A MODULAR MODELING APPROACH TO STOCHASTIC SIMULATION OF PRODUCTION - LOGISTIC SYSTEMS

Giovanni DAVOLI, Sergio A. GALLO,

Andrea GOVONI, Riccardo MELLONI

Department of Mechanical and Civil Engineering (DIMeC),

University of Modena and Reggio Emilia (Italy),

via Vignolese 905, 41100, Modena

Italy

E-mail: giovanni.davoli@unimore.it

Abstract:

The economic scenario today is highly competitive in terms of costs and number of competitors, so it is necessary to adopt strategies that allow the constant improvement of manufacturing processes within the spending constrains. Simulation models are useful to support and drive company management in improving the performances of production and logistic systems. The costs of simulation model development could be reduced by the reuse of some of its parts. This work presents a case study concerning stochastic modeling of a small manufacture operating into the wood products field. A modular simulation model composed of reusable sub-models has been developed using AutoMod™ software package. The aim of the modular architecture is to allow the use of sub-models in different production systems with little changes, decreasing the costs of development in order to became more affordable in a SME (small medium enterprise) contest.

Keywords:

Simulation, processes, modular model, SME, AutoMod™.

1. INTRODUCTION

Today the economic scenario is highly competitive in terms of costs and number of competitors. All companies need to adopt strategies that allow the constant improvement of manufacturing processes within the spending constrains. The purpose of making efficient and flexible manufacturing systems is often related to the possibility to analyze the system considering at the same time a wide number of parameters and their interactions. Simulation models are proved to be useful to support and drive company management in improving the performances of production and logistic systems. In particular, production improvement in terms of production capacity, scheduling algorithms, layout and processes optimization are key areas in which simulation can support operational decision-making. However, to achieve the expected results, a detailed model of the production and logistic system is needed. Carry on the development of such a simulation model is an expensive activity, especially in term of resources and time used. Moreover the model developed for a specific enterprise, in most cases, is not reusable for another one. This aspect contributes to elevate the costs of simulation model development. The construction of the simulation model is similar to any other software development procedure going through the requirements specification, analysis, design, implementation, test and maintenance steps [3]. The authors of this paper and many other researchers [7] argue that simulation is one of the major tools to assist in the Business Process Re-engineering (BPR) and improve performance. Even if there are many arguments to support this thesis, simulation has not been widely applied to SMEs [7]. Many simulation studies are focused on large enterprises because of the prohibitive cost of simulation tools and the expertise required to develop credible and validated models. SMEs can enhance their production and logistic by the use of apply simulation tools.

Simulation can be developed following two different strategies. One is using commercial specialized simulators, and the other is using general-purpose computer languages. Manufacturing-oriented simulation language such as AutoMod™, are simulation software mainly focused on user friendly features and simulation tool for typical manufacturing systems. These software also have the advantage of faster modeling compared to general purpose languages. On the other hand, general computer languages have the advantage of flexibility. In fact a simulation model developed using languages such as C++ can be easily interfaced with other software or modules. The major disadvantage of this kind of programming is that expertise is required [2].

Languages	**Examples of software**	**Characteristics**
Simulation language	Arena ™, MODSIM III ™, SIMPLE++ ™	Flexible
Manufacturing-oriented language	AutoMod ™, AutoSched ™, Quest ™	Powerful, rapid modeling
Manufacturing-oriented simulator	FACTOR ™, FactoryFLOW ™, WIT-NESS ™	Easy to use, limited adaptation area
General computer language	C++, Java	Need for expertise, high flexibility

Table 1.1: Examples of simulation software and their characteristics [2]

Simulation is not an optimizing tool, it can be used to find the best solution between the considered scenarios. Simulation experiments often require the examination of a potentially large number of scenarios dealing with many solution strategies. This is often an iterative and time-consuming task. Because the development time to build new models or make changes to existing models can be quite substantial, there exists the need for research into new simulation techniques to reduce the effect of these problems. One such technique is to develop generic, modular simulation solution systems. The possibility of developing simulation models with a modular approach can lead to some benefits: the division of the model in parts as to be checked in an easier way, the possibility of reusing parts of the model in another one, the separation between logic and physical parts. In the area of system modeling, modularity is used to improve system understanding, to reduce the model complexity, and to facilitate the reuse of standardized components [1].

