Journal of Computational Design and Engineering Vol. 1, No.3 (2014) 213~222
www.jcde.org

Survey on the virtual commissioning of manufacturing systems

Chi G. Lee[1] and Sang C. Park[2,*]

[1] *Department of Mechanical and Industrial Engineering, Toronto University, 5 King's College Road, Toronto, Ontario, M5S 3G8, Canada*
[2] *Department of Industrial Engineering, Ajou University, San 5, Woncheon-dong, Yeongtong-gu, Suwon, Korea*

(Manuscript Received May 1, 2014; Revised May 28, 2014; Accepted June 2, 2014)

Abstract

This paper reviews and identifies issues in the application of virtual commissioning technology for automated manufacturing systems. While the real commissioning of a manufacturing system involves a real plant system and a real controller, the virtual commissioning deals with a virtual plant model and a real controller. The expected benefits of virtual commissioning are the reduction of debugging and correction efforts during the subsequent real commissioning stage. However, it requires a virtual plant model and hence still requires significant amount time and efforts. Two main issues are identified, the physical model construction of a virtual device, and the logical model construction of a virtual device. This paper reviews the current literature related to the two issues and proposes future research directions to achieve the full utilization of virtual commissioning technology.

Keywords: Virtual commissioning; Virtual plant model; Virtual device model; DEVS; PLC simulation

---

1. Introduction

The ability to design good products has always been an important factor for the success of a manufacturer in the market. Good products, however, do not necessarily lead to high profit, which is essential for the manufacturer to sustain. A product can remain profitable only if it is produced with the cost less than the price. While the price is subject to the market mechanism, the cost is more under the control of the manufacturer and can be reduced by improving the efficiency of manufacturing system. Modern manufacturing system is highly integrated and consists of automated workstations. Workstation may have robots with tool-changing capabilities, handling systems, storage systems, and computer control system [1-3]. Since manufacturing system requires a heavy investment, a manufacturing system has to be designed so that the long-term profits should remain positive.

Generally, manufacturing systems are dynamic systems and the state changes coincide with the occurrence of various events, thereby exhibiting the characteristics of a discrete event system. The discrete event simulation is among the most popular approaches to the verification of a manufacturing system and has been a powerful tool for calculating utilization statistics, identifying bottlenecks, tracking scheduling errors, and even for creating manufacturing schedules [4-6]. Consequently, various simulation languages have been developed and used in academia and industry alike. However, there are limitations due to the high level of abstraction of the simulation models [7]. As shown in Figure 1, real manufacturing systems are electrically controlled by low level control programs involving sensors and actuators [8, 9], but conventional simulation models describe the dynamic behaviors of the manufacturing system with high level scripts.

If manufacturers are to remain competitive in an ever changing marketplace, they have to continuously improve both the products and the production systems [10]. Thus, an efficient prototyping environment for production systems is crucial, which leads to the notion of virtual manufacturing system (virtual commissioning), a computer based environment to simulate individual manufacturing processes [6, 12-18].

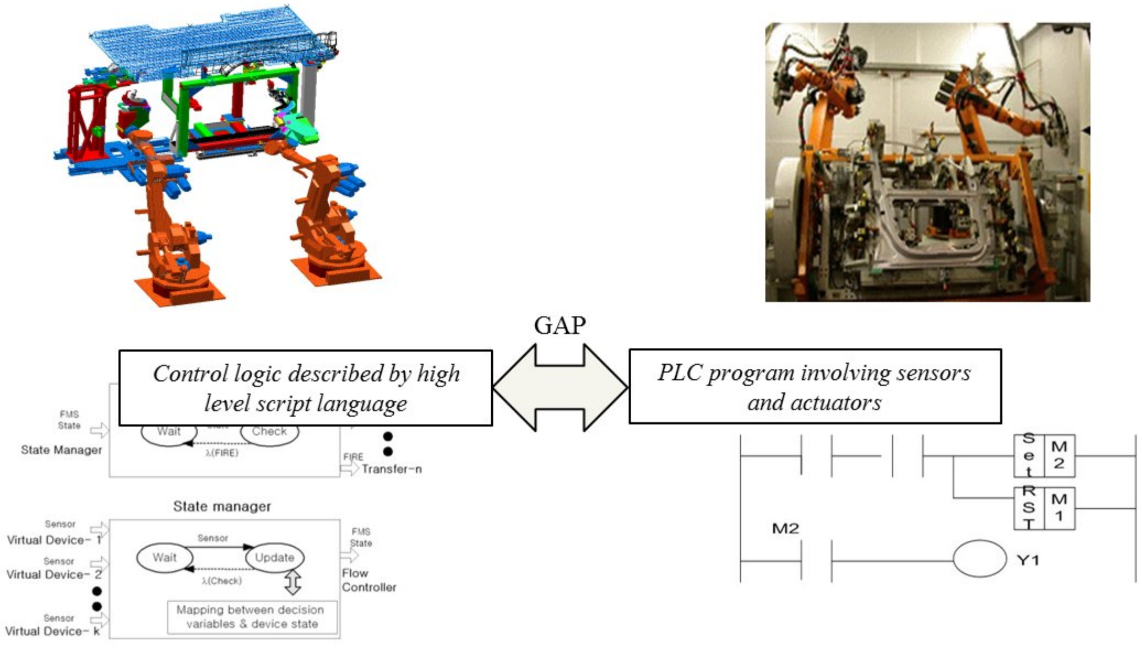
Virtual commissioning enables the full verification of a manufacturing system by performing a simulation involving a virtual plant and a real controller [19-26]. This requires the virtual plant model to be fully described at the level of sensors and actuators.

