

THE TEMPLATE MODEL APPROACH FOR PLC SIMULATION IN AN AUTOMOTIVE INDUSTRY

Minsuk Ko, Daesoon Chang, Ginam Wang, Sang C. Park
Department of Industrial Engineering
University of Ajou
Suwon, Korea
sebastianminsuk@gmail.com

KEYWORDS

Control Simulation, PLC Program, Virtual model, Simulation.

ABSTRACT

Product systems are quickly and frequently changed because product life cycle is continuously reduced and adopting new product is steadily fast. Thus, various studies are progressed using PLC simulation which is one of control level simulation. The research that is concerning simulation of control verification for shortening the commissioning which has a lot of trial and error is in progress. Also, simulation of control verification has strength that it can catch the errors in advance. However, a virtual device in simulation needs both physical and logical model for representation of control information. For this reason, excessive time and energy is put into constructing virtual models. So, in this paper, we proposed the template model approach for the PLC simulation to cope with this problem. The proposed template model approach provides an efficient construction method for a virtual model based on control information, extracted from the test run procedure of the PLC program. We can minimize the set up time of the PLC simulation environment, as well as cut process time down using our suggestion.

1. INTRODUCTION

As product life cycles are reduced in the continuously changing marketplace, modern manufacturing systems must have sufficient responsiveness to adopt their behaviors efficiently to a wide range of circumstances. To respond to these demands, including high productivity and production flexibility, the use of the concept of a virtual manufacturing (VM) has been widely accepted. VM is a key concept that summarizes computerized manufacturing activities dealing with models and simulations instead of objects and their operations in the real world (B.K. Choi 2000). VM was applied to the small size (cell) manufacturing system in the past. However, recently, with the development of computer technology, it is possible to apply VM technology to the huge size (line, factory) manufacturing system (Hibnio 2006). As part of this revolution, offline programming for robots (Orady 1998) and verification of the control program (David 1998; H. T. Park 2010; S.

C. Park 2008) with virtual models have emerged in an automotive industry.

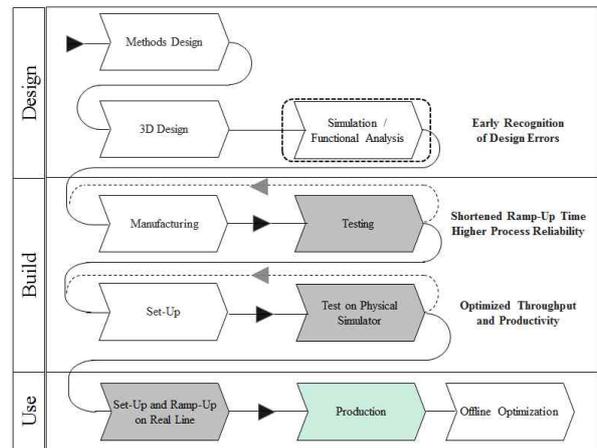


Figure 1. Manufacturing process in production system

Figure 1 shows a manufacturing process that applies to virtual manufacturing technology (VMT). Since early recognition of design errors could be found at the design phase, and design (mechanical & electronic) verification could be done at the build phase, applying VMT to manufacturing process could reduce time for feedback the verified information from the use phase to the design phase. Especially, a simulation based on the VMT has been considered an essential tool in the design and analysis of complex system that cannot be easily described by analytical or mathematical models (Al-Ahmari 1999; Klingstam 1999). Since the implementation of a manufacturing line requires heavy investment, many companies applied VM simulation to the production system design to ensure that a highly automated manufacturing system will successfully achieve the intended benefits.

Automotive manufacturers continuously strive to improve their production system, as well as their products, to remain competitive in this continuously changing market place. Automotive manufacturing is complex task with several automated functions. Machined or purchased components are assembled on sub-systems, and the sub-assemblies are assembled during the final assembly system. Therefore, many engineers in automotive industry want to create much more realistic simulation models, which can forecast not only the production capability of the system but also the logical verification, physical validity, and efficiency of

co-working machines.

This demand has resulted in the concept of PLC simulation. PLC simulation can be described as a model executing digital manufacturing processes within a computer simulation. To verify the mechanical and electrical designs of the production system, it provides a realistic effect as a test run for the production system using both the 3D graphic model that appear to the same as real shop floor and the logical model that drives connected a PLC in a real factory.