General computer languages, such as C++, allow a very effective modularity and flexibility, they can be used for modeling simple production systems like “work station 1” – buffer – “work station 2”, but the use of this type of language is not easily applicable to complex systems. The simulation of a complete production line by the use of this type of programming language requires expertise and much more time than using a specific manufacturing-oriented language. For these reasons, even if modular modeling by the use of manufacturing-oriented language is not versatile as could be general computer languages, in the majority of applications should be preferred because of his simplicity and rapidity. Especially in the SMEs, there are no C++ programmers for developing the models, for that reason manufacturing-oriented language should be preferred instead of general computer languages. We intend make use of modular modeling developed within the

Automod™ software and we want to demonstrate the efficacy of the approach using a case study from an Italian SME. The possibility to insert C++ code inside the Automod™ language is useful in particular situations but, in the present work, it is not considered an effective solution for increase the modularity of the code, mainly because of the lack of C++ expertise inside the SME staffs.

The authors have long advocated the use of modular simulation code to enable rapid evaluation of multiple scenarios. Other authors have addressed modularity in simulation using a variety of languages. Standridge [9] uses ModNet, a language designed to support modular network simulation. Valente de Olivera et al. [6] and Luna [6] have made use of the Smalltalk language. Valente de Olivera et al. describes an object-oriented mobile robot simulator, while Luna makes use of object oriented code construction to concurrently apply both hierarchical and modular features. Robinson and Kisner [8], for example, use the well-established LISP language to simulate a nuclear power plant environment. In their work, objects communicate via message passing and the compile link-run cycle is eliminated. Sang et al. [6] discuss a collection of library functions with pre-defined data structures, called Si, to enhance the C programming language. Guasch [6] argues that continuous system simulation environments offer a higher level of modularity than their discrete counterparts because they facilitate the use of automatic model building through the use of hierarchically connected sub-models. The integration of these sub-models is simpler than in discrete models. Other authors [5] discusses an interesting use of object oriented programming with animation to study just-in-time (JIT) systems.

2. CASE STUDY

This work presents a case study concerning stochastic modeling of a small manufacture operating into the wood products field. The enterprise is a SME of approximately 10 employees, it needs the increase productivity in order to face off the peaks in the products demand. In figure 2.1 is represented the plant lay-out, the different working zones are shown and named: ZONA 0, ZONA 1, ZONA 2, ZONA 3, ZONA 4, ZONA 5.

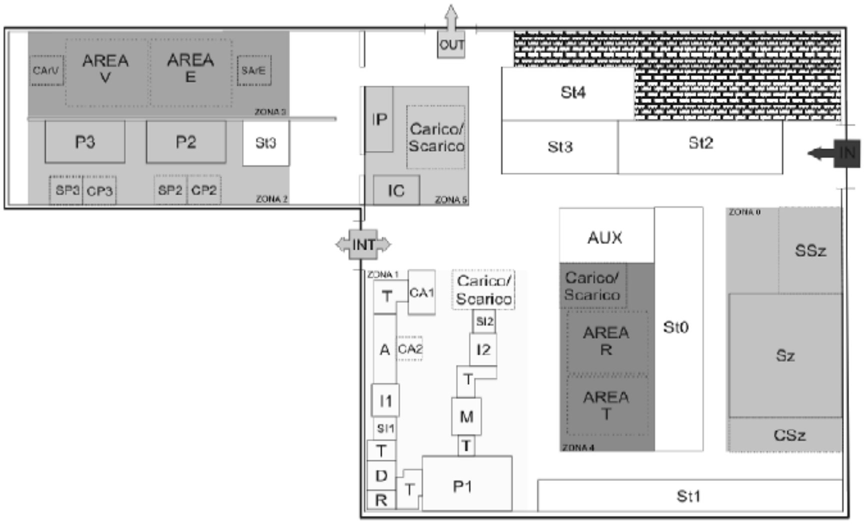


Figure 2.1 - Plant lay-out

The increase of the production lines efficiency should be done respecting the spending limits imposed by management. The modeling process takes start from the analysis of business processes that was carried out in collaboration with the enterprise staff.

2.1. Modular model development

The purpose of the modeling activities is to provide a virtual representation of the productive and logistic system. The model is useful for the development of alternative solutions in order to improve the efficiency of the productive and logistic system.

A modular simulation model of the production and logistic system is developed using AutoMod™ software package. AutoMod™ suite includes templates to accurately model material handling systems, including conveyors, lift trucks, operators, automated vehicles, overhead cranes, power and free, ASRS (Automated Storage/Retrieval System), and any cinematic device. AutoMod™ uses CAD-like features to define physical manufacturing layout, material handling, processing, and distributed systems. AutoMod™ powerful graphical interface accurately captures physical constraints of distance, size and space in three dimensions. In the AutoMod™ simulation environment a model can be developed hierarchically using "sub-system" objects. In this case the production and logistic system is split in two main sub-system: the production sub-system and the logistic sub-system. The production sub-system is split in two sub-system that coincide with different working areas: the cutting area and the varnishing and finishing area. Even the logistic system in split in two sub-system: the in-coming materials area and the out-coming materials area. So a three level modular model is developed.

