Available online at www.sciencedirect.com
ScienceDirect
Procedia CIRP 00 (2012) 000–000
www.elsevier.com/locate/cirp

8th CIRP Conference on Intelligent Computation in Manufacturing Engineering, Gulf of Naples, Italy, 18 - 20 July 2012

ISA standard simulation model generation supported by data stored in low level controllers

G. Popovics[a,b]*, L. Monostori[a, b]

[a]*Fraunhofer Project Center on Production Management and Informatics, Computer and Automation Research Institute, Hungarian Academy of Sciences*
[b]*Faculty of Mechanical Engineering, Budapest University of Technology and Economics*
* Corresponding author. Tel.: +36 1 279 6115; *E-mail address*: popovics@sztaki.hu.

Abstract

One of the most widespread techniques to evaluate various aspects of a manufacturing system is discrete-event simulation (DES). However, building a simulation model of a manufacturing system is a difficult task and needs great resource expenditures. Automated data collection and model buildup can drastically reduce the time of the design phase as well as support model reusability. Since most of the manufacturing systems are controlled by low level controllers (e.g., PLCs, CNCs) they inherently store structure and control logic of the system to be modeled by a DES system. The paper introduces an ongoing research of PLC code processing method for automatic ISA standard simulation model generation of a conveyor system of a leading automotive factory. Results of the validation process and simulation experiments are also described through a case study.

Keywords: Programmable logic controllers; manufacturing; computer simulation; automatic recognition

1. Introduction

During the last few decades planning and control of production systems developed in parallel with computer sciences. As it is well known, one of the most important factors for a production firm is to secure the widest product variability and the minimum lead time. These requirements lead to complex production systems that run in a rapidly changing environment. To plan, monitor and control a manufacturing system it is necessary to create the digital map of the real factory and processes.

To make responsible management decisions related to the planning and control of a production system based on the information of the digital factory is always a nerve-racking task because there might be differences between the digital and the real factories. So it is essential to keep digital representation valid by following the changes occurred in the real factory. The conventional way to achieve this is to refresh it manually after any (some) changes in the real factory that is a time and resource consuming task.

Discrete event simulation (DES) is one of the most widely spread techniques to evaluate various aspects of a manufacturing system. (Banks 1998, Law and Kelton 2000, O'Rielly and Lilegdon 1999). However, on the one hand the design phase of a simulation project needs great resource expenditures. On the other hand, simulation is applied to long-term planning, design and analysis of manufacturing systems. These models are termed "throw away" or "stand-alone" models because they are seldom used after the initial plans or designs have been finalized. As opposed to the "traditional" use of simulation, Son et al. proposed in [11] that once the system design has been finalized, the simulation model that was used for evaluation also could be used as the basis for system control. In their concept simulation was created by using neutral system components, i.e., they made efforts to build simulation models for Shop Floor Control System (SFC), generated automatically.

Data needed to build simulation models of manufacturing systems are available in production databases or can be collected. Nowadays majority of the enterprises apply PLCs for process control. Subsequently, the topology and the control logic of the manufacturing system needed for a simulation model are inherently kept in these PLCs. In our understanding topology is a directed graph in which nodes represent the elements (e.g. robots, tool machines) of the manufacturing system. The edges are the connections between the elements. They exist if there is possible material flow between the elements. The direction of the edges represents the direction of the material flow. Accordingly, building material flow simulation models of a production system can be supported by data and control logic extracted from PLC programs.

The paper introduces an ongoing research of PLC program and historical data processing method that generates a structured dataset that can be used by manufacturing simulation software to automatically create and parameterize a model.

2. Automated simulation model building

As stated by Ryan and Heavey (2006) the most commonly used rule of a simulation project is the so called "40-20-40 rule". The rule states that time spent developing a simulation project can be divided as follows:

- 40% to requirements gathering,
- 20% to model translation,
- 40% to experimentation.

Several previous studies aimed at reducing the time needed by the development phase of a simulation project of a manufacturing system that highlights the importance of this topic. The time-consuming requirements gathering phase contains input data collection and preparation. Significant planning time reduction can be achieved by automating data gathering and preparation.