Considerable work has been undertaken in the field of PLC simulation and validation of the production system. C.M Park (2009) proposed that verifies the controller logic process for the automobile industry via simulation using a state-based model that creates a virtual car body assembly line. Xusong (2006) discussed the concept, the modeling method of building an AS/RS (automatic storage and retrieval system) real time simulation and control integrated system based on the AS/RS control system structure. He proposed key features and the more significant methods used in building simulation models by using virtual reality. S.C Park (2008) proposed the architecture of a PLC program environment that enables a visual verification of PLC program integrated a PLC program with corresponding plant model. The proposed architecture provides a theoretical construction method for a plant model based on the DEVS (Discrete Event Systems Specifications) formalism for the PLC simulation. H.T Park (2010) reports an automated procedure for constructing a plant model for a simple manufacturing cell. Although those researches contributed to construct a theoretical architecture for a control level simulation environment in automobile industry, they remain limited about efficient way to construct an adjusted simulation model. The objective of this paper is to develop an efficient method for constructing models for PLC simulation in an automotive manufacturing system. The proposed model construct method employs a template model consisting of the physical model, and the logical model. The overall structure of the paper is as follow. Section 2 illustrates the architecture of the proposed template based modeling methodology, while Section 3 describes an efficient construction methodology for a template, which can be synchronized with a control program. Section 4 shows an example and illustrations. Finally, concluding remarks are given in Section 5.

2. APPROACH FOR CONTROL LEVEL SIMULATION

Figure 2 shows a conventional implementation procedure of a production system. After the abstract level simulation is performed to verify or optimize the designed process, the mechanical and electrical design phases are performed sequentially. Mechanical design makes layout and physical models (2D, 3D models) that are used for manufacturing devices. It has to be verified with mechanical simulators to inspect motion, kinematic,

and assembly errors. On the other hand, we can get control programs as well as logical models through electrical design. To get control programs through the electrical design, usually, electrical engineers manually write control programs by referring to the rough control logic of abstract level simulators. Since control programming is a very tedious and error-prone job, it is essential to verify the control programs offline to reduce the stabilization time of a production system. (Chang Mok Park 2009; David 1998; S.C Park 2008; H.T Park 2010).

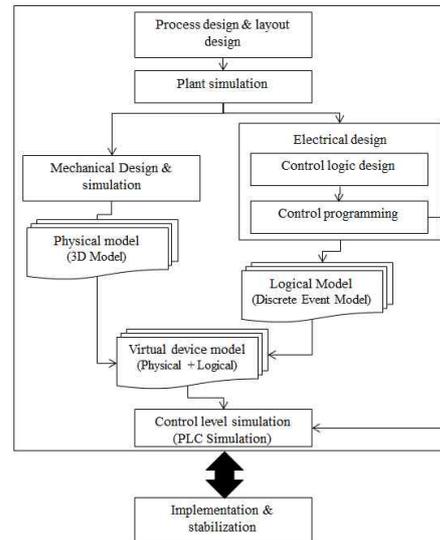


Figure 2. A process for constructing the control level simulation

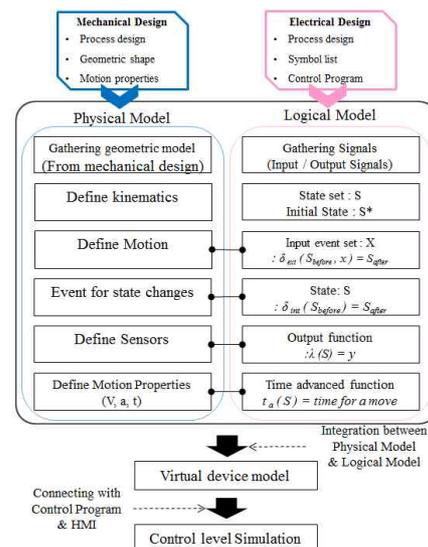


Figure 3. A process for constructing the control level simulation model

Since PLC programs only contain the control information, without device models, it is necessary to build a corresponding virtual model to perform simulation, as shown in Figure 2. However, constructing a virtual device model using for the physical model and the logical model alike requires an excessive amount of

time and effort, because both models are not suitable for the simulation to use immediately. Sometimes, the virtual model construction requires much more time than PLC programming. This serves as the motivation for exploring the possibility of finding a template based modeling methodology for building a virtual model from given both models.

