.12
Group 4	2400	50	15.4	29.9	23.2	13.53
Group 5	8000	50	28.2	39.3	33.3	9.12

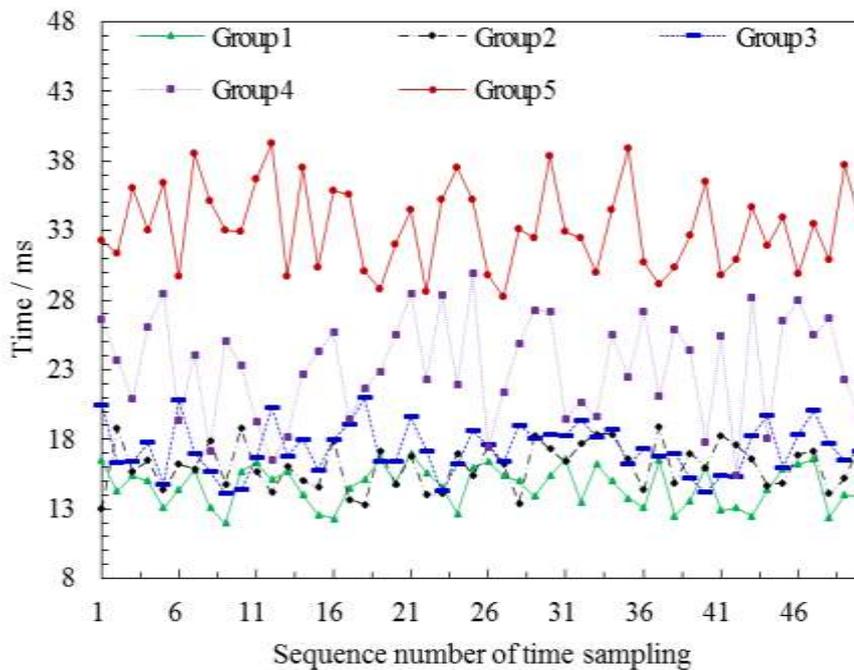


Figure 8. Scatter Plot of Data Synchronization Time in Different Sampling Stages

7. Conclusion

A fully functional PLC process controller was developed with software technology based on the IEC61131-3 protocol. Soft PLC technology has become a heavily researched topic for its flexible development, open architecture, high cost-effectiveness, easy linking to networks, reliable portability, and convenient programming. Soft PLC technology has been used in thousands of applications worldwide. An increasing number of organizations are discovering the tremendous advantages of the soft PLC over PC-based control systems that rely upon problematic Windows operating systems. The soft PLC's closeness

to embedded open architecture control allows for its installation in tight conditions for even the most challenging applications, including those requiring quick response speeds, a significant amount of communication, and high reliability. Our research focused on the energy-saving control system and developed a soft PLC system to improve the efficiency of the energy-saving products. Data synchronization is important for eliminating data inconsistency among different modules. We analyzed the data transfer process of a soft PLC system and data flow model based on PN. A hierarchical data structure based on a hash table was detailed, and an observer pattern integrated with delegation and the agent model was proposed. The data synchronization model and approach based on a PN for soft PLC systems were then obtained. Results showed that the proposed method effectively synchronizes data. We will also conduct experiments on more complex control logic involving more data mapping and synchronization.

Acknowledgments

The authors would like to thank the financial support provided by National Natural Science Foundation of China under Grant 61540066 and 51475097, Science and Technology Foundation of Guizhou Province under Grant R[2015]13, JZ[2014]2004, JZ[2014]2001, ZDZX[2013]6020, G[2014]4001, ZDZX[2014]6021, and [2014]3037 Torch Program projects of Guizhou Province under Grant [2013]5051, and National Key Technology Support Program of China under Grant 2012BAH62F00.