Several approaches have been used for automating simulation model buildup by automatic input data gathering and processing. Park et al. (2010) suggest a naming rule in PLC programs to automatically identify objects and control logic in code giving a basic data set to build simulation model. This approach needs a renaming process on PLC programs if naming rule suggested is not applied.

Bagchi et al. (2008) describe a discrete event simulator developed for daily prediction of WIP position in an operational wafer fabrication factory to support tactical decision-making. Model parameters are automatically updated using statistical analysis performed on historical event logs generated by the factory, while "snapshot" (specified later) of current status of production is generated by using the manufacturing execution system (i.e., aggregated info of PLC).

The most widely spread applications of using PLC programs for generating simulation models aims at verifying PLC programs themselves. Han et al. (2010) propose a prototyping to improve limitations of existing control logic verification methods and ladder programming. The technique proposed by them supports functionality verification of PLC program on low control level. Contrarily PLC program process method proposed by the authors is for evaluating the effects of changing PLC programs on the overall system.

Research is carried out to develop a method to reduce the time required for building simulation models. Wya et al. (2011) proposed a generic simulation modeling framework to reduce the simulation model building time. The proposed framework composed several software that contained information of layout and control logic of the modeled objects. According to this approach layout and control logic of the manufacturing system must be designed by the appropriate software.

As it was highlighted several developed method apply PLC program processing for supporting automatic simulation model building and some of them is for evaluating PLC programs themselves. Contrarily PLC program process method proposed by the authors of this paper is for evaluating the effects of changing PLC programs on the overall system.

3. Novel solution for reducing simulation model building time

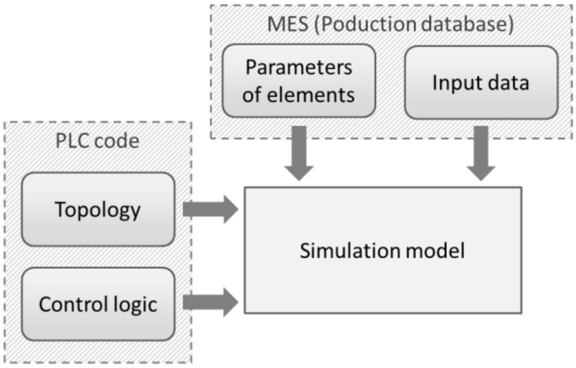

Fig. 1: Data gathering for simulation model

3.1. Data needed to build a simulation model

Simulation models related to manufacturing systems require several types of data (see Fig. 1.).

Shop floor layout provides information on the physical structure. Basically the layout identifies elements of the system, their dimensions and internal distances. Elements controlled by PLC also have unique identification, so a list of elements can be retrieved from PLC program. Relations of elements also can be extracted from PLC programs, because there are references in the code between elements that are connected. Hence topology of controlled system is incorporated in the PLC program.

Second indispensable information of manufacturing systems is the control logic of their elements. Control logic describes the response to be given on PLC's output depending on the input. The control logic consists of structured methods so variables and object's relationship can be transformed to the language of the simulation software.

It is also necessary to parameterize elements of the simulation model. Most of PLC programs do not operate with these kinds of parameters of controlled element however, AMSs usually apply Manufacturing Execution System (MES) that stores status changes of controlled elements and timestamp of state change events. Possible states and parameters of elements can be retrieved by statistical evaluation accomplished on these data as shown on Fig. 1.

3.2. Data needed to run a simulation model

The main data needed to run a simulation model is the input dataset. This set consists of all the parameters (e.g. production mix, applied dispatching rules etc.) that can be set to achieve a certain optimum of the system. After building and parameterizing a simulation model it is ready to evaluate performance of a real system in several production scenarios.

Simulation experiments can be applied to evaluate behavior of systems in situations that have never happened before. In these experiments input parameters are set to a combination that has never occurred in the real system. This type of experiments is so called "what-if-scenario" experiments.

Evaluations of the response given to cut-off scenarios are commonly used technique of testing maximum performance of a system. They are also called stress tests. In this experiments input parameters are set to a hypothetical level that leads maximum performance of the system. Detecting this set of parameters can be a difficult task in case of a complex system, so this type of evaluation might be a series of iterating experiments.