To construct the PLC simulation environment, it is important to understand basic procedure for implementing components used to consist of the PLC simulation. Figure 3 shows general procedure for constructing virtual model in an automotive industry. There are three models, a physical model, a logical model, and a virtual model. A logical model represents the behavioral aspects of a device, while a physical model contains the mechanical aspects of a device. The logical model of a virtual device acts as an interface communicating with PLC programs and determines the behavior of the physical models. Once both physical and logical models are obtained, it becomes possible to make the virtual model, which is a digital model imitating the physical and logical aspects of a real manufacturing device. Because the virtual device model contains the logical core expressed by DEVS formalism and the physical shell shows corresponding motions reflecting the change of states of the logical core during the simulation, it represents such a manufacturing device in the real production system.

Table 1. Four steps for verifying control program based on digital manufacturing technology

#	Topic	Input	Output
1	Control simulation environment	<ul style="list-style-type: none"> • Symbol list • PLC Program • Process Design 	<ul style="list-style-type: none"> • Logical model (DEVS Model with control signals)
		<ul style="list-style-type: none"> • 3D Model • Layout 	<ul style="list-style-type: none"> • Physical model (Geometric + Motion + Layout)
		<ul style="list-style-type: none"> • Logical model • Physical model 	<ul style="list-style-type: none"> • Virtual model (Logical model + Physical model)
2	Manual Mode Simulation	<ul style="list-style-type: none"> • HMI • Virtual model 	<ul style="list-style-type: none"> • Error report <ul style="list-style-type: none"> ▪ Device operation error ▪ Error processing
3	Automatic Mode Simulation	<ul style="list-style-type: none"> • HMI • Robot Program • Product code 	<ul style="list-style-type: none"> • Error report <ul style="list-style-type: none"> ▪ Entire process ▪ Error processing ▪ Interaction between facilities
4	Control simulation result review	<ul style="list-style-type: none"> • Error report • Time chart of control signals 	<ul style="list-style-type: none"> • Verified PLC program

To apply the virtual device model to the PLC simulation connected with field environment, we have to proceed sequentially four topics as shown in Table.1. After constructing the simulation environment using virtual models and layout, users have to proceed 'Manual mode simulation' and 'Automatic mode simulation'. Each of both simulation modes has different purpose as shown in Table.1. The input factors of the both modes, HMI means a user interface for the local control console to perform control using a PC. After doing the simulation, users could obtain both the verified control program and the report containing the time chart of the control signals.

In this PLC simulation environment, the behavior of a virtual device model should be the same as the device that of actual system to achieve the control level verification. However, users have to invest so much the time and effort of building the virtual device model suitable for the control level environment, as shown in Figure 4. It is necessary to add virtual sensors and correct motions to achieve the intended task for each task to the physical model when a process design change occurs. Furthermore, the most time-consuming task in developing the logical model is analyzing device behaviors based on the process information, because the process design information contains only the process sequence of the production system not control level information. Therefore, to define the logical model of a device as a DEVS model, the user has to analyze the device behavior specifically according to the signals in control program. Since the logical model has high modeling DOF (Degree Of Freedom), users have to consider how to determine the set of DEVS components to represent device behaviors.

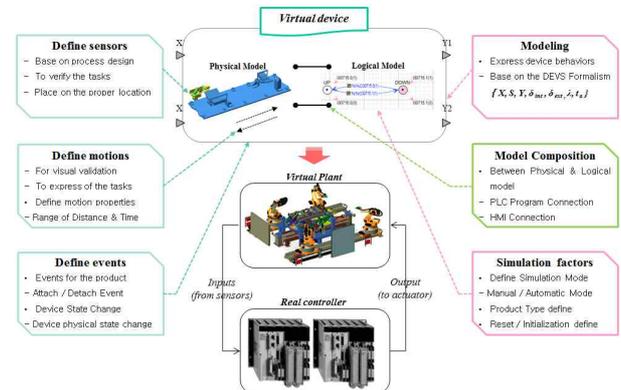


Figure 4. Components of a virtual device model

Once the both model are determined suitable for the simulation, then they have to be connected with one another. The output of this step is usually the virtual device model which becomes a practical guideline for the simulation. Obviously, manufacturer can benefit greatly from PLC simulation using the virtual device model, but there are still many difficulties complicating the full utilization of the virtual device model. One of the main obstacles to build a logical model comes from the understanding device behaviors which are set of tasks that are assigned to the device. Since a logical model interacts with a virtual factory may consist of hundreds of machines and product, it is difficult to find a modeling error of the designed logical model, and physical model during the simulation. Sometimes, finding modeling errors of the virtual device model becomes a bottleneck in the simulation time delay. Therefore, it is necessary to verify both logical model and physical model before the simulation.