Without virtual commissioning, a manufacturing system will have to be stabilized solely by real commissioning with real plants and real controllers, which is very expensive and time consuming. Therefore, virtual commissioning is to identify and address design flaws and operational faults without real plant nor controllers so that a significant savings can be achieved in the actual implementation of the manufacturing system. A recent study [21] showed the positive effect of

---

[*] Corresponding author. Tel.: +82-31-219-2656
E-mail address: scpark@ajou.ac.kr

doi: 10.7315/JCDE.2014.021



(a) Conventional simulation model (b) Real production system

Figure 1. Gap between a simulation model and a real production system.

virtual commissioning on the error rate during real commissioning. They showed a reduction of real commissioning time by 75%, resulting from enhanced quality of the manufacturing system at the start of real commissioning.

As shown in Figure 2, there can be four configurations in commissioning: (1) real commissioning involving a real plant and a real controller; (2) virtual commissioning (hardware-in-the-loop commissioning) involving a virtual plant and a real controller as shown in Figure 3(a); (3) reality-in-the-loop commissioning involving a real plant and a virtual controller; and (4) constructive commissioning involving a virtual plant a virtual controller as shown in Figure 3(b). Engineers often focus on the virtual commissioning and the constructive commissioning requiring a virtual plant instead of a real plant. In practice, the application of the virtual commissioning has been limited. Virtual commissioning has been traditionally applied to small-size manufacturing systems such as manufacturing cells until recently. The increasing computing technology, however, allows virtual commissioning technology to large-scale manufacturing system such as manufacturing lines and factories.

Most of automated manufacturing systems are controlled by a PLC (Programmable Logic Controller), which is currently the most suitable and widely employed industrial control technology [9, 27-32].

PLCs emulate the behavior of an electric ladder diagram.

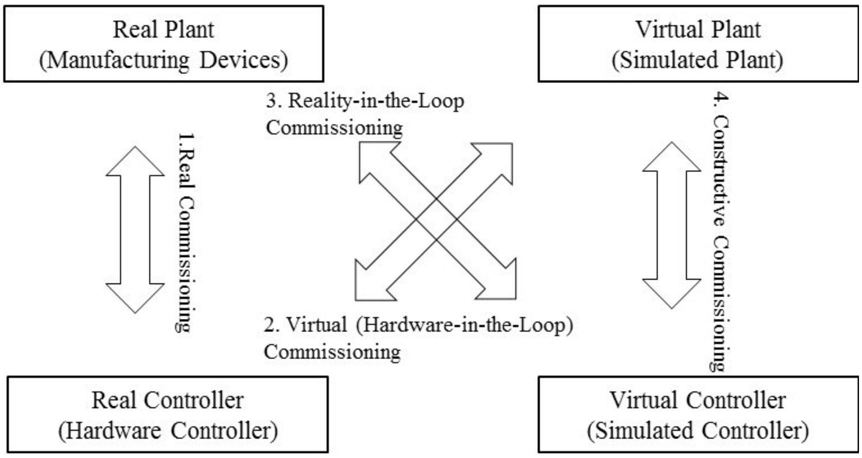
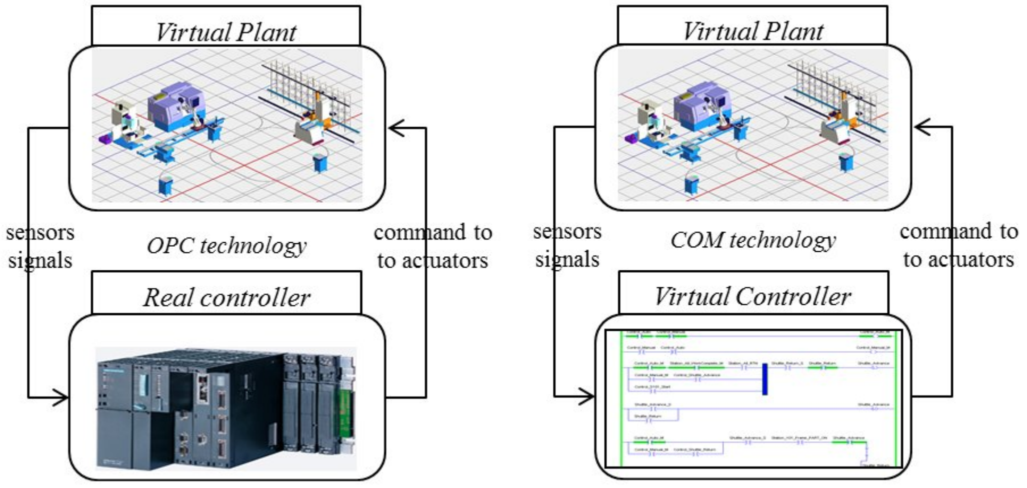


Figure 2. Commissioning configurations of a manufacturing system.



(a) Virtual (Hardware-in-the-Loop) Commissioning (b) Constructive Commissioning

Figure 3. Virtual and constructive commissioning.

As they are sequential machines, to emulate the workings of parallel circuits that respond instantaneously, PLCs use an input/output symbol table and a scanning cycle. The execution of a program in a PLC requires the continuous execution of a scanning cycle. The program scan solves the Boolean logic related to the information in the input table with that in the output and internal relay tables. In addition, the information in the output and internal relay tables is updated during the program scan. In a PLC, this Boolean logic is typically represented using a graphical language known as a ladder diagram.

The objective of this paper is to review key issues and the existing literature of virtual commissioning, and to suggest the directions for future research. The paper is organized as follows. Section 2 discusses the key issues of virtual commission. Section 3 addresses the issues of physical device modeling, whereas Section 4 deals with the logical device modeling. Finally, Section 5 concludes the paper.