Ramp up phase of the simulation run distorts results of it so it is necessary to ignore or avoid it. In order to achieve this goal simulation run must start from a steady state. A so called snapshot of real system is created and initial states and parameters of the model are set according to states and parameters of snapshot. To analyze the behavior of a system in real time, snapshots are created periodically. This adaptation of simulation model needs real time connection with production database hence it is an on-line application of the model.

Analyzing the behavior of the system in cut-off scenarios or in "what-if-scenarios" is a way to identify the strengths and weaknesses of it. Possessing this kind of information of a production system supports the planning and managing decisions. It also helps planners to realize the relationships among the processes in the system that is a useful knowledge while planning a new production system.

3.3. Proposed methods

As mentioned above, several types of data are needed to build up a simulation model. Each type of data is generated by different methods and processes.

3.3.1. Variable and value identification of the PLC programs

The proposed method was developed by using the code of Siemens PLCs. The PLC based data acquisition was carried out after a detailed inspection of the PLC programs, in order to determine the blocks of the code, which are essential for the model building. During the exportation of these blocks a suitable data format has to be chose, which offers high level of data consistency and simple accessibility to the stored data. Considering these requirements Instruction List (AWL) programming language was chosen out as a textual export format. Since it is a low level programming language, it has a strict syntax which allows less difference in the code, which is desirable for further processing of it.

However, the file format of the exported data is a plain text, which contains no relevant information about the structure of the code and therefore it has to be included. This was accomplished by applying grammar analysis. The grammar analysis is based on a grammar which comprises the rules and class definitions concerning the PLC program. An appropriate grammar is closely related to the IEC 61131-3 standard and the analyzed PLC program therefore, it can be used to parse the PLC program and to create a structured set of data, which highlights the desired information for model building. Grammar analysis can be applied on different PLC programs and to be extended dynamically, which makes it able to be a part of automated model generation. A software named ProGrammar was used to develop the grammar.

The test PLC program as all the PLC programs consists of blocks. One block describes the behavior of one actuator (e.g. electric engine actuating the conveyor). Every block consists of two main parts:

- In the first part the logical conditions and the corresponding data operations are stored.
- The second part of the block is for function block call. The arguments belonging to the call are the input parameters which are needed to instantiate a block.

The grammar analysis results a structure called parse tree, which is a graph object that is composed of the elements defined in the grammar. According to the scope of the analysis these elements can reference various types of information: from the logical rules of operation to the logical or physical connections between the elements.

In order to extract the required information from the parse tree its further processing is needed. This can be achieved by a program, which is able to handle the parse tree and supports transferring the processed results to a simulation environment. A technical computing software (Mathematica, www.wofram.com/mathematica) is applied to process the parse tree. Mathematica is able to parse the input text by using a parsing engine of the ProGrammar by an API (application programming interface) that provides interface for external software.

In this paper a method is introduced which is purposed to reconstruct the physical topology of a conveyor system, that contains 6 dispatch stations, 3 reception stations, 24 control area and several sections for storage and ordering the pallets. The full length of the conveyor system is 5,3 km and consists of c.a. 1500 conveyor elements. Therefore, the processing of the parse tree was executed in order to acquire the information which is essential to reconstruct the connections between the elements of the conveyor system,

Since the connections of the conveyor system can be represented by a directed graph, the data which describe the connections can be stored in a matrix. Identifying all the elements of the system and their connections makes it possible to create the connection reference matrix. However, this can be difficult and requires general purpose methods, which are able to handle the code-specific differences of the input data and that is the point where the flexibility of the created grammar is fundamental.

