A Novel Logic and Approach to Speed up Simulation and Analysis of Production Systems

[1]Francesco Aggogeri, [2]Marco Mazzola, [3]Alberto Borboni, [4]Rodolfo Faglia
*[*1]DIMI, University of Brescia, via Branze 38, 25123 Brescia (Italy)*
francesco.aggogeri@ing.unibs.it
[2]DIMI, University of Brescia, via Branze 38, 25123 Brescia (Italy)
marco.mazzola@ing.unibs.it
[3]DIMI, University of Brescia, via Branze 38, 25123 Brescia (Italy)
alberto.borboni@ing.unibs.it
[3]DIMI, University of Brescia, via Branze 38, 25123 Brescia (Italy)
rodolfo.faglia@ing.unibs.it

Abstract

The paper describes a novel logic to model, simulate and analyze production systems using Discrete Event Simulation tools. An adaptable and flexible DES model has been developed to represent different manufacturing environments. The goal of the proposed DES-based parametric model is to offer to industrial analysts a compact, fast and easy way to configure and use decision support tools. The logic is based on elementary structures with a vertical multilayer architecture that permits entities to access and loop in multiple times. Since it has been assumed manufacturing systems are combination of elementary parts, the architecture of the model becomes quite simple since representing few structures only, and the simulation of heterogeneous systems is the result of as many iterations as needed by a user. This approach is adaptable both to SMEs and large companies, permitting to avoid the use of dedicated SW and allocated skilled person.

Keywords: *Discrete Event Simulation, Parameterisation, Configurability, Production Systems*

1. Introduction

With the increasing demand on productivity, quality and availability, factory simulation tools have become more strategic and important to gain competitive business advantages [1-3]. The decision-making processes involve a lot of information, driven by managerial strategies, technological implications and layout constraints [4-7]. In this way many factors that affect decisions and their combinations have to be carefully managed to determine best solutions and optimize performances [8]. Discrete Event Simulation (DES) can help the decisions making processes by providing effective models and playing what-if games with them, moving towards optimal scenarios [9,10]. The modeling of manufacturing processes, validation and optimization require time and cost consuming activities [11], especially for SMEs. The proposed study aims to create an adaptable and flexible model able to be automatically tailored on user’s specific needs without modeling expertise. The goal of the DES-based parametric model is to offer to industrial analysts a compact, fast, easy way to configure and use decision support tool. The main expected benefits are to use DES tools not holding specific knowledge of simulation tools, saving time and avoiding costs related to training, modeling efforts and analysis. In particular the proposed model offers the possibility to interpret results and key performance indicators (KPIs) at different detail level (from workstations to shop floor) by inputting parameters.

The model consists on an extremely flexible working environment, enabling to effectively represent and configure manufacturing cells and lines. In this way, any user could tailor the DES model on their processes not holding specific knowledge of dedicated simulation software. An innovative combination of an easy and fast parametric-driven configuration and the traditional optimization opportunities of DES can be obtained. In realizing a parametric DES-based model, the main objective is to enhance flexibility and users’ accessibility to analysis, simulation and results. The traditional approach consists on formulating the problem, building the simulation and running the model, and analyzing the output [12]. In this way the effort and the support of a DES expert or analyst is necessary and fundamental. The novel approach can lead to many advantages, even if penalizing some innate opportunities of DES tools such as visualization of flows or free-to-model architectures. The user can interact with the model

doi:10.4156/jcit.vol8.issue3.83

by configuring a pre-defined architecture, sending data automatically in the appropriate way and receiving simulated performance values. If a model can represent multiple manufacturing processes, extremely variable by typology, complexity and characteristics, there is the need to determine common key elements and structures [13]. Following this logic the application of DES tools is possible without specific knowledge or expertise of dedicated simulation software. A flexible and adaptable model is ready to be used, the user following a structured roadmap is guided through many options in order to configure its manufacturing processes with the less amount of required information.

The philosophy lying under the proposed model reflects that [14]:

- manufacturing streams are sequences of basic manufacturing processes to transform or move a product;
- manufacturing processes are basically iterations of elementary functional blocks. Variants of the blocks can be logically related to opportune set of parameters (technological, managerial and/or physical).

It is clear that the model cannot solve every case study, nevertheless it has been designed to cover as many layouts or, in general, manufacturing alternatives as possible. Lot of efforts have been spent to understand the heterogeneity of the manufacturing systems and the key elementary structures. Particular attention has been devoted in finding similarities (technological, managerial or either logical) between manufacturing systems, even if they could appear extremely different.

2. Parameterisation of production systems

The parameterization is the way chosen to reach the desired high level of configurability and adaptability of the DES-based model. In informatics, a parameter is the constrained value a function, a program or a system awaits to perform the task they have been thought to [15]. The model is intended to simulate multiple manufacturing systems, thus the design phase dealt with defining how to differentiate from a logical point of view physically different environments. The answer is a combination of options and parameters. The parameters are the way that guarantees to the user an immediate access to simulative results.

The interaction between potential users and the model is critical. Thus, the design of the model architecture cannot consider the simulation environment only, but the software development must comprehend how the user would potentially interact and how the data flows should be. The logical process under the software development is (Figure 1):

- define the whole architecture of the model, identifying the alternatives that the model should cover and the characteristics of every alternative;
- define the functional blocks that compose the model and what is the parameterisation need to obtain every alternative as one of the potentially infinite combinations of blocks;
- describe the potential configuration path of every alternative and design it like a web application;
- for every path, define the data needed from the user through parameter valuation and the automatic setup of values depending of the path chosen;
- organize data in the appropriate way accordingly with the DES language.