We propose a template based modeling approach of the PLC simulation to cope with the problem. The proposed template model has a role for the practical guideline of the virtual device model, since it has

containing process knowledge information in legacy simulation results. In addition that it is possible to verify the validity of the task of the virtual device model and sensor conditions. Therefore, the proposed method can reduce the modeling time and errors of the simulation model instead of reducing the modeling DOF.

3. TEMPLATE MODEL OF THE VIRTUAL DEVICE

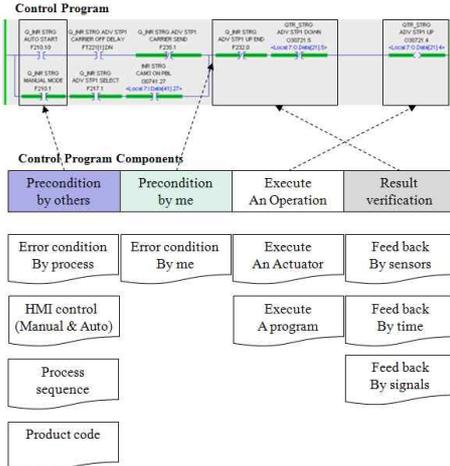


Figure 5. Control program components

To construct a template model of the virtual device, we extracted five major control components from the control program, as shown in Figure 5. PLC program starts with preconditions which are occurred by the others. To proceed this condition, it is necessary to satisfy error conditions caused by the other device, process sequence, HMI signals, and product code. After checking the *precondition by others*, the PLC program checks the self-condition of the device. These preconditions have a role for checking whether the device is located at the proper position to execute the task or not. Once the conditions satisfied, then the device executes an aimed task. The output of the task is usually verified with sensor, timer, or a signal which indicating the task is complete. Since the proposed template model contains these components in the logical model, and physical model, the modeling errors can be avoided. Since signals in the *precondition by others* are occurred by other devices or the controlling process, it is necessary to build a HMI model to send combined signals to the virtual device. If this information is predefined, users can instinctively understand about the controlled device in the control level.

Figure 6 shows a simple template model using a sample of an AGV (Automatic Guided Vehicle) with two tasks, T1 (movement from p1 to p2) and T2 (movement from p2 to p1). As the two tasks should triggered by external events, terms here as E_X1 and E_X2 . When the user select this template model, it is possible to instantiate the logical model automatically. And, the physical model is instantiated from the

predefined physical model, which has two motions $M1$, $M2$, and two position sensors $S1$, $S2$. And functional relationships will be automatically defined between DEVS components of the logical model and physical activities in the physical model, as shown in Figure 3. Using this template, users can intuitively understand both the logical and the physical aspect of the device. The logical model (DEVS atomic model) of the virtual device, corresponding to the AGV, can be described as follows:

$$\begin{aligned}
 M &= \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, t_a \rangle \\
 X &: \{ X1, X2 \}; \\
 S &: \{ S_P1, M_P1_P1, S_P2, M_P2_P1 \} \\
 Y &: \{ Y1, Y2 \}; \\
 \delta_{int}(M_P1_P2) &= S_P2 \\
 \delta_{int}(M_P2_P1) &= S_P1 \\
 \delta_{ext}(S_P1, X1) &= M_P1_P2 \\
 \delta_{ext}(S_P2, X2) &= M_P2_P1 \\
 \lambda(M_P1_P2) &= Y1; \lambda(M_P2_P1) = Y2 \\
 t_a(M_P1_P2) &= T1; t_a(M_P2_P1) = T2
 \end{aligned}$$