3.3.2. Topology graph generation

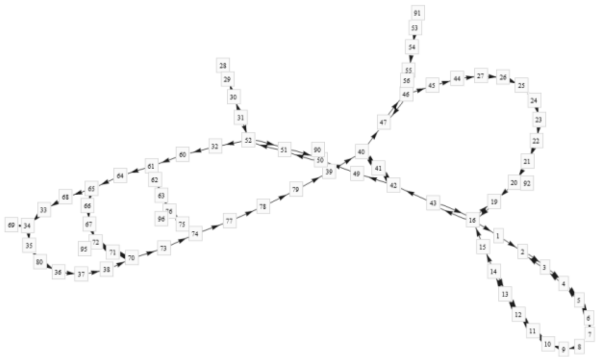

Fig. 2: Part of topology of a conveyor system

The resulted matrix of the above introduced method is visualized by using the graph plotting features of the Mathematica. The nodes of the directed graph represent the elements of the conveyor system, while the directed edges represent the connections between them (see Figure 2). Since the gathered data do not contain information about the layout of the system, the arrangement of the graph was used instead. Nevertheless, by setting the parameters of the elements properly the real operation of the system can be modelled.

Therefore, by gathering the relative coordinate data of each element, the graph can be used to provide element arrangement data to the simulation model.

Processing the parse tree results in three different data structures, which are the following.

- A list which contains all the existing elements of the conveyor system. These are the elements to be created in the simulation model.
- A connection reference matrix, which describes the connection between the elements.
- An element arrangement table, which contains the relative coordinates of the elements.

These data have to be transferred to the simulation software (in this case Tecnomatix Plant Simulation ver. 10), where the simulation model can be built up. The data transfer was carried out by using an SQL-server, which can be accessed by the built-in SQL-client of the Mathematica and the Plant Simulation as well.

Using the built-in features of Mathematica for parsing, processing and transferring the data increases the level of integration, this is essential to carry out the automated model generation.

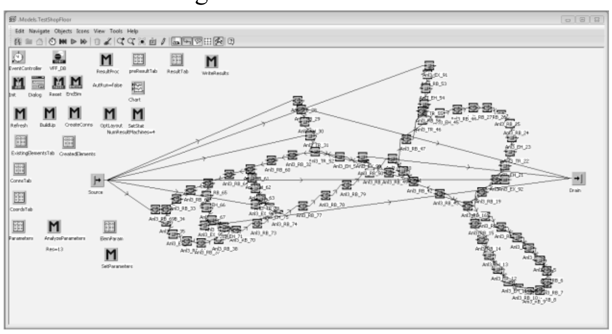

Fig. 3: Screenshot of test run executed on simulation model

As Mathematica is suited to call the parsing method of ProGrammar using a predefined command, it can be fed with data of connection among the parts of the conveyor system. After a conversion of data, the topology of the examined system can be generated by a graph creating command. The data received from ProGrammar can also be converted to a connection table form. The simulation software and its internal connection tables and methods are applied to generate the simulation object instances and run the simulation (see Fig. 3).

Depending on the aims of the simulation evaluation the representation level of the real system might differ from the level of the PLCs that are used as the base of the system topology creation. Certain groups of the mapped objects can be contracted in the case the type of the evaluation allows high level representation. In the examined industrial case the conveyor elements between two junctions can be extracted – considering the capacity of the section – because they run in a synchronized way.

In cases when contraction of the elements is feasible it is necessary to define the rules of grouping that requires human knowledge. The defined rules can be implemented in an automatic grouping function of the simulation element generating method. This way the simulation evaluation can be fulfilled on a higher level than the level of the PLCs from that codes are extracted.

3.3.3. Standardization of the simulation model

One of the main goals of the introduced research is to reduce the planning and building phases of a discrete event simulation evaluation project of any automatically controlled manufacturing system. To reach this aim it is essential to create a standard set of simulation elements and to store the relevant information in a standardized format. As the standards introduced by ISA (International Society of Automation) are one of the most widespread and most commonly used standards it is obvious to apply them.

The simulation objects and the storage of relevant data applied in the research of the paper were designed applying ISA-95 standard that is the international standard for the integration of enterprise and control systems (www.isa-95.com). This standard consists of models and terminology that determine the information to be exchanged in a production system.

The standardized map of production systems consist of standard classes of elements. Each element can be classified based on the behavior of it. The attributes of the classes are also standardized, so as the key performance indicators (KPI) of them. Comparing the performance of two or more different production systems is possible by applying the standardized KPI-s.