2. Key issues in virtual commissioning

Since virtual commissioning requires a virtual plant con-

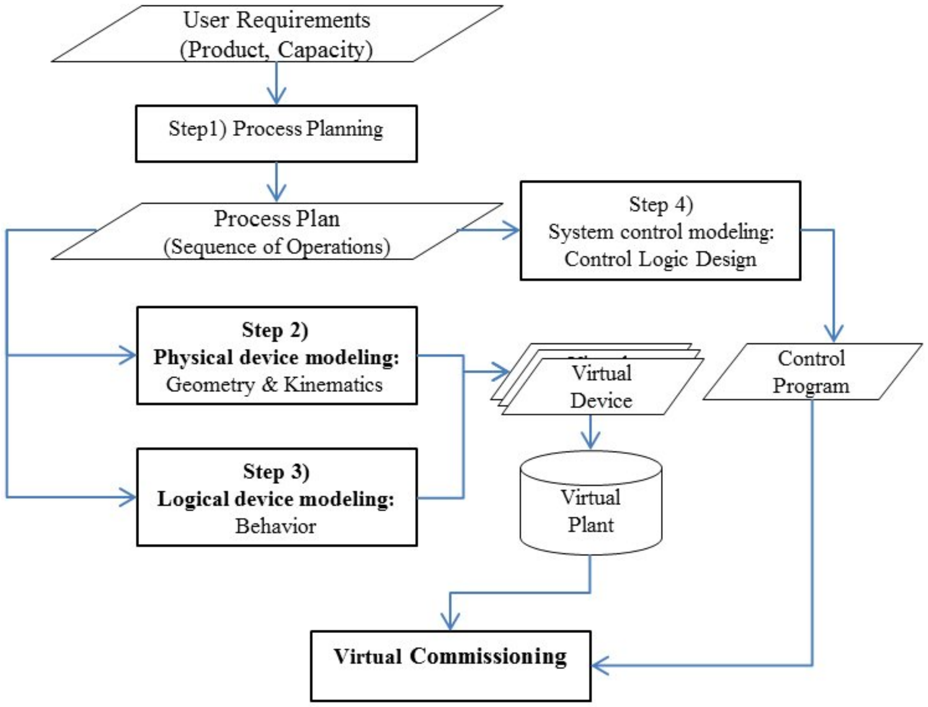


Figure 4. Concurrent design procedure for virtual commissioning.

sisting of virtual devices, the design procedure of a manufacturing system needs to be fully implemented. Ko et al. proposed a concurrent design procedure for virtual commissioning [32], as shown in Figure 4. The proposed design procedure consists of 4 major steps; (1) Process planning, (2) Physical device modeling, (3) Logical device modeling, and (4) System control modeling. In the first step, process engineers prepare a process plan (SOP, a sequence of operations) by identifying effective manufacturing processes and devices, which can economically produce the intended products. Once a process plan (SOP) is prepared, the other three steps can be addressed simultaneously.

While Step 1 and 4 are part of the conventional design procedure, Step 2 and 3 are to support virtual commissioning. Step 2 and Step 3 model the physical aspect of a virtual device and a logical its relationships (logical aspects) with other devices or PL aspect of a virtual device, respectively. Since virtual devices need to communicate with a real controller, the virtual device should behave in the same way as the real device. A virtual device needs to maintain C programs as well as the inherent attributes (physical aspects) of the device, such as the kinematics and geometric shape. While solid modeling technology is usually adapted for the modeling of the physical aspects of a virtual device, Zeigler's DEVS (Discrete Event Systems Specifications) formalism [33, 34] is usually employed for the modeling of the logical aspects of a virtual device. DEVS formalism supports the specification of discrete event models in a hierarchical, modular manner. The semantics of the formalism are highly compatible with object-oriented specifications for simulation models. Within the DEVS formalism, one must specify two types of submodels: (1) the atomic model, the basic models from which larger models are built; and (2) the coupled model, how atomic models are connected in a hierarchical manner. Formally, an atomic model M is specified by a 7-tuple:

$M = < X, S, Y, \delta_{int}, \delta_{ext}, \lambda, t_a >$
X: input events set;
S: sequential states set ;
Y: output events set;
δ_{int}: $S \rightarrow S$: internal transition function;
δ_{ext}: $Q * X \rightarrow S$: external transition function
$Q = \{(s,e) | s \in S, 0 \leq e \leq t_a(s)\}$: total state of M;
λ: $S \rightarrow Y$: output function;
t_a: $S \rightarrow Real$: time advance function.

The four elements in the 7-tuple, namely δ_{int}, δ_{ext}, λ, and t_a, are called the characteristic functions of an atomic model. The second form of the model, termed a coupled model, shows a method for coupling several component models together to form a new model. Formally, a couple model DN is defined as:

$DN = < X, Y, M, EIC, EOC, IC, SELECT >$
X: input events set;
Y: output events set;
M: set of all component models in DEVS;
$EIC \subseteq DN.IN * M.IN$: external input coupling relation;
$EOC \subseteq M.OUT * DN.OUT$: external output coupling relation;
$IC \subseteq M.OUT * M.IN$: internal coupling relation;
$SELECT$: $2^M - \emptyset \rightarrow M$: tie-breaking selector,

Where the extensions .IN and .OUT represent the input port set and the output port set of the respective DEVS models. The implementation of plant model requires the shell part of a virtual device to be represented as an atomic model, and the entire plant model as a coupled model, including the atomic models (virtual devices) and the coupling relationships between them.

The full benefits of virtual commissioning can be achieved through an efficient method to construct a virtual plant including virtual devices. There are two issues in the construction of a virtual device: (1) physical aspect modeling of a virtual device; and (2) logical aspect modeling of a virtual

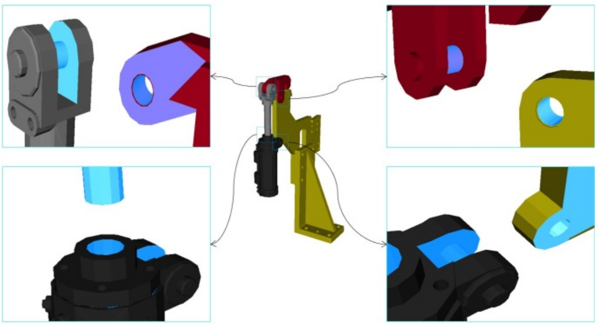


Figure 5. Physical aspect modeling: A CAD model with kinematics.