Ideally, an extremely adaptable and flexible model could be represented by an almost infinite sequence of redundant events; the model could be adapted to different industrial scenarios by configuring only part of this infinite repetitive structure. But, both the model and the computational sizes would be unacceptable. The logic under the parametric DES model uses elementary structures, achieving the goal of adaptability not through a redundant design, but thanks to an iterative reading of information. Since it has been assumed manufacturing systems are combination of elementary parts, the architecture of the model becomes quite simple since representing few structures only, and the simulation of heterogeneous systems is the result of as many iterations as needed by a user.

The key structure of the whole model is a manufacturing operation (or step). By the way, a step has to comprehend every type of operation that can incur in manufacturing processes. This structural element has been designed in a functional, smart way, ready to be used and re-used as many time as needed to model a manufacturing system. The model is DES-based. This means the model respects some typical assumptions and concepts of DES. The model has been built basing on a COTS tool: Arena simulation (by Rockwell) [16]. This commercial software is a powerful tool to describe in a flexible, dynamical, stochastic way processes from a logic point of view. The choice of Arena has been driven by the need to develop models accessible by as heterogeneous as possible users, not really interested in visualization of the processes but rather in logically representing them and valuating the key performance indicators governing them. The same logic can be applied to other COTS tools.

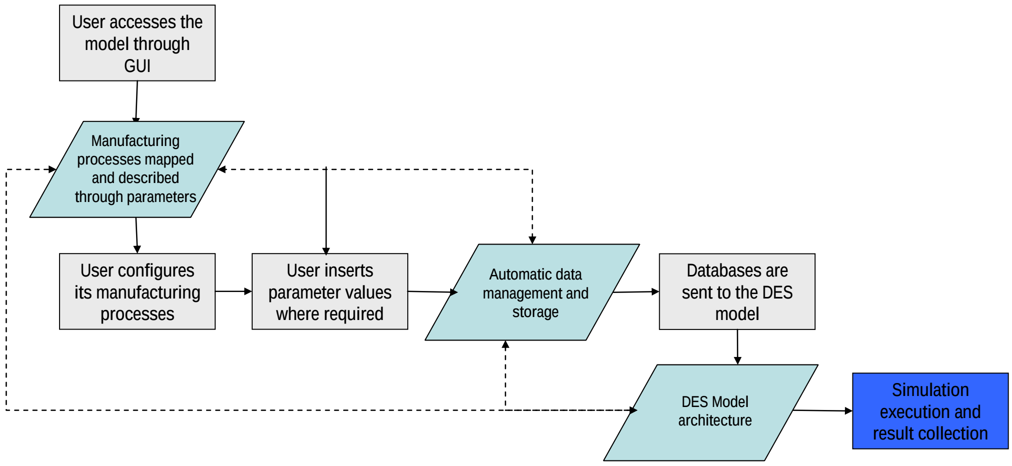


Figure 1. The objective of the model and its roadmap

In DES models, the events are generated by the entities. In representing a manufacturing process, the entities are basically the products flowing throughout the factory. The entities potentially interact with some resources, elements characterized by a finite capacity like machines, equipments, operators, transporters. Entities progressing in a DES model can be marked with attributes, a sort of labels. The values of the attributes determine the characteristics of every single entity. Coming back to the manufacturing systems, attributes can be used to describe the product type, the workstation, the product (it is visiting), the operators needed for a specific operation and so on. But, since the key element of the model is the step, the relative attribute (STP) controlling the entity progress within the model becomes fundamental. Every time an entity enters a step in the model, some attributes are updated to link the entity to the relative step. Once defined with attribute values, the step has to be appropriately characterized by valuating a set of parameters.

In practice, an entity enters the structure of the model and loops in it as many time as needed to evolve from the simulation of a single elementary operation to a complex manufacturing process. An entity progressing in the model is redirect to the same elementary structures instead of being sent to the next. When representing either a complex or simple sequence of operations, a DES model appears like a horizontally developed structure, where the model backbone reflects the sequences of dynamic events (Figure 2a). As the number of operations growths, the model becomes big and redundant.

The model architecture reflects that it has been designed to represent heterogeneous sequences of operations; thus, until a user does not configure the model, the dimension of the problem is unknown. Furthermore, the model would have to simulate either a single operation (a step), a sequence of steps (a workstation) or sequences of sequences of steps (a system composed of multiple parallel lines).

The impossibility to know an “a priori” dimension of the model, the extreme variability of cases potentially solvable by the same model lead to design the DES model basing on a vertical architecture (Figure 2b and 2c). In this configuration, every basic element (the step) is a layer and manufacturing processes becomes stack of layers. In a DES model, entities have to evolve throughout series of event. To do this, a horizontally developed model would need multiple blocks, whilst the “vertical” one has only a key piece of software, a vertical multilayer structure, the entities access and loop in multiple times. Since the verticality acts at logic level, the DES model is basically a generic unparameterized elementary structure with unknown, variable dimension. Figure 2 shows the evolution towards the vertical logic behind the model. The benefits of the model compression can be appreciated immediately.

2.1 Parameterisation of production systems

The step is the elementary structure of the model. By combining steps opportunely, it is possible to adapt the model to different manufacturing scenarios. Operation or Step is the fundamental array. By following a bottom-up logic, other arrays can be found in characterizing every manufacturing system. Basically, every Operation is the final branch of a nested series of arrays.

The fundamental arrays (or attributes in the model) are: Operation (