Computers & Industrial Engineering 60 (2011) 138–147

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

A data-driven generic simulation model for logistics-embedded assembly manufacturing lines ☆

Juyoung Wy [a], Sangwon Jeong [a], Byung-In Kim [a,*], Junhyuk Park [a], Jaejoon Shin [a], Hyunjoong Yoon [b], Sujeong Lee [c]

[a] Department of Industrial and Management Engineering, Pohang University of Science and Technology, Pohang, Kyungbuk, 790-784, Republic of Korea
[b] School of Mechanical and Automotive Engineering, Catholic University of Daegu, Daegu, 712-902, Republic of Korea
[c] Mechatronics & Manufacturing Center, Samsung Electronics Co., Suwon, Kyunggi, 441-742, Republic of Korea

ARTICLE INFO

Article history:
Received 28 April 2009
Received in revised form 13 September 2010
Accepted 18 October 2010
Available online 23 October 2010

Keywords:
Generic simulation
Logistics
Assembly lines

ABSTRACT

Simulation has been used to evaluate various aspects of manufacturing systems. However, building a simulation model of a manufacturing system is time-consuming and error-prone because of the complexity of the systems. This paper introduces a generic simulation modeling framework to reduce the simulation model build time. The framework consists of layout modeling software and a data-driven generic simulation model. The generic simulation model was developed considering the processing as well as the logistics aspects of assembly manufacturing systems. The framework can be used to quickly develop an integrated simulation model of the production schedule, operation processes and logistics of a system. The framework was validated by developing simulation models of cellular and conveyor manufacturing systems.

1. Introduction

Simulations have been used for various purposes in manufacturing. They are used for strategic capacity planning, automation systems design, manufacturing process validation, and evaluation of various manufacturing execution scenarios. They can be used to analyze how system performance is affected by the layout configuration, the number of material handling resources used, the resource operating policies, and the usage of different types of material handling systems (MHS). A typical example simulation study can be found in Mackulak and Savory (2001).

However, a simulation model that encapsulates the details of a manufacturing system is very time-consuming to build and debug. Engineers and researchers typically build simulation models with their own style based on their own experiences, without consideration of reusability, and the models are sometimes error-prone. The severe competitive pressures in the manufacturing industry require much quicker turnarounds for simulation projects. In order to cope with this challenge, we recently proposed a data-driven generic simulation modeling framework for semiconductor fabrication lines (Kim et al., 2009). As Mackulak, Lawrence, and Colvin (1998) stated, using a generic simulation model has two primary advantages: quick model construction and the reuse of more accurate (bug-free) models.

This paper applies the framework to assembly manufacturing lines with development of a data-driven generic simulation model for the lines. The generic model described in this paper was developed with in-depth consideration of the logistics aspects, i.e., the material flows. The data representation scheme for the generic simulation model and the logical flow of the model are presented.

The remainder of this paper is organized as follows: A brief literature review of the simulation program generator and generic simulation models is provided in Section 2. After our proposed generic modeling framework is briefly described in Section 3, the generic simulation model is presented for logistics-embedded assembly manufacturing lines in Section 4. Section 5 discusses the model verification and application of the proposed framework. Section 6 contains concluding remarks.

2. Literature review

Various approaches have been used to reduce the simulation model build time. A simulation program generator can automatically generate simulation models in the target simulation language based on user input. Mathewson (1984) defines a simulation program generator as “an interactive software tool that translates the logic of a model described in a relatively general symbolism into the code of a simulation language.” Pidd (1992) defines a data-driven generic simulation model as “one which is designed

☆ This manuscript was processed by Area Editor Paul Savory.

* Corresponding author. Tel.: +82 54 279 2371; fax: +82 54 279 2870.
E-mail address: bkim@postech.ac.kr (B.-I. Kim).

0360-8352/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2010.10.011

to apply to a range of systems which have structural similarities." Such a model should be generic to a target domain and should be able to simulate different instances of systems in the domain. In our opinion, the difference between the approaches is that while a simulation program generator makes a different simulation code for each instance of a system, a data-driven generic simulation model is used in a range of instances without change of code. However, both approaches can be considered to be similar because every simulation program generator should have a data-driven generic simulation model within its code generator.

Mathewson (1984) introduces a simulation program generator, called DRAFT. DRAFT generates SIMSCRIPT simulation language code from a descriptive input model expressed with an entity cycle diagram of the system of interest. Pidd (1992) proposes a set of modules as the components of a data-driven simulator: a simulation model, simulation logic, libraries, a model configurator, a file handler, an experimental frame, and a report generator. Our simulator is a data-driven generic simulator since it can be applied to various manufacturing line simulations and, in an ideal case, instance of a system may be specified by the input data only, without any programming.