3.3.4. Parameterization of the generated simulation elements

A production database is used for providing parameters of the elements and input data for the simulation model (see Figure 1). Directly and indirectly usable data are gathered from MES database, and transformed as well as processed to the format that simulation software is able to apply (Kádár et al. 2010, Pfeiffer et al. 2009., Pfeiffer 2007.). The proposed simulation element parameterization method is demonstrated by an example in the next sections.

As test model elements were generated and connected based on the information stored in the low level control devices of a conveyor system it is obvious to calculate the transportation time which is the most relevant information of a conveyor element. Production database of the test system stores data of the number of pallets located on each conveyor sections in the function of time (see Fig. 4.)

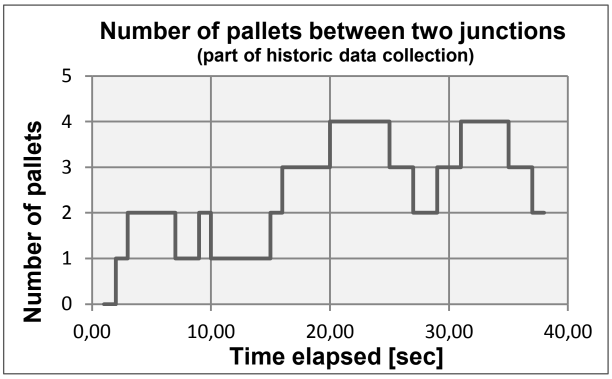

Fig. 4: Number of pallets located on a section of the conveyor system in the function of time

On Fig. 4. every rising edge represents the event when a pallet enters the observed section and falling edge is dedicated to the exiting event of a pallet. As the sequence of the pallets does not change between two junctions on the conveyor system the entering and exiting time stamp of every pallet can be calculated by a built in method of the simulation software. It also calculates the distribution of the transportation time of every section of the test area and sets it as a parameter of the appropriate simulation element.

4. Preliminary simulation results

Preliminary test runs were taken on a simulation model that was built automatically based on the data on the PLC programs of a test area of the above mentioned conveyor system. The generated simulation elements were parameterized based on the historical data of this section by the suggested parameterizing method. Input data was also generated based on historical data and time dependent capacity limitations of the output of the system were implemented in the simulation model as well.

The aim of the test run was to validate the simulation model by comparing historical data and simulated data. The compared parameter was the number of pallets located in test area. The values are plotted in the function of time (see Figure 5.). Ram up phase of the simulation run can be recognized at the start phase of observed time interval. In this phase there are significant differences between the real and simulated values because at the initial moment of the simulation run the model does not contain any pallet. After the ramp up phase the trend of the curves are similar that shows the simulated system behavior is adequate to the behavior of the real system. As uncertainty raises as the observed time interval of the forecast, this phenomenon can be observed on this graph as well. On the right side of the diagram the curves are deviating, because the uncertainty rises as the function of time.

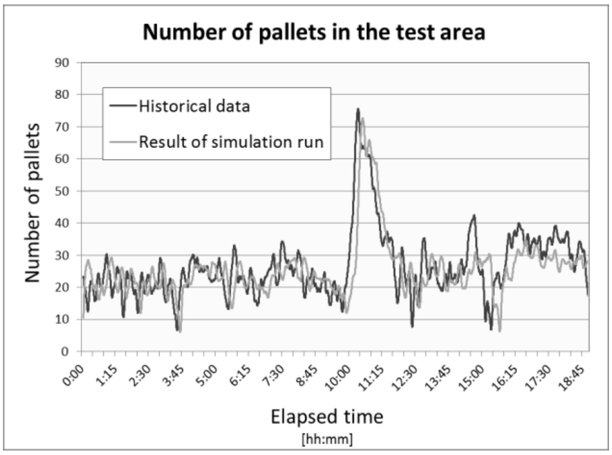

Fig. 5: Number pallets located in the observed test area

As the test run revealed it is possible to represent the behavior of a manufacturing system by a simulation model generated and parameterized automatically based on the information stored in low level control devices of the system.

Consequently, these results of simulation experiments can be easily adapted in real factory, thus enabling fast, flexible and smooth changes in factory, e.g., analyzing possible effects of changes in some part of PLC program of control system.