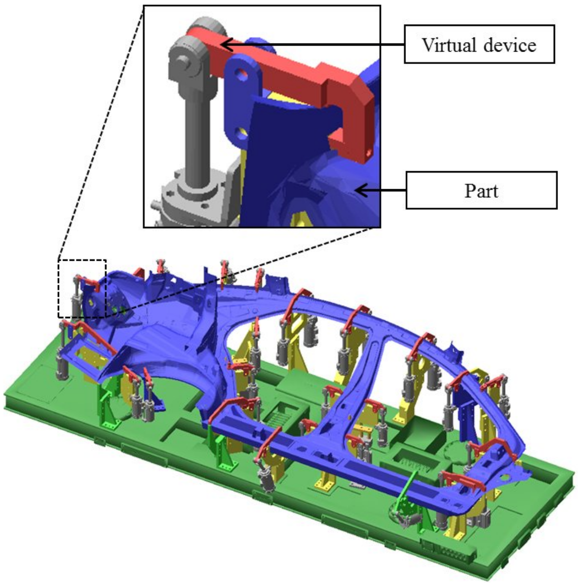
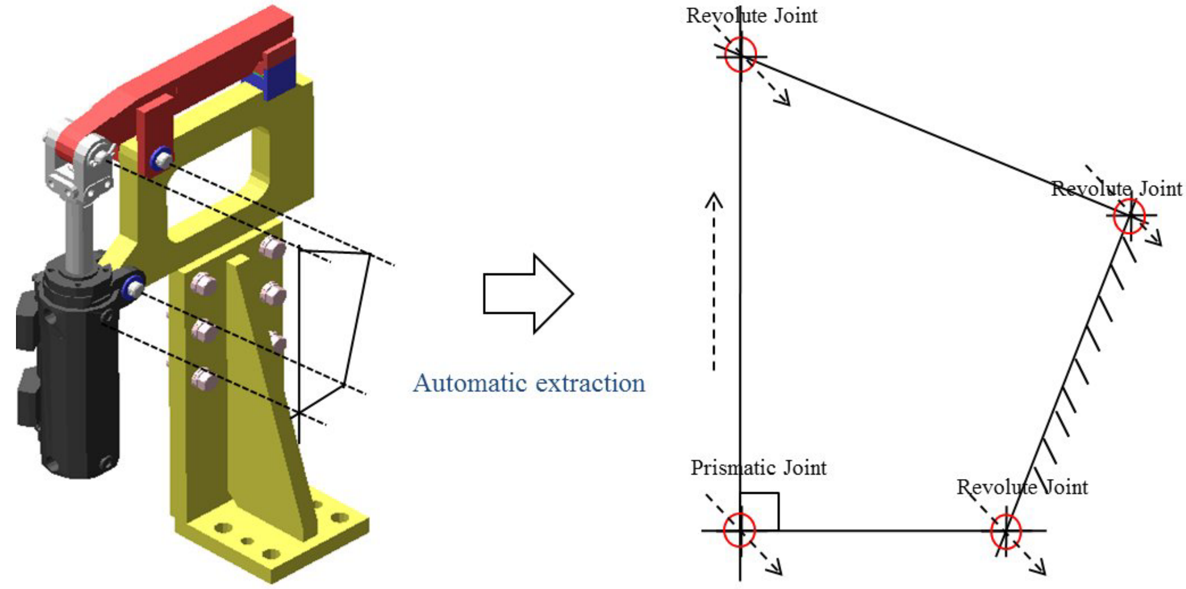


Figure 6. Geometry modeling of a virtual device for a given part.



(a) Geometric model (b) Kinetic model

Figure 7. Kinetic model extraction from a given geometric model.

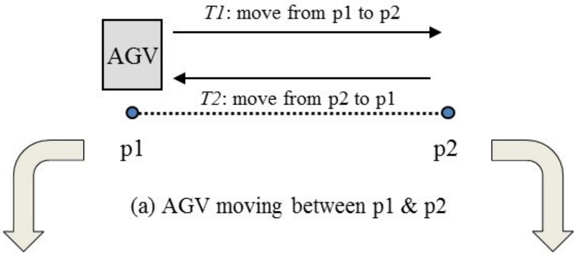
device. Section 3 will examine the two issues in more detail.