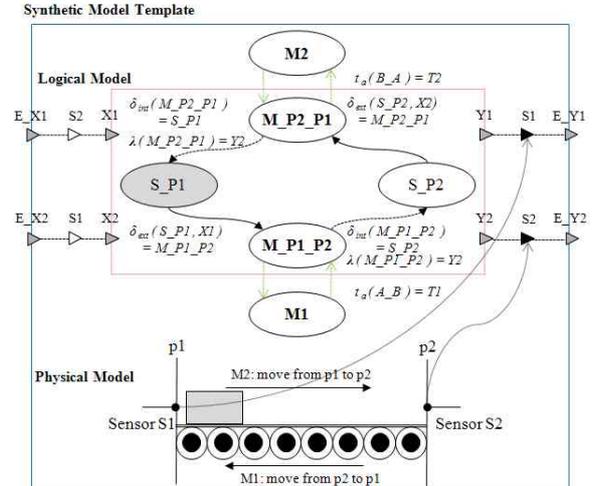


Figure 6. A template model of the virtual device for PLC simulation

This simple template model contained the major components of the PLC program, as mentioned above. Since $S1$, $S2$ have a role for sensing the AGV position, the preconditions of the AGV can be verified. And the time to travel between $P1$ and $P2$ can be used for verifying the AGV motion verification. If the time is not satisfied with the purpose of the motion, the position of sensors or motion properties has to be readjusted. Finally, the external functions have to be verified with the sensors before sending an external signal. To export E_Y1 , which verifies the task is finishing, it is necessary to satisfy the $S1$ and $Y1$ simultaneously. When the simulation has to be initialized by the process reset signal, physical model moves to the $P1$ position and the state of the logical model set up with initial state S_P1 .

4. EXAMPLE & ILLUSTRATIONS

As depicted above, we will describe an applying method to the Side Inner Line (called SIL) and Side Quarter Line (called SQL), which have a role for transporting a side inner (side quarter) panel from the AS/RS to the side assembly line, in this chapter. The SIL (SQL) line is typical example of the flow shop. Various types of side sub panels are transported by the handling robot, and conveyor system.

The detailed process of transporting a panel from the pallet in the AS/RS to the side assembly line will be explained with the help of a work cell shown in Figure 7. The work cell consists of three handling robots {ROBOT_1, ROBOT_2, and ROBOT_3}, two conveyor system {CONVEYER_1, CONVEYER_2}, and two idle jigs {JIG_1, JIG_2}. A scenario for this work cell is as follow: If a panel arrives at 'PLT_1', then 'ROBOT_1' moves the panel to 'CONVEY_1'. On the other hand, if a panel arrives at 'PLT_2', then 'ROBOT_2' moves the panel to either 'JIG1' or 'JIG2' by the car type. And 'ROBOT_3' moves the panel from the 'JIG_1 (JIG_2)' to the 'CONVEYOR_2'. When the panel arrived at the conveyor, it moves the carrier which is loading the panel from the front station 'LIFTER_FRT' to the rear station 'LIFTER_RR'. If the carrier arrives at 'LIFTER_RR', it will be removed by the handling robot of the side assembly line, which is the next step production line of the SIL (SQL) line.

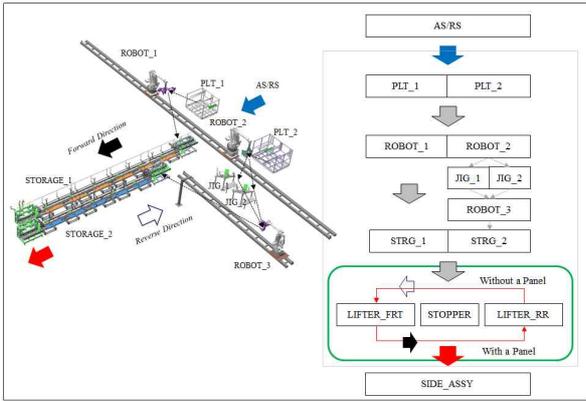


Figure 7. A process flow of Side Quarter and Side Inner line

The proposed template methodology of a virtual device model will be explained with the help of the conveyor system sub-system of the SIL (SQL). It is necessary to identify the behaviors and the tasks assigned to the conveyor to define sub-devices assembled to the conveyor. Although there are various models of the conveyor could be defined by the virtual model, we extracted simple templates for the virtual model, which can explain the behaviors of the conveyor. Figure 8 shows mechanical design of the conveyor system, which consists of ten stoppers and two lifters. To proceed the carrier, the conveyor has used the motor, which continuously running the belt, and stoppers to prevent the carrier to proceed. To describe this system

realistically, we have to construct a virtual device model of the motor moving the carrier on the belt. However, it is difficult to describe behaviors of the motor as the virtual device model, since it has continuous behaviors, running continuously until receiving the stop signal. To remedy this problem, we analyzed both mechanical design and PLC program of the conveyor system. Stoppers are attached to the conveyor (top layer: forward direction, bottom layer: reverse direction) to move the carrier circularly, and the distance between stoppers is uniformly defined as $d_{stopper}$, as shown in Figure 8. And control logic for the conveyor to move the carrier using stoppers could be defined as follow;

```
IF( STP( N+1 ) Has NO CARRIER && STP( N+1 ) is UP ) { STP(N) go DOWN; break; }
ELSE IF ( STP( N+1 ) Has a CARRIER && STP( N ) is DAWN ) { STP(N) go UP; break; }
```

