Automated simulation model building for a complex furniture manufacturing process

Tadej Kanduč[1], **Blaž Rodič**[1],

[1]*Faculty of information studies, Ulica talcev 3, Novo mesto, Slovenia*
E-mail: tadej.kanduc@fis.unm.si

Abstract

In this paper we present the first part of a project of flexible manufacturing process optimisation in a Slovenian furniture company. In this part we analysed the current state of the manufacturing system by constructing a discrete event simulation model that reflects the manufacturing processes. As the manufacturing process involves about 30,000 subprocesses it is not feasible to manually construct the model to include every subprocess, hence a method for automated model construction was developed. The method analyses open orders and constructs a corresponding model which includes only the machines and subprocesses required for the manufacture of ordered products. The obtained simulation model lets us inspect current manufacturing processes and allows us to easily apply modifications of the model parameters in manufacturing process optimisation.

1 Introduction

Optimising manufacturing processes is one of the key goals in every manufacturing company. If the processes are simple enough, they can be efficiently represented by exact mathematical model which can be effectively optimised with analytical mathematical algorithms. In practice, the processes are usually too complex to tackle them with this approach and a common alternative is to construct a discrete event simulation model (DES) in order to understand and analyse every stage process in a floor.

Our primary focus in the project is to investigate how the layout of machines on the factory floor affects the efficiency of manufacturing processes and to optimise the layout. We restricted our criteria of efficiency to measuring total time it takes to complete an order, machine utilisation, and total distance the manufactured products need to make on the floor. In this paper we describe the method of automated model building used in the first part of the project.

1.1 Description and complexity of manufacturing processes

The furniture company uses approximately 140 machines and produces over 30,000 different products. To manufacture a final product, each product has its own *technical procedure*, i.e. a sequence of prescribed tasks. For each product and task there is a list of suitable machines and expected completion times. More complicated products are manufactured by joining smaller semi-finished products according to the prescribed *bill of materials* (BOM). After a task at a machine is finished, the entire series of products is moved by carts to the next prescribed location (machine). List of all final products that need to be manufactured is determined by the clients' orders.

Due to the large set of different products and dynamic nature of the manufacturing processes, static model development is not feasible in this case. Instead, the model is built automatically, based on available machines and for current open orders. Technical procedures and BOMs are read dynamically from input data during the simulation.

1.2 Previous research (review of literature)

Simulation is commonly used for the evaluation of scenarios [1],**Napaka! Vira sklicevanja ni bilo mogoče najti.**,**Napaka! Vira sklicevanja ni bilo mogoče najti.**. However, the models developed with the visual interactive modeling method (VIM) are usually manually constructed through careful analysis of the real-life system and communication with process owners. Automated model development is more common with methods that allow easier and more standardized formal description of models, e.g. Petri nets [4],[5]. Automation of model construction and adaptation can importantly facilitate the development of models of complex systems **Napaka! Vira sklicevanja ni bilo mogoče najti.**,[7] and generation of simulation scenarios.

Several papers deal with factory layout optimisation, with the paper **Napaka! Vira sklicevanja ni bilo mogoče najti.** stating that multiproduct enterprises requires a new generation of factory layouts that are flexible, modular, and easy to reconfigure. Evolutionary optimisation methods are often proposed due to optimisation problem complexity **Napaka! Vira sklicevanja ni bilo mogoče najti.**.

Factory layout design optimisation is further discussed in **Napaka! Vira sklicevanja ni bilo mogoče najti.**,**Napaka! Vira sklicevanja ni bilo mogoče najti.**,**Napaka! Vira sklicevanja ni bilo mogoče najti.**. Authors **Napaka! Vira sklicevanja ni bilo mogoče**

najti. propose a new facility layout design model to optimise material handling costs. **Napaka! Vira sklicevanja ni bilo mogoče najti.** and **Napaka! Vira sklicevanja ni bilo mogoče najti.** propose genetic algorithm based solutions to respond to the changes in product design, mix and volume in a continuously evolving work environment.