5. Conclusions

This paper revealed a discrete-event simulation approach applied for decision support of control related production applications. Design phase is a significant part of a simulation project; hence reducing time of it heavily affects the effectiveness of the whole project. Automated data gathering supporting the buildup of simulation models is a possible solution to achieve this goal and is also a solution to create reusable models. Several approaches were studied in the topic and revealed that PLC programs store information needed to build up a simulation model. A new process for extracting topology and control logic data of system from PLC programs has been introduced. Data stored in production database were used to parameterize objects of the model and generating input for simulation experiments. Preliminary test runs revealed that the behavior of the automatically generated simulation model is similar to the behavior of the real system, hence is suited to perform simulation experiments and forecasts.

Acknowledgements

Research has been partially supported by National Office for Research and Technology (NKTH) grant "Digital, real-time enterprises and networks", OMFB-01638/2009.

The research reported in this paper has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No: NMP2 2010-228595, Virtual Factory Framework (VFF).

Research has been partially supported by National Office for Research and Technology (NKTH) grant "Digital, real-time enterprises and networks", OMFB-01638/2009, as well as NKTH grant OMFB-01638/2009, T_AMOP-4.2.1/B-09/1/KMR-2010-0002 grant.

References

[1] Bagchi, S., Chen-Ritzo, C., Shikalgar, S.T., Toner, M. (2008) A full-factory simulator as a daily decision-support tool for 300mm wafer fabrication productivity, in Proceedings of the 2008 Winter Simulation Conference, S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds., pp. 2021-2029

[2] Banks, J. (1998) Handbook of Simulation, Principles, Methodology, Advances, Application and Practice. John Wiley & Sons Inc.

[3] Juyoung Wya, Sangwon Jeong, Byung-In Kim, Junhyuk Park, Jaejoon Shin, Hyunjoong Yoon, Sujeong Lee (2011), A data-driven generic simulation model for logistics-embedded assembly manufacturing lines. Computers & Industrial Engineering. Vol 60 Issue: 1. pp. 138-147.

[4] Kádár, B., Lengyel, A., Monostori, L., Suginishi, Y., Pfeiffer, A., Nonaka, Y. (2010) Enhanced control of complex production structures by tight coupling of the digital and the physical worlds, CIRP Annals - Manufacturing Technology, Vol 59, pp. 437–440. (http://dx.doi.org/10.1016/j.cirp.2010.03.123)

[5] Kwan Hee Han, Seock Kyu Yoo, Bohyun Kim, Geon Lee (2010) Rapid Virtual Prototyping of PLC-Based Control System. ICAI'10 Proceedings of the 11th WSEAS international conference on Automation & information pp. 21-26.

[6] Law, A., Kelton, D. (2000) Simulation modeling and analysis, McGraw-Hill, New York.

[7] O'Reilly, J.J., Lilegdon, W.R. (1999) Introduction to FACTOR/AIM. In: Proc. of the 1999 Winter Simulation Conference, pp. 201-207.

[8] Park, Hyeong-Tae, Kwak, Jong-Geun,Wang, Gi-Nam, Park, Sang C. (2010), Plant model generation for PLC simulation. International Journal of Production Research Vol. 48, Issue 5, pp. 1517-1529.

[9] Pfeiffer, A. (2007) Novel Methods for Decision Support in Production Planning and Control, Thesis (PhD). Budapest University of Technology and Economics.

[10] Pfeiffer, A., Kádár, B., Szathmári, M., Popovics, G., Vén, Z., Monostori, L. (2009) Self-building simulation tool for daily decision support in production control, In proc. of the 7th International Workshop on Modelling & Applied Simulation, MAS 2009, 23-25 September, 2009, Tenerife, Spain, pp.: 246-254.

[11] Son, Y.J., Wysk, R.A. (2001) Automatic simulation model generation for simulation-based, real-time shop floor control, Computers in Industry 45. pp 291-308.

[12] Ryan, J., Heavey, C. (2006) Process modeling for simulation, Computers in Industry 57. pp. 437–450