References

- [1] G. Liang, Z. Li, W. Li, et al., "On an LAS-integrated soft PLC system based on WorldFIP fieldbus," *ISA TRANSACTIONS*, vol. 51, (2012), pp. 170-180.
- [2] Z. Li, "Research and Implementation of New Algorithm about Transforming Soft PLC Ladder Diagram to Instruction List," *Journal of Liaocheng University (Natural Science Edition)*, vol.26, no.1, (2013), pp. 105-110. (in Chinese)
- [3] Y. I. Al, "Design and Implement of a Programmable Logic Controller (PLC) for Classical Control Laboratory," *Intelligent Control and Automation*, vol.3, no.1, (2012), pp. 44-49.
- [4] Y. Niu, L. Cao, X. Wu, and S. Song, "Optimal co-design of control and scheduling for distributed industrial automation based on embedded soft PLC," *Advances in Information Sciences and Service Sciences*, vol. 4, no.13, (2012), pp. 135-143.
- [5] Y. Qin, W. Dai, J. Yang, and P. Zhou, "Design and experiment of operational control system for mineral grinding process based on soft PLC technology," *Journal of Northeastern University*, vol.36, no.3, (2015), pp. 309-313+317. (in Chinese)
- [6] T. Levendovszky, A. DUBEY, W. R. Otte, et al., "Distributed real-time managed systems: A model-driven distributed secure information architecture platform for managed embedded systems," *Software, IEEE*, vol. 31, no.2, (2014), pp. 62-69.
- [7] C. Hennig, K. Kneupner, D. Kinna, et al., "Connecting programmable logic controllers (PLC) to control and data acquisition a comparison of the JET and Wendelstein 7-X approach," *Fusion Engineering and Design*, vol. 87, no.12, (2012), pp. 1972-1976.
- [8] P. Lin, Q. Li and P. Jia, "A Real-Time Temperature Data Transmission Approach for Intelligent Cooling Control of Mass Concrete," *Mathematical Problems in Engineering*, no.1, (2014), pp.1-10.
- [9] X. Wang, F. Zhou, S. Sun, and J. Li, "Development and implementation of Soft PLC compiling system," *Journal of Beijing University of Technology*, vol. 34, no.11, (2008), pp. 1139-1144. (in Chinese)
- [10] D. Schütz, A. Wannagat, C. Legat, et al., "Development of PLC-based software for increasing the dependability of production automation systems," *IEEE Transactions on Industrial Informatics*, vol. 9, no.4, (2013), pp. 2397-2406.
- [11] H. Ding, L. Sun and X. Yu, "The Design of Clamping Positioning Equipment with Unilateral Floating Based on Soft PLC Control System," *Modular Machine Tool & Automatic Manufacturing Technique*, no.6, (2015), pp. 140-143. (in Chinese)
- [12] C. Chen, S. Gao and J. Wang, "Implementation of Ladder Diagram Edit Module of Soft PLC Development System," *Modular Machine Tool & Automatic Manufacturing Technique*, no.10, (2010), pp. 64-67. (in Chinese)
- [13] Z. Dai, G. Yang and S. Li, "Analysis and Implement Graphical of Soft PLC Logical Control Elements," *Modular Machine Tool & Automatic Manufacturing Technique*, no.10, (2012), pp. 74-76. (in Chinese)
- [14] W. Chen, G. Yang, S. Li, et al., "Modular design and implementation of soft-PLC's logic control unit for intelligent control," *Journal of Guizhou Normal University (Natural Sciences)*, vol.31, no.5, (2013), pp. 96-100. (in Chinese)

- [15] E. Serral, J. De Smedt, M. Snoeck, et al., "Context-adaptive Petri nets: Supporting adaptation for the execution context," *Expert Systems with Applications*, vol. 42, no.23, (2015), pp. 9307-9317.
- [16] J. Luo, H. Ni and M. Zhou, "Control program design for automated guided vehicle systems via Petri nets," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no.1, (2015), pp. 44-55.
- [17] S. A. Khan, F. Ahmad, I. Fakhir, et al., "Structural analysis methods for Petri net based control systems: A review," *Journal of American Science*, vol. 8, no.12, (2012), pp.834-843.
- [18] M. Zhou and K. Venkatesh, "Modeling, simulation, and control of flexible manufacturing systems: a Petri net approach," *Mechatronics*, vol. 11, no.7, (2001), pp.947-950.
- [19] U. Mengali, "Synchronization techniques for digital receivers", Springer Science & Business Media, New York, (2013).
- [20] K. Aljasser, "Implementing design patterns as parametric aspects using ParaAJ: The case of the singleton, observer, and decorator design patterns," *Computer Languages, Systems and Structures*, vol. 45, no.C, (2016), pp. 1-15.
- [21] L. Zhao, X. Chi, P. Li, and L. Guan, "A MPR optimization algorithm for FSO communication system with star topology," *Optics Communications*, vol.356, (2015), pp. 147-154.

Authors



Guanci Yang, he is an associate professor at the Key Laboratory of Advanced Manufacturing Technology of Ministry of Education, Guizhou University, China. He received his Ph.D degree in Computer Software and Theory from Graduate University of Chinese Academy of Sciences, Chengdu, P.R. China, in 2012. His current area of interest mainly focuses on computational intelligence, human–robot interaction, and intelligent control systems.



Shaobo Li, he is a professor at the School of Mechanical Engineering, Guizhou University (GZU), China. He is the Dean of the School of Mechanical Engineering at GZU since 2015. He is the author of more than 130 papers in major journals and international conferences. His current research interests include intelligent manufacturing, The Internet of things technology and big data.



Wang Yang, he is a Ph.D. student at the Chengdu Institute of Computer Applications, Chinese Academy of Sciences. He obtained his M.S. degrees in Computer Application from Guizhou University, Guiyang, P.R. China, in 2014. His major interests lie in big data & analytics, and distributed data processing.



Lin He, she is currently an associate professor at the Key Laboratory of Advanced Manufacturing Technology of Ministry of Education, Guizhou University, China. She received his Ph.D. degree in Mechanical Engineering from Nanjing University of Science and Technology, Nanjing, P.R. China, in 2012. Her current area of interest mainly focuses on intelligent control systems and structure optimization design.



Jie Hu, he is Ph.D. student at the Key Laboratory of Advanced Manufacturing Technology of Ministry of Education, Guizhou University, China. He obtained his B.S. degrees in Computer Science and Technology from Guizhou University, Guiyang, P.R. China, in 2012. His research interests in natural language processing, deep learning, big data & analytics, and intelligent control systems.