1.3 Placement of project within end-users activity

Primary goal of the company is to reduce overall costs in manufacturing processes. This can be achieved by removing bottlenecks (overloaded machines), reducing transport distances (distances the carts need to travel between the machines), reducing overall time to finish a work order or by increasing overall machine utilisations. It is also possible to add machines if considerable improvements can be achieved. The goal of the project is try to tackle these problems by finding a more effective layout of machines.

1.4 Goals of the project and used methods and tools

Our first step was to develop a manufacturing process simulation model that would reflect the current situation in the company. The verification of the model was made by running our model with test and historic input data, prepared by the company planners.

The next step of the project, which is currently going on, is to propose a new machine placement on the floor. The improvements of the layout can be achieved by applying heuristic optimisation methods and with conjunction with expertise from company planners and workers. At the moment, we are focusing on optimising total length the products need to travel on the factory floor.

The discrete event simulation model was built in Anylogic software (http://www.anylogic.com). The model is contained in XML file. The main result of this phase of the project is an application in Java which constructs the model by modifying an XML file with the template model.

2 Methodology

In this section we will describe machines, procedures and processes in the company in more detail. Then we will describe our simulation model and the algorithm for automatic model building and all important methodologies we have used.

The company floor layout consists of approximately 140 machines (working places). Each machine works/is operated independently and takes certain space on the factory floor. In the layout around the machines a network of possible paths is defined. We presume that all carts move on these paths with constant speed and by the shortest route possible.

Each machine can perform specific *operations*. Before starting an operation, the machine needs to be *set up* to be able to perform the prescribed tasks. Both set up and operation take specific amount of time.

A machine has the *input* and the *output pallet* (a landfill for the products). Once the operations for the series of products is finished, the series of products is moved to the next chosen machine by a cart. Every machine is described by the following pseudo-algorithm:

(1) If input pallet is not empty, pick a product among the list; else wait.
(2) Set up the machine.
(3) Apply the operations for the whole series of the product (products then wait on the output pallets).
(4) Pick a suitable machine for the next operation that is at that moment the least loaded.
(5) Series of product is moved to the next machine.
(6) Go to (1).

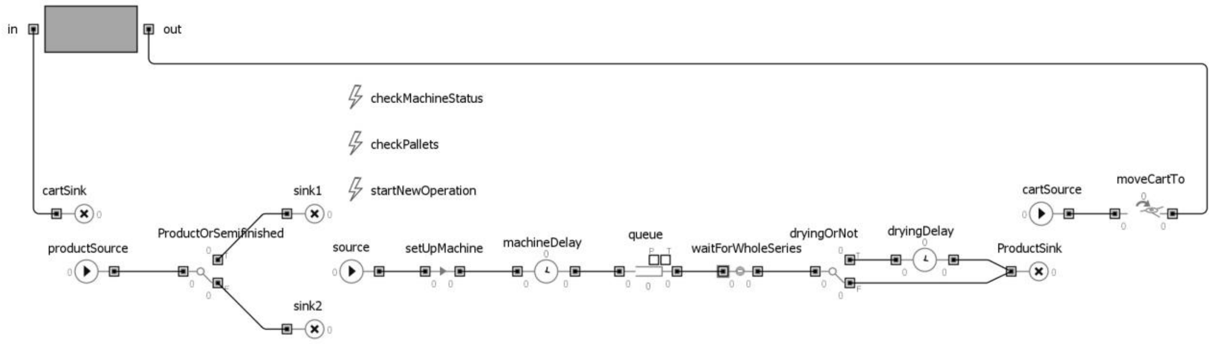

Figure 1: machine block logic

Implementation of the machine block in Anylogic is shown in Figure 1: machine block logic. On the input of the block, carts filled with products enter the system. Products are sorted according to their type at *productOrSemifinished*. The corresponding sinks, *sink1* and *sink2*, monitor products on input pallets. Once the product is chosen for operation, it is injected at *source*. Blocks *setUpMachine* and *machineDelay* are standard service blocks. Block *waitForWholeSeries* plays a role of output pallet. Some products need to wait at *dryingDelay* according to the technical procedure (paint drying, etc.). Filled carts are injected at *cartSource* and moved to the next location at *moveCartTo*. Output of the main machine block sends a cart to the input of the next machine.