3. Physical device modeling

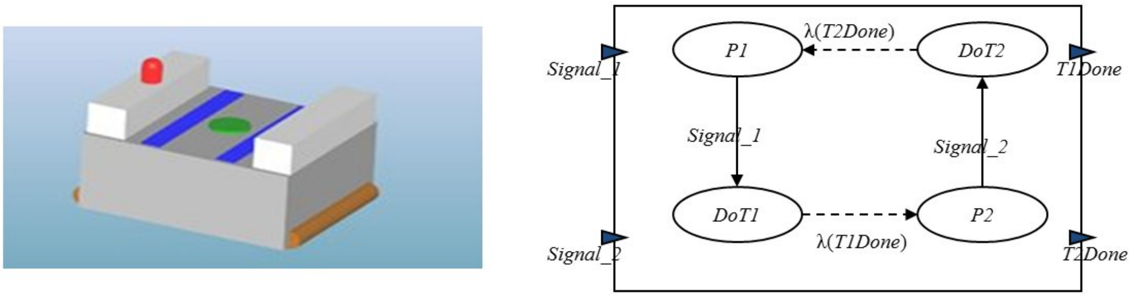
As shown in Figure 5, the physical aspect of a virtual device includes the inherent attributes of the device, which can be described as a 3D CAD model with kinematics [35]. One of the major obstacles of the virtual commissioning is the excessive time and efforts for the construction of virtual device models. A virtual plant consists of various virtual devices such as robots, conveyors, fixtures, machining and assembly tools. Many of these devices need both a geometric model and a kinetic model, as shown in Figure 5. For the geometric modeling of a virtual device, the CSG (constructive solid geometry) modeling scheme has been employed. In the CSG modeling scheme, a user can interactively construct a solid model by combining various primitives, such as cylinders, spheres, boxes and cones [10]. To construct a kinetic model of a virtual device, it is necessary to define the moving joints and the attributes of each joint. Since one of the major roles of manufacturing devices is to handle (locating, holding, and supporting) parts during various operations, the geometric model of a virtual device should be designed with the consideration of the geometry of the given part [36-41], as shown in Figure 6. There have been many studies on the geometric modeling of a virtual device for a given part. Asada and By used the Jacobian Matrix to model the device-part relationship in 3D space [42]. Based on the device-part relationship, the configuration of a holding device can be changed automatically depending upon the part geometry. For the verification of their methodology, they demonstrated an example, the holding of a plastic cover of an electrical appliance with complex shape. Trappey and Liu applied quadratic programming to construct the general verification model of holding devices, and discussed the time-variant stability problem with consideration of fixture force limits and directions [43]. Later, Kang et al. proposed a framework for the modeling of a virtual device [35]. The proposed framework uses two models, a geometric model and a kinetic model. The two sub-models are applied to three areas of fixture applications including locator analysis, tolerance analysis, and stability analysis. Recently, Mervyn et al. developed an evolutionary search algorithm exploring the large number of possible alternatives and suggesting an appropriate geometric design of a virtual device [39].

Although the majority of relevant literature deals with the geometric modeling of a virtual device, Chang, Ko et al. proposed a procedure for the kinetic modeling of the slider-crank mechanism, a four-axis system with three revolute and one prismatic axis [41]. They used the concept of ‘moment of inertia’, which is a measure of an object’s resistance to changes in its rotation rate, to identify the kinetic model of the slider-crank mechanism. Their algorithm extracts the kinetic model from the geometric model of a fixture to reduce the time and efforts of fixture modeling, as shown in Figure 7. Although, their algorithm works efficiently, its application is limited to the slider-crank mechanism.

For the full benefits of virtual commissioning technology, it is essential to develop an efficient methodology to construct virtual device models. As depicted earlier, a virtual device model consists of two sub-models, a geometric model and a kinetic model. While the geometric modeling of a virtual device has been given a great deal of attention, the kinetic modeling of a virtual device has rarely been brought into



(a) AGV moving between p1 & p2



(b) Physical aspect of the AGV (Solid Model) (c) Logical aspect of the AGV (DEVS model)

Figure 8. An example of a virtual device.

focus. Currently, the kinetic modeling of a virtual device is done manually, and it takes much time and effort.

4. Logical device modeling

While mechanical engineers design the physical configurations of a manufacturing system, electrical engineers are concerned with the logical behavior of the system determined by control programs. The verification of control programs (PLC programs) have long been an important issue in industry. The previous results on the control programs can be classified into two categories: (1) Verification of a given PLC program [44-50]; and (2) Generation of a dependable PLC program [51-57].

Researchers of the first group have been mainly focusing on the checking of theoretical attributes (safety, liveness, and reachability) of a control program, and developed various software tools for the verification of PLC-based systems via the use of timed automata, such as UPPAAL2k, KRONOS, Supremica and HyTech, mainly for programs written in a statement list language also termed Boolean [9]. Those software tools checks some of the theoretical attributes of a target system; however, it is not easy for users to determine whether the PLC programs actually achieve the intended control objectives.

In the second group, researchers have tried to generate dependable PLC programs by using two step approaches: (1) describe the control logic with well-organized formalisms including state diagrams, Petri nets and IDEF0; and (2) generate PLC programs form those formalisms. Although those formalisms can help the design process of control logics to some extents, it is still not possible to guarantee error-free PLC programs which are the most difficult part of the verification of a control program.

To cope with the problem, it is necessary to have a more transparent PLC programming environment helping users to recognize hidden errors, which results in the concept of virtual commissioning [7]. Since real devices communicate with PLC programs through input/output symbols, the behavior of each virtual device should be the same as that of the real device. As mentioned earlier, the logical behavioral model of a virtual device has been described with the DEVS formalism.

Figure 8 shows an example of a virtual device, an AGV (Automatic Guided Vehicle) with two tasks, *T1* (movement from *P1* to *P2*) and *T2* (movement from *P2* to *P1*). As explained in the previous section, the physical aspect of the device can be constructed with the solid modeling technology as shown in Figure 8(b). For the modeling of the logical aspect, we need to consider the tasks. Since the two tasks of the AGV should be triggered by external events, behavioral model the AGV must have two input ports, termed here as *Signal_1* and *Signal_2*, as shown in Figure 8(c). If we assume that the AGV always moves between *P1* and *P2* whenever it is triggered by external events, then the DEVS atomic model of the virtual device can be described as follows:

Shell of a virtual device: $M = < X, S, Y, \delta_{\text{int}}, \delta_{\text{ext}}, \lambda, t_a >$
$X = \{Signal\_1, Signal\_2\}$
$S = \{P1, DoT1, P2, DoT2\}$
$Y = \{T1Done, T2Done\}$
$\delta_{\text{int}}(DoT2) = P2$
$\delta_{\text{int}}(DoT2) = P1$
$\delta_{\text{ext}}(P1, Signal\_1) = DoT1$
$\delta_{\text{ext}}(P2, Signal\_2) = DoT2$
$\lambda(DoT1) = T1Done$
$\lambda(DoT2) = T2Done$