The developed model was able to obtain significant results in terms of system analysis and production capacity increase within the economic constrains. Moreover, developed sub-models are independent objects that can be reused to model the production and logistic system of a different manufacture. All the sub-systems provide a data base of ready to use objects that enable the modeling team to save time and resources in the subsequent cases.

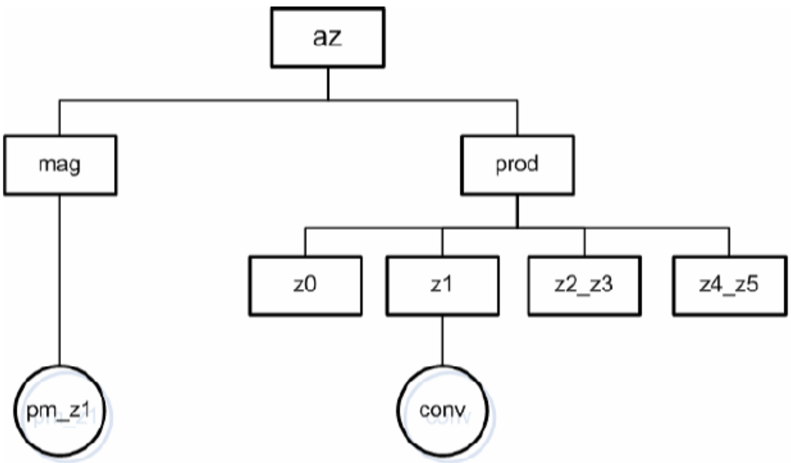


Figure 2.1.1 – Model structure

Figure 2.1.1 shows the model structure: az = main module, prod = production model, mag = warehouse module, z0,z1,z2_z3,z4_z5 = zone modules, pm_z1 = path movers, conv = conveyors.

The logical system can be identified by two type of logic blocs:

1. the working zone (ZL)
2. the control unit (UC)

The ZLs in Automod™ consist of both graphical and programming parts. The UC is composed by one or more sentinels for controlling the information data stream. In this case study three UC e six ZL have been identified: a UC for production, a UC for the warehouse, a UC for the whole firm, and six ZL related to the production zones.

2.2. Test and validation

The model shows the physical and technological constraints and it also encodes the organizational strategies used. A comparison between the real data and the results from simulations allow to validate the model. The validation process consists of comparing the real data with the simulated ones. To validate the model the total production time is considered with an appropriate lot size. The historical average data suggest to use a lot size of 3500 for the production of product PR1-1 and *a* lot size of 150 for the production of product PR2. Table 2.2.1 presents the average values of times needed for the production of a single lot.

Production	Lot size	Real total time (s)	Simulated total time (s)
PR1-1 - Z1A	3500	28800	25200
PR1-1 - Z1B	3500	28800	30000
PR2 - Z2A	150	28800	25200

Table 2.2.1 – Comparison of times for the validation process

By the analysis of these values, we have checked that the model is validated, in fact the error between real and simulated values is acceptable.

2.3. Results

The simulation results suggest that machine M is a bottleneck for the production in ZONE 1B, mainly because it processes two pieces contemporary but them both have to be loaded at the same time. Different scenarios simulation proved that machine M is the limiting factor in its production zone.

Once identified the bottleneck, alternative solutions are developed and simulated. The scope of the alternative solutions are to increase the use rate of the working zones. Many possible alternatives have been checked and the one that presents more advantages without exceeding the spending limit is focused to replace machine M with two parallel machines. We consider two alternatives, a quite expensive solution with two M1 machines, and a budget solution with two M2 machines. The M1 and M2 machine processes a single piece at a time, despite of M machine that works two pieces contemporary. The difference between M1 and M2 is the process time.

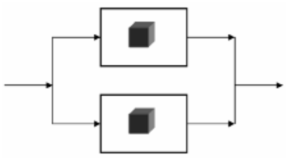
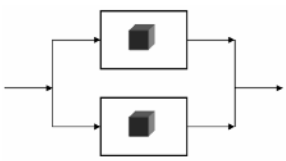
Machine	Scheme	Capacity	Process time for one piece (s)	Process time for two pieces (s)
1 x M		2 pieces	15	15
1 x M1		1 piece	5	10
1 x M2		1 piece	10	20
2 x M1 parallel		2 pieces	5	5
2 x M2 parallel		2 pieces	10	10

Table 2.3.1 – Characteristics of M, M1 and M2 machines.