An *order* is a list of products that needs to be manufactured. Usually there are several open (active) orders in the production at the same time. An order consists of considerably smaller number of different products than 30,000. To simulate manufacturing processes for a specific time interval, only products that are specified in currently open orders need to be considered.

Manual modifications of the simulation model can be time consuming, especially if a large set of variations of the model needs to be built. In Anylogic, simulation model is typically constructed by adding different blocks and connections on the canvas by "click and drag" technique. On the other hand, the model can be constructed automatically by writing an external script file. The Anylogic model is an XML file which can be easily modified.

Our simulation model is hence constructed automatically from the available set of machines that can be determined as an input data. Technical procedure data to construct the products are read dynamically "on the fly" during simulation. All necessary data to construct the model are obtained from the relational database used by the Preactor manufacturing scheduling application and transferred to an intermediary Excel spreadsheet that allows easier editing. Since Preactor can only be used for scheduling the orders and following the realisation of schedule (using app. 100 PDA equipped control points on the floor), Anylogic was used for floor layout optimisation. Model of the current layout in Anylogic was produced for verification of the optimisation method and validation of new layouts.

2.1 Overview of the system

Our modelling and simulation system (Figure 2) is composed of four main elements:

- Manufacturing process simulation model that runs in the Anylogic environment,
- Java application that constructs XML file describing the Anylogic simulation model,
- MS Excel as an intermediate data storage and analysis tool,
- MS SQL server database with data from the Preactor manufacturing scheduling software, which serves as the data source for open and technical procedures.

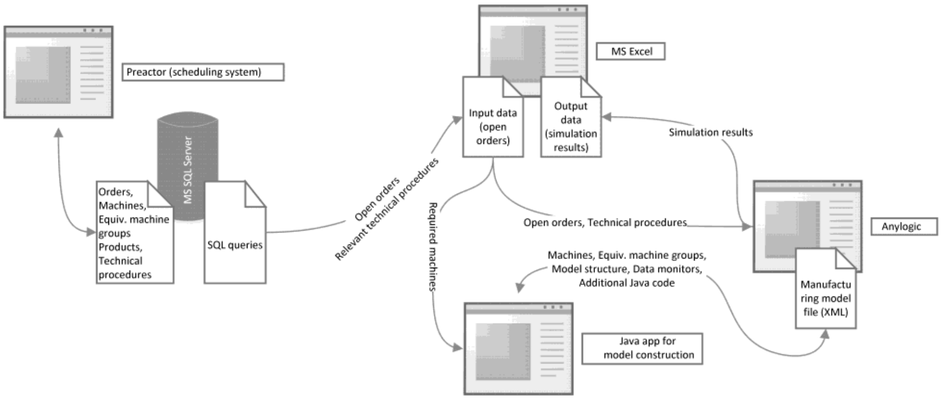

Figure 2: System schematics

2.2 Description of algorithm and model data structure

Although it would be possible to write an algorithm that builds the entire simulation model from scratch, it is more convenient to construct a template model in Anylogic first. The template includes basic blocks and connections such as a block representing a machine, floor layout, input reader, output writer etc. The algorithm then reads the template blocks in the template file and copies them according to input data.

The newly obtained XML structure is saved to a new file. The simulation model is constructed automatically according to the list of machines required by the orders. Products and carts play a role of transactions in DES and are therefore constructed dynamically during simulation. All the technical procedures for manufacturing processes are also read dynamically.

Anylogic XML simulation model file stores information on blocks/agents (machines, pallets, data sources and sinks), connectors between blocks, network (paths between the machines), statistical monitors, input reader and output writer. The algorithm manipulates XML code to change data on machines and all other relevant abstract objects such as connectors, sources and sinks that are connected to the blocks of machines.

2.3 Description of output data

During simulation various statistical data are measured. Once the simulation is finished, all the data are stored in output Excel file. Between every pair of machines we measure different types of flows (number of products, number of used carts, overall volume of products and total distance of carts). For every machine we monitor utilisation, overall setup time, flow of products and volume, and queue. For each series of products we measure starting and ending time, and store the sequences of machines that were chosen during simulation. Furthermore, we measure flow of carts and record routes of the carts.