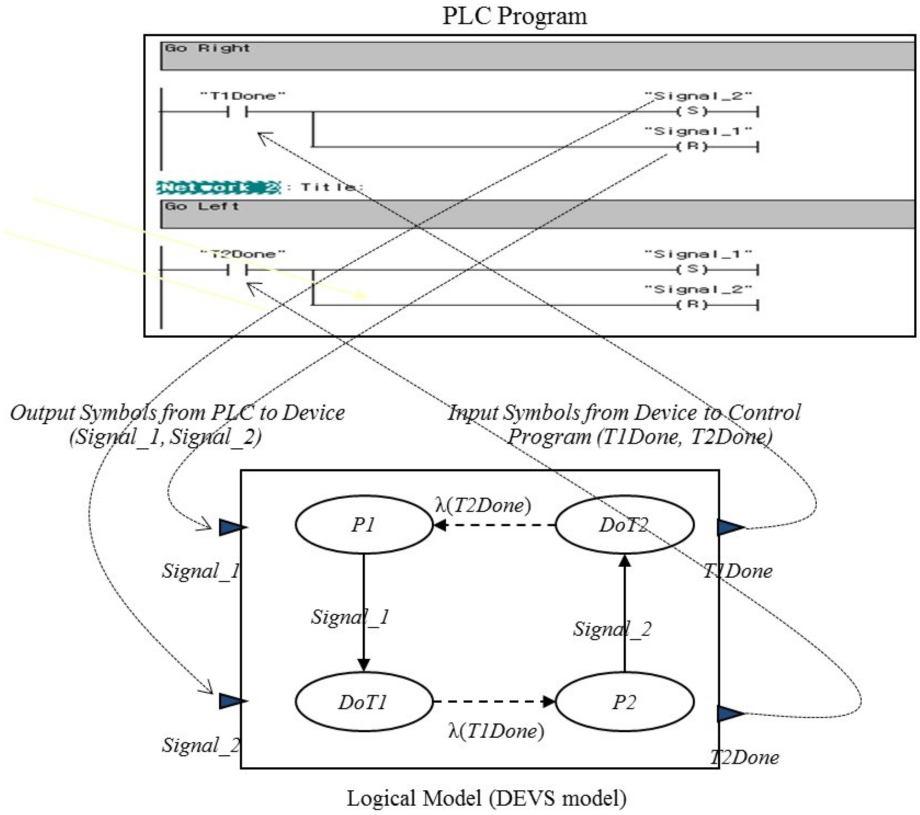


Figure 9. Input / Output symbol mapping between a control program and a logical DEVS model.

$t_a (DoT1)$ = Time_1
$t_a (DoT2)$ = Time_2.

Once the logical model of a virtual device is constructed, it is necessary to define the I/O mapping (Input / Output symbol mapping) with the corresponding control program, as shown in Figure 9. [7] A PLC program triggers the tasks of a virtual device through output symbols, and monitors the system status through input symbols.

One of the difficulties about the logical modeling of a virtual device is that the modeling procedure requires in-depth knowledge on the discrete event system modeling. To cope with the difficulties, there have been studies on methodologies assisting the construction procedure of the logical model [3, 58]. Park et al. developed a naming rule for PLC symbols so that the symbol names include sufficient information on the logical aspect of the plant model [3]. They presented an example where the logical model of a virtual device was generated from PLC symbol names. Park et al. proposed a method to generate the logical aspect of a virtual plant model using both log data (time-stamped signal history) and a PLC I/O signal table extracted from the existing production system [58]. As a result, the time and effort for the construction of a virtual plant model can be reduced.

Other than the automatic generation of logical models, Ko et al. tried to expand the DEVS modeling formalism [59]. They proposed a FMS (Flexible Manufacturing System) modeling formalism by expanding DEVS formalism. While the original DEVS formalism has only one output function, the expanded formalism has three output functions (a, d and m) to overcome the limitations of the conventional output function.

5. Conclusions

The basic idea of virtual commissioning is to connect a virtual plant with a real controller, so that engineers can detect potential errors of control programs before real commissioning stage. Although virtual commissioning can significantly reduce time and effort required at the real commissioning stage, there are obstacles to the implementation of virtual commissioning. Since a virtual plant needs to communicate with a real controller, the virtual devices should be modeled at the level of sensors and actuator, which is not easy for control engineers who do not have in-depth knowledge on modeling and simulation.

In this paper, we discussed two key issues: physical model construction of a virtual device (a solid model with kinematics), and logical model construction of a virtual device (a DEVS model). A physical model of a virtual device consists of two sub-models, a geometric model and a kinetic model. While most of existing studies focus on the geometric modeling of a virtual device for a given part geometry, the kinetic modeling can be found in some special type of kinematics such as the slider-crank mechanism. Currently, the kinetic modeling of a virtual device is performed manually, which it very time-consuming. The logical model of a virtual municates with a real controller though input/output symbols, and is usually represented in the DEVS formalism. Therefore, the logical modeling building requires a high expertise on the modeling and simulation. To overcome this challenge, some attempt to develop methodologies that ultimately assist the construction procedure of the logical model.

Although commercial products for virtual commissioning have been developed by major vendors including DELMIA and SIMENS, there are still challenges to fully utilize the virtual commission technology. In the case of the physical model construction of a virtual device, the kinetic model generation methodologies need to be developed to avoid the time consuming manual modeling of kinematics. For the logical model construction, it is necessary to consider various scenarios of virtual commissioning. For example, the virtual commissioning of an existing production system is quite different from that of a newly designing production system.