As shown in table 2.3.1 a single M machine can processes two pieces in the same time as two M2 machine in parallel, the advantage of the solution of two machines is that the two pieces have not to be loaded contemporary. To simulate the appropriate solution, also the conveyor has been modified to feed both parallel machines. At this point, the main advantage of modularity is shown, in fact the programmed code have been modified only in a couple of lines for developing the new model. If the code was not programmed in a modular way, it would be necessary to re-wrote many logic functions in order to face the new modified scenario. Table 2.3.2 shows the results in terms of production capacity.

Scenario:	Time (h)	Number of products (pieces)	Standard Deviation (pieces)
1 x M – current situation	8	2764	28
2 x M1 – quite expensive solution	8	- -	- -
2 x M2 – budget solution	8	3154	31

Table 2.3.2 – Results.

It is possible to notice that in a working day of 8 hours the pieces processed passes from a value of about 2764 to about 3154, this show an increment of 14% in the production. The scenario with 2xM1 parallel machines has not been simulated because of it exceeds the spending constrains.

3. Conclusions

To develop a model of the production system, Automod™ software suite is chosen because of: ease of use, rapid modeling, possibility of modular modeling. The approach carried on allows to develop a modular simulation model of the production and logistic system. The model is able to provide reliable results and allows to identify bottlenecks in the production line. Once a bottleneck has been discovered it is possible to provide different solutions in order to increase the productivity. The models of alternative solutions are easily developed thanks to the modular structure of the Automod™ model. Modeling and simulation process has allowed to provide the expected increase in production capacity within the spending constrains.

Modularity gives the opportunity to modify a model without the need of rewriting large portions of code. This feature is an advantage during model implementation and the model becomes a valid tool ready to be used in future for analyzing new problems in the same production and logistic system. Thanks to the modular approach, the code could be easily modified in order to follow the changes in the production system or any of its subsystems.

The present experience points out some limitations of the software tool for modular modeling. During model development it is impossible to open more than a module at a time, so every module must be closed in order to open a new one. Moreover, avoiding the use of low-level languages such as C++, a new data structure definition is prevented.

4. References:

[1] El Haouzi H., Thomas A., Petin J.F., *Contribution to reusability and modularity of manufacturing systems simulation models: Application to distributed control simulation within DFT context*, International Journal of Production Economics, Volume 112, Issue 1, Special Section on Recent Developments in the Design, Control, Planning and Scheduling of Productive Systems, March 2008, Pages 48-61, ISSN 0925-5273, DOI: 10.1016/j.ijpe.2006.12.067;

[2] Woo Seok Kim, Young Ha Jeon, Sung Jo Kwak, Doo Yong Lee, Hyun Joong Yoon, *High-fidelity simulation of integrated single-wafer processing tools for evaluation of scheduling algorithms*, Robotics and Computer-Integrated Manufacturing, Volume 25, Issue 1, February 2009, Pages 107-121, ISSN 0736-5845, DOI: 10.1016/j.rcim.2007.10.003;

[3] Kovacs G. L., Kopacsi S., Nacsa J., Haidegger G., Groumpos P., *Application of software reuse and object-oriented methodologies for the modelling and control of manufacturing systems*, Computers in Industry, Volume 39, Issue 3, July 1999, Pages 177-189, ISSN 0166-3615, DOI: 10.1016/S0166-3615(98)00134-1;

[4] Law Averill M., Kelton W. David, *Simulation modeling and analysis* - 4. ed. - Boston: McGraw Hill, 2007. ISBN 0073294411;

[5] Lyu J., *Using animated object oriented simulation to study a JIT environment, Proceedings of the First Joint Conference of International Simulation Societies*, Zurich, Switzerland, 1994, pp. 124-128;

[6] Meinert Timothy S., Taylor G. Don, English John R., *A modular simulation approach for automated material handling systems*, Simulation Practice and Theory, Volume 7, Issue

1, 15 March 1999, Pages 15-30, ISSN 0928-4869, DOI: 10.1016/S0928-4869(98)00016-0;

[7] O'Kane J., Papadoukakis A., *Simulation usage in SMEs,* Journal of Small Business and Enterprise Development, 2007, Vol. 14 No. 3 pp 514-527.

[8] Robinson J.T., Kisner R.A., *An intelligent dynamic simulation environment: An object-oriented approach*, Proceedings of the IEEE International Symposium on Intelligent Control 1988, Arlington, VA, USA, 1988, pp. 687-692;

[9] Standridge C.R., *Modular modeling for network simulation languages: Concepts and examples*, Proceedings of the 1995 Winter Simulation Conference, Arlington, VA, USA, 1995, pp. 736-743;