Ozdemirel and Mackulak (1993) and Lung, Cochran, Mackulak, and Urban (1994) describe a module-based generic model generator for discrete manufacturing systems. A group technology classification scheme is used to identify systematically the common model features for the generator. The 14 generic modules corresponding to the identified features are then developed in SIMAN code. Based on the generic modules, they built an intelligent simulation environment called IntelliSIM. Savory, Mackulak, and Cochran (1994) create a tutorial on the environment. Mackulak et al. (1998) develop a generic simulation model for semiconductor-automated material handling systems evaluation. Using the generic model, the model building and analysis time were reduced from over 6 weeks to less than 1 week.

McLean, Jones, Lee, and Riddick (2002) provide an overview of the development of a prototype generic machine shop simulator for small machine shop manufacturers underway at the National Institute of Standards and Technology (NIST). They used the Arena simulation language and Microsoft Visual Basic extensions to build a prototype simulator. Lee and McLean (2006) discuss a neutral data model developed at NIST. The model has been developed to represent machine shop data and exchange it between a simulation and other applications.

Aytug and Dogan (1998) propose a framework for defining Kanban-controlled pull manufacturing systems and an automatic simulation model generator for the systems. The generator uses work center, process, and customer models, as well as layout and experiment data models. They do not consider any aspects of material handling devices such as transporters and conveyors.

Son, Jones, and Wysk (2000) generate an automatic simulation model from a neutral description of a job shop system and a data instance. An executable ProModel simulation model was automatically generated from the given shop floor station data, product data, process flow data, and job arrival information. Son, Jones, and Wysk (2003) extend their previous work and automatically generate an Arena model from neutral data for the same job shop system. Their generators assume that the process route of a product is fixed and also do not include time distributions and material transporters.

Lee, Cho, and Jung (2000) present a methodology to generate a WITNESS simulation model from graph-based process plans and the resource configuration. They convert the process plans to a machine-centered graph which is then extended to a transport-tending part-routing graph which specifies material handling activities between machines. Material handling devices including a conveyor, an automated guided vehicle (AGV), and robots in a job shop system were explicitly considered in their case study model.

Tyan, Du, Chen, and Chang (2004) propose an integrated modeling framework for analyzing the combined effects of machine dispatching rules and material handling vehicle dispatching rules in a fully automated semiconductor fab. The framework consists of the input data and parameters, the simulation model, and the presentation mechanism. Several other works related to the semiconductor industry are reviewed in Kim et al. (2009).

Although several previous studies attempted to develop simulation program generators or generic simulation models for manufacturing lines as summarized above, no works fully consider the practical aspects of material handling logistics in assembly lines such as parts feeding, cart circulation, kitting of parts, and grouping of parts or sub-assemblies. The data-driven generic simulation approach of this paper has the following unique features.

(a) Accurate computer aided design (CAD) physical layout data is converted into the simulation model. Layout data includes the locations of workstations in a line and storage areas of materials as well as the material handling path networks. Although this feature can significantly reduce the initial modeling time, it is not commonly used in other works.
(b) The generic simulation model is designed for assembly lines. It covers cellular assembly lines as well as traditional conveyor assembly lines.
(c) The production process and material handling activities are integrated in a generic model. Not many works take the integrated perspective. The model incorporates part feeding, cart circulation, grouping of parts or sub-assemblies, and the behavior of material handling operators.
(d) AutoMod simulation software (Brooks Automation Inc., 2006) is used as a basic simulation language. Although some research works on generic simulation and simulation program generators use ProModel or Arena as the basic simulation software, it is hard to find a study using AutoMod. Note that Samsung Electronics Co. uses AutoMod as its quasi-standard simulation software.
(e) The *work lists* of AutoMod, which are the lists of possible job request locations of material handling resources, do not need to be specified manually. Many AutoMod users think that the lists should be set manually, which is a big hurdle for the development of a generic simulation model. Our generic model, however, generates the lists automatically.

3. Generic modeling framework

The proposed modeling framework is composed of AutoLay™ and AutoLogic-Assembly™, as shown in Fig. 1. AutoLay stands for AutoCAD to AutoMod Layout tool and AutoLogic-Assembly stands for AutoMod generic logic model for assembly lines. AutoLay has been developed to convert layout data in the CAD file format into simulation models. AutoLogic-Assembly is a generic AutoMod simulation model developed to consider generalized logistics-embedded assembly lines. AutoLay and AutoLogic-Assembly are proprietary software developed by the authors.

AutoLay, implemented in Borland C++, reads plant layout files of drawing interchange format (DXF) and generates proper AutoMod simulation model components such as queues, resources, and flow paths. This software has been developed for the steps of layout drawing and defining resources in the simulation model building.

Using AutoLay, a simulation analyst can easily extract the locations of workstations, storage areas, and flow paths of material handling resources from a DXF CAD file, define the moving directions on the paths, add control points on the paths and names of the equipment, and write an AutoMod model consisting of queues,