Acknowledgments

This work was partially supported by the Agency for Defense Development (UD120035JD, UD110006MD), and the Natural Research Foundation grant (2010-0021040) funded by the Ministry of Education, Science and Technology, Korea.

References

[1] Choi BK, Kim BH. New trends in CIM: virtual manufacturing systems for next generation manufacturing. Current Advances in Mechanical Design and Production VII, Proceedings of the Seventh Cairo University International MDP Conference; 2000 Feb 15-17; Cairo, Egypt; p. 425-436.

[2] Groover MP. Fundamentals of modern manufacturing. 5th ed. Wiley; 2006. 1128 p.

[3] Park HT, Kwak JG, Wang GN, Park SC. Plant model generation for PLC simulation. International Journal of Production Research. 2010; 48(5): 1517-1529.

[4] Anglani A, Grieco A, Pacella M, Tolio T. Object-oriented modeling and simulation of flexible manufacturing system: A rule-based procedure. Simulation Modeling Practice and Theory. 2002; 10(3-4): 209-234.

[5] Klingstam P, Gullander P. Overview of simulation tools for computer-aided production engineering. Computers in Industry. 1999; 38(2): 173-186.

[6] Ye L, Lin F. Virtual system simulation – A step beyond the conventional simulation. In: Proceedings of the 22nd International Conference on Computer and Industrial Engineering; 1997 Dec 20-22; Cairo, Egypt; p. 304-306.

[7] Park SC, Park CM, Wang GN. A PLC programming environment based on a virtual plant. International Journal of Advanced Manufacturing Technology. 2008; 39(11-12): 1262-1270.

[8] Barancikova G, Senesi N, Brunetti G, Dougall DJ. Application and benefits of real-time I/O simulation for PLC and PC control system. ISA Transaction. 1998; 36(4): 305-311.

[9] Manesis S, Akantziotis K. Automated synthesis of ladder automation circuits based on state-diagrams. Advances in Engineering Software. 2005; 36(4): 225-233.

[10] Park SC. A methodology for creating a virtual FMS model. Computers in industry. 2005; 56(7): 734-746.

[11] Choi SS, Jo HJ, Lee JY, Noh SD. A rule-based system for the automated creation of VR data for virtual plant review. Concurrent Engineering: Research and Applications. 2010; 18(3): 165-183.

[12] Hibnio H, Inukai T, Fukuda Y. Efficient manufacturing system implementation based on combination between real and virtual factory. International Journal of Production Research. 2006; 44(18): 3897-3915.

[13] Huang H, Yeh C. Development of a virtual factory emulator based on three-tire architecture. In: Proceedings of IEEE International Conference on Robotics and Automation; 1999 May 10-15; Detroit, MI; p. 2434-2439.

[14] Iwata K, Onosato M, Teramoto K, Osaki S. A modeling and simulation architecture for virtual manufacturing systems. CIRP. 1995; 44(1): 399-402.

[15] Lin M, Fu L, Shih T. Virtual factory – A novel testbed for an advanced flexible manufacturing system. In: IEEE International Conference on Robotics and Automation; 1999 May 10-15; Detroit MI; p. 2422-2427.

[16] Onosato M, Iwata K. Development of a virtual manufacturing system by integrating product models and factory models. CIRP. 1993; 42(1): 475-478.

[17] Xu Z, Zhao Z, Baines RW. Constructing virtual environment for manufacturing simulation. International Journal of Production Research. 2000; 38(17): 4171-4191.

[18] Auinger F, Vorderwinkler M, Buchtela G. Interface driven domain-independent modeling architecture for soft-commissioning and reality in the loop. In: Proceedings of the 1999 Winter Simulation Conference; 1999 Dec 5-8; Phoenix, AZ; p. 798-805.

[19] Drath R, Weber P, Mauser N. An evolutionary approach for the industrial introduction of virtual commissioning. In: Proceedings of IEEE International Conference on Emerging Technologies and Factory Automation; 2008 Sep 15-18; Hamburg, Germany; p. 5-8.

[20] Hoffman P, Maksoud TMA, Schuman R, Premier GC. Virtual commissioning of manufacturing systems – A review and new approaches for simplification. In: Proceedings of the 24th European Conference on Modeling and Simulation; 2010 Jun 1-4; Kuala Lumpur, Malaysia; p. 407-411.

[21] Koo LJ, Park CM, Lee CH, Park SC, Wang GN. Simulation framework for the verification of PLC programs in automobile industries. International Journal of Production Research. 2011; 49(16): 4925-4943.

[22] Kovacs GL, Kopacsi S, Nacsa J, Haidegger G, Groumpos P. Application of software reuse and object-oriented methodologies for the modeling and control of manufacturing systems. Computers in Industry. 1999; 39(3): 177-189.

[23] Lucas MR, Tilbury DM. Methods of measuring the size and complexity of PLC programs in different logic control design methodologies. International Journal of Advanced Manufacturing Technology. 2005; 26(5-6): 436-447.

[24] Pavlovic O, Pinger R, Kollmann M. Automated formal verification of PLC programs written in IL. In: Conference on Automated Deduction (CADE); 2007 Jul 17-20; Bremen, Germany; p. 152-163.

[25] Reinhart G, Wünsch, G. Economic application of virtual commissioning to mechatronic production systems. Production Engineering Research and Development. 2007; 1(4): 371-379.

[26] Bauer N, Engell S, Huuck R, Lohmann S, Lukoschus B, Manuel R, Olaf S. Verification of PLC programs given as sequential function charts. Integration of Software Specification Techniques for Applications in Engineering, Lecture Notes in Computer Science. 2004; 3147: 517-540.

[27] Chuang CP, Lan X, Chen JC. A systematic procedure for designing state combination circuits in PLCs. Journal of Industrial Technology. 1999; 15(3): 2-5.