3 Results and discussion

The developed simulation model and accompanying model construction tool lets us change the floor layout according to end-user input or optimisation method outputs, inspect the resulting manufacturing processes, execute simulation for various sets of open orders with different products without manual modifications of the model and verify the model operation and results with the end users. The automated model building tool considerably speeds up the iterative process of modifying the simulation model and is essential sub-method to apply optimisation algoritms.

The next step in the project is to the develop optimisation methods for machine layout. We are considering several methods and are currently developing a physical model based optimisation method. First results of this method show that by rearranging the machines we can reduce the overall traveling distance of the manufactured products for more than 30%. Improvements in overall time to finish an order by rearranging the machines are negligible. This is due to the fact that machine operations take much more time than time moving the carts. The same goes for the machines utilisations. Improving these criteria could only be achieved by adding new machines to remove bottlenecks. This option will be further explored in the project.

Acknowledgements

Work supported by Creative Core FISNM-3330-13-500033 'Simulations' project funded by the European Union, The European Regional Development Fund. The operation is carried out within the framework of the Operational Programme for Strengthening Regional Development Potentials for the period 2007-2013, Development Priority 1: Competitiveness and research excellence, Priority Guideline 1.1: Improving the competitive skills and research excellence.

References

[1] M. Kljajić, I. Bernik, A. Škraba: Simulation approach to decision assessment in enterprises. Simulation, 2000, 75(4), pp. 199-210.

[2] R. S. Edis, B. Kahraman, O. U. Araz and M. K. Özfirat: A facility layout problem in a marble factory via simulation. Mathematical and Computational Applications, 2011, 16(1), pp. 97-104.

[3] P. Tearwattanarattikal, S. Namphacharoen and C. Chamrasporn: Using ProModel as a simulation tools to assist plant layout design and planning: Case study plastic packaging factory. Songklanakarin Journal of Science and Technology, 2008, 30(1), pp. 117-123.

[4] R. Conner: Automated Petri net modeling of military operations, in IEEE Proceedings of the IEEE 1990 National Aerospace and Electronics Conference - NAECON 1990, Volume 2, 1990, Dayton, Ohio, USA, pp. 624-627.

[5] D. Gradišar and G. Mušič: Automated Petri-Net Modelling for Batch Production Scheduling, in (ed: Pawel Pawlewski) Petri Nets - Manufacturing and Computer Science, InTech, 2012, pp. 3-26. http://dx.doi.org/10.5772/48467

[6] A. D. Lattner, T. Bogon, Y. Lorion and I. J. Timm: A knowledge-based approach to automated simulation model adaptation. In Proceedings of the 2010 Spring Simulation Multiconference (SpringSim '10). Society for Computer Simulation International, San Diego, CA, USA, 2010, Article 153.

[7] R. Kannan and H. Santhi: Automated construction layout and simulation of concrete formwork systems using building information modeling, In (eds: Djwantoro Hardjito & Antoni) Proceedings of The 4th International Conference of Euro Asia Civil Engineering Forum 2013 (EACEF 2013), National University of Singapore, 26-27 June 2013, pp C7-C12.

[8] S. Benjaafar, S. S. Heragu and S. A. Irani: Next generation factory layouts: Research challenges and recent progress. Interfaces, 2002, 32(6), pp. 58-76.

[9] A. Sadrzadeh: A genetic algorithm with the heuristic procedure to solve the multi-line layout problem. Computers and Industrial Engineering, 2012, 62(4), pp. 1055-1064.

[10] K. K. Krishnan, S. H. Cheraghi and C. N. Nayak: Facility layout design for multiple production scenarios in a dynamic environment. International Journal of Industrial and Systems Engineering, 2008, 3(2), pp. 105-133.

[11] J. S. Kochhar and S. S. Heragu: Facility layout design in a changing environment. International Journal of Production Research, 1999, 37(11), pp. 2429-2446.

[12] M. Enea, G, Galante and E. Panascia: The facility layout problem approached using a fuzzy model and a genetic search. Journal of Intelligent Manufacturing, 2005, 16(3), pp. 303-316.