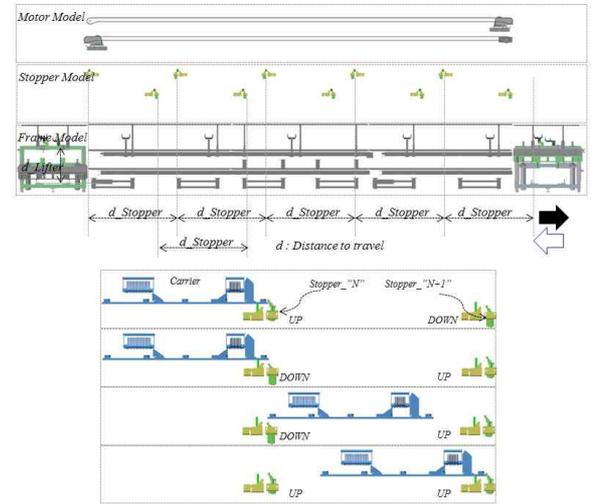


Figure 8. Mechanical design of an automated conveyor

The DEVS atomic model of the virtual device, corresponding to the stopper, can be described as follow;

```
X: { STP_UP, STP_DOWN };
S: { UP, MOVE_FORWARD, DOWN, MOVE_REVERSE };
Y: { STP_UP_DONE, STP_DOWN_DONE };
δint ( MOVE_FORWARD ) = DOWN
δint ( MOVE_REVERSE ) = UP
δext ( UP, STP_UP ) = MOVE_FORWARD
δext ( DOWN, STP_DOWN ) = MOVE_REVERSE
λ( MOVE_FORWARD ) = STP_DOWN_DONE
λ( MOVE_REVERSE ) = STP_UP_DONE
ta( MOVE_FORWARD ) = TIME_TO_TRABEL_FWD
ta( MOVE_REVERSE ) = TIME_TO_TRABEL_BWD
```

Figure 9 shows a virtual model template of the stopper. To define the physical model of the stopper, which describes the motions for transports the carrier on the belt, we defined the *virtual probe* model obtained physical properties of the belt such as velocity, acceleration, and deceleration. The *virtual probe* has the

predefined motions ‘M_Forward (start to end position)’ and ‘M_Reverse (end to start position)’ and an initial position. When the state ‘UP’ was transitioned to ‘Move forward’ by the ‘STP_DOWN’, the *virtual prove* attach the nearest carrier and move forward the distance $d_{stopper}$, then detach it at the destination. Furthermore, a virtual sensor of the virtual model sends a signal to the PLC to indicate whether the carrier is arrived at the proper position or not. This template model of the stopper is applied to all stopper models in the conveyor. Although the signals used in the PLC is different, we used the template model repeatedly according to connect signals to the proper ports of the template. As a results, we can reduce the time for constructing the virtual model, and carry out the ‘manual mode’ simulation efficiently.

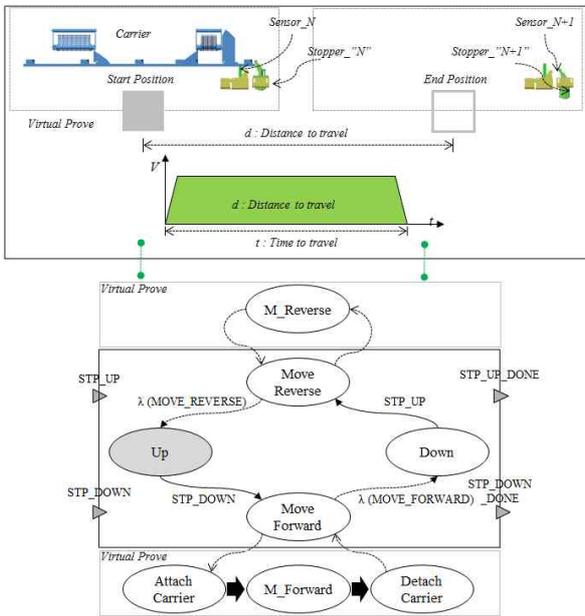


Figure 9. A virtual model template of a stopper

We could obtain the virtual device models from the proposed template model of robot, jig, stopper, and lifter. Since virtual models are built by the template, it could inherit the template properties of each model. Figure 10 shows a virtual model of the stopper and Figure 11 shows a virtual model of the vertical lifter, which consists of ‘PUSHER’, ‘LOCKER’, ‘JIG’, ‘LIFTER’.

We applied the proposed method to the SIL (SQL) line constructed by Korea automaker. The proposed method was implemented in C++ language, and test run were made on a personal computer, as shown in Figure 12. The PLC program was written using RS Emulator of the PLC program provided by AB Corporation.

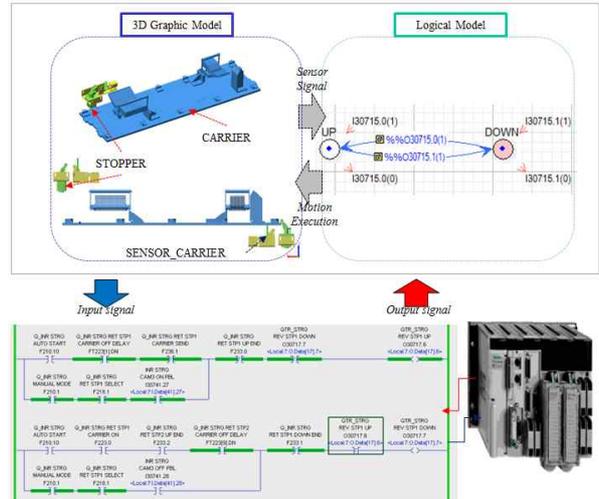


Figure 10. Virtual devices model of the stopper & vertical lifter

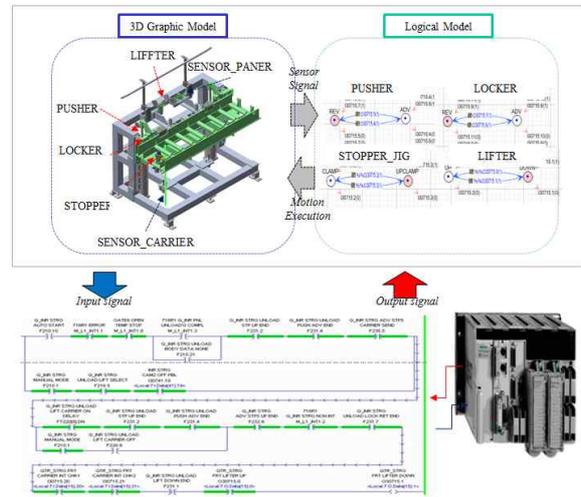


Figure 11. Virtual devices model of the stopper & vertical lifter

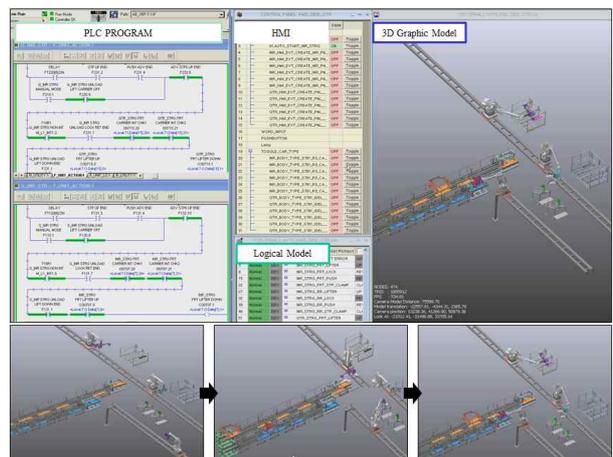


Figure 12. PLC simulation environment of SQ & SIL

5. CONCLUSION

To minimize the stabilization time of a production system to reduce the time to market, it is necessary to carry out the PLC simulation using virtual device

models to fix various errors caused by mechanical devices not being properly adjusted, faults in the PLC program. However, constructing a virtual device model using both physical model and logical model requires an excessive amount of time and effort, because both models are not suitable for the simulation to use immediately. To remedy this problem, this paper proposes an efficient method for constructing the virtual device model for the PLC simulation using template method. The proposed template model contains the logical core expressed by DEVS formalism and the physical shell shows corresponding motions reflecting the change of states of the logical core during the simulation, it represents such a manufacturing device in the real production system. Thus, template could be used as a guide line for the production system, as well as creating the virtual model using given signals. We applied this proposed method for the PLC simulation in the SIL (SQL) line manufactured by Korean automaker. After finishing the project, we stored the template model to use for the next simulation project, since it is possible to reuse the contained physical and logical models repeatedly.

ACKNOWLEDGEMENTS

This work was supported by the Defense Acquisition Program Administration under the Contract No. UD10009DD and the Agency for Defense Development under the Contract No. UD110006MD. This paper was also completed with Ajou University Research Fellowship of 2011. The authors wish to express sincere gratitude for the financial support.

REFERENCES

- Al-Ahmari, A.M.A. and Ridgway, K., *An integrated modeling method to support manufacturing system analysis and design*. Computers in Industry, 38 (3), 225–238, 1999.
- Anglani, A., et al., *Object-oriented modeling and simulation of flexible manufacturing system: a rule-based procedure*. Simulation Modeling Practice and Theory, 10(3-4), 209–234, 2002.
- B.K. Choi, B.H. Kim, *New trends in CIM: virtual manufacturing systems for next generation manufacturing*, in: Current advances in Mechanical Design and Production Seventh Cairo University International MDP Conference, Cairo, February 15-17, 2000, pp. 425-436.
- Chang Mok Park, Sang C. Park, and Gi-Nam Wang, *Control Logic Verification for an automobile body assembly using simulation*, International Journal of Production Research, Vol. 47, No 24, 2009
- David, J.D., *Application and benefits of real-time I/O simulation for PLC and PC control system*. ISA Transaction, 36 (4), 305–311, 1998.
- Elsayed A. Orady, T. A. Osman, *Virtual reality software for robotics and manufacturing cell simulation*, Proceedings of the 21st International Conference on Computers and Industrial Engineering, October, 1997, pp. 87-90.
- Hibnio, H., Inukai, T., and Fukuda, Y., *Efficient manufacturing system implementation based on combination between real and virtual factory*.

- International Journal of Production Research, 44 (18), 3897–3915, 2006.
- H. T. Park, J. G. Kwak, G. N. Wang, and S. C. Park., *Plant model generation for PLC simulation*, International Journal of Production Research, 48(5), 1517-1529, 2010.
- Jang, J., Koo, P.H., and Nof, S.Y., *Application of design and control tools in a multirobot cell*. Computers and Industrial Engineering, 32 (1), 89–100, 1997.
- Klingstam, P. and Gullander, P., *Overview of simulation tools for computer-aided production engineering*. Computers in Industry, 38 (2), 173–186, 1999.
- Sang C. Park, Chang Mok Park, and Gi-Nam Wang, *A PLC programming environment based on a virtual plant*, International Journal of Advanced Manufacturing Technology, 2008, 39(11-12):1262-1270
- Xu XuSong, Xiong HongBin, *AS/RS real time simulation and control integrated system research*, International conference on management of logistics and supply chain, 686-693, 2006.
- Zeigler, B.P., *Multifaceted modeling and discrete event simulation*. New York: Academic Press, 1984.

AUTHOR BIOGRAPHIES

MIN SUK. KO was born in Korea, and went to the Ajou University of Suwon, where he studied Industrial Information System Engineering. He is a PH.D candidate student in Modeling & Simulation Lab, the Department of IE. His research interests include geometric engineering knowledge management, discrete event system simulation, modeling & simulation. His e-mail address is : sebastianminsuk@gmail.com.



DAE SOON. CHANG was born in Korea, and went to the Ajou University of Suwon, where he studied Industrial Information System Engineering. He is a graduated student in Modeling & Simulation Lab, the Department of IE. His research interests include discrete event system simulation, modeling & simulation. His e-mail address is : webmacome@ajou.ac.kr.



GI NAM. WANG Is a professor in the Department of Industrial and Information System Engineering at Ajou University. His e-mail address is : gnwang@ajou.ac.kr and His Web-Page can be found at: <http://udmtek.co.kr>.



SANG CHUL. PARK. Is a professor in the Department of Industrial and Information System Engineering at Ajou University. Before joining Ajou, he worked for Daimler- Chrysler Corp. and CubicTek Co., developing commercial and in-house CAD/CAM/CAPP software systems. His research interests include geometric engineering knowledge management, and discrete event system simulation. His e-mail address is : scpark@ajou.ac.kr and his Web-Page can be found at : <http://ie.ajou.ac.kr>.