[28] Han KH, Choi SH, Park JW, Lee JW. Rapid virtual prototyping and operational monitoring of PLC-based control system. International Journal of Mathematics and Computers in Simulation. 2011; 5(3): 198-205.

[29] Jang J, Koo PH, Nof SY. Application of design and control tools in a multirobot cell. Computers and Industrial Engineering. 1997; 32(1): 89-100.

[30] Ko MS, Park SC, Chang MH. Control level simulation of an AS/RS in the automotive industry. The International Journal of Concurrent Engineering: Research and Applications. 2013; 21(1): 13-25.

[31] Rullan A. Programmable logic controllers versus personal computers for process control. Computers and Industrial Engineering. 1997; 33(1-2): 421-424.

[32] Ko MS, Ahn E, Park SC. A concurrent design methodology of a production system for virtual commissioning. The International Journal of Concurrent Engineering: Research and Applications. 2013; 21(2): 129-140.

[33] Kim TG. DEVSIM++ User's Manual. Department of Electrical Engineering. KAIST, Korea: 1994.

[34] Zeigler BP. Multifaceted modeling and discrete event simulation. New York (NY): Academic Press Inc.; 1984. 372 p.

[35] Kang Y, Rong Y, Yang JA. Geometric and kinetic model based computer-aided fixture design verification. Journal of Computing and Information Science in Engineering. 2003; 3(3): 187-199.

[36] Chou YC, Chandru V, Barash MM. A mathematical approach to automatic configuration of machining fixtures: Analysis and synthesis. Journal of Manufacturing Science and Engineering. 1989; 111(4): 229-306.

[37] Choubey AM, Prakash, Chan FTS, Riwari MK. Solving a fixture configuration design problem using genetic algorithm with learning automata approach. International Journal of Production Research. 2005; 43(22): 4721-4743.

[38] Mervyn F, Kumar AS, Nee AYC. Automated synthesis of modular fixture design using and evolutionary search algorithm. International Journal of Production Research. 2005; 43(23): 5047-5070.

[39] Mervyn F, Kumar AS, Nee AYC. Fixture design information support for integrated design and manufacturing. International Journal of Production Research. 2006; 44(11): 2205-2219.

[40] Pehlivan S, Summers JD. A review of computer-aided fixture design with respect to information support requirements. International Journal of Production Research. 2008; 46(4): 929-947.

[41] Chang MH, Ko MS, Park SC. Fixture modeling for an automotive assembly line. International Journal of Production Research. 2011; 49(15): 4593-4604.

[42] Asada H, By A. Kinematic analysis of workpart fixturing for flexible assembly with automatically reconfigurable fixtures. IEEE Journal of Robotics and Automation. 1985; 1(1): 86-94.

[43] Trappey AC, Liu CK. An automatic workholding verification system. Robotics and Computer-Intergrated Manufacturing. 1992; 9(4-5): 321-326.

[44] Alexandre D, Gerd B, Kim GL, Wang Y. A tool architecture for the next generation of UPPAAL. In: Proceeding of Formal Methods at the Crossroads; 2003 Mar 18-20; Lisbon, Portugal.

[45] David R, Alla H. Petri nets for modeling of dynamics systems. Automatica. 1994; 30(2): 175-202.

[46] Gourcuff V, De Smet O, Faure J. Efficient representation for formal verification of PLC programs. In: Proceedings of the 8th International Workshop on Discrete Event Systems; 2006 July 10-12; Ann Arbor, MI; p. 182-187.

[47] Kim GL, Paul P, Wang Y. UPPAAL in a nutshell. International Journal on Software Tools for Technology Transfer. 1997; 1(1-2): 134-152.

[48] Tapken J. Interactive and compilative simulation of PLCPLC-automata. Simuation in Industry, ESS'97 SCS; 1997: p. 552-556.

[49] Twiss E, Zhou MC. Design of industrial automated system via relay ladder logic programming and Petri Nets. IEEE Transactions on – Systems, Man, and Cybernetics, Part C: Applications and Reviews. 1998; 28(1): 137-150.

[50] Uzam M, Jones AH. Discrete event control system design using automation Petri Nets and their ladder diagram implementation. International Journal of Advanced Manufacturing Systems. 1998; 14(10): 716-728.

[51] Lee JS, Hsu PL. A PLC-based design for the sequence controller in discrete event systems. In: Proceedings of the 2000 IEEE International Conference on Control Applications; 2000 Sep 25-27; Anchorage, AK; p. 929-934.

[52] Fray G. Automatic implementation of Petri net based control algorithms on PLCs. In: Proceedings of the American control conference ACC 2000; 2000 Jun 28-30; Chicago, IL; p.2819-2823.

[53] Jafari MA, Boucher TO. A rule-based system for generating a ladder logic control program from a high level system model. Journal of Intelligent Manufacturing Systems. 1994; 5(2):103-120.

[54] Richardsson J, Fabian M. Automatic generation of PLC program for control of flexible manufacturing cells. In: Proceedings of the IEEE Conference merging Technologies and Factory Automation; 2003 Sep 16-19; Lisbon, Portugal; p. 337-344.

[55] Sacha K. Automatic Code Generation for PLC Controllers. Computer Safety, Reliability, and Security Lecture Notes in Computer Science. 2005; 3688: 303-316.

[56] Estévez E, Marcos M, Orive D. Automatic generation of PLC automation projects from component-based models. The International Journal of Advanced Manufacturing Technology. 2007; 35(5-6): 527-540.

[57] Park SC, Ko MS, Chang M. A reverse engineering approach to generate a virtual plant for PLC simulation. International Journal of Advanced Manufacturing Technology. 2013; 69(9-12): 2459-2469.

[58] Ko M, Park SC, Choi JJ, Chang M. New modeling formalism for control programs of flexible manufacturing systems. International Journal of Production Research. 2013; 51(6): 1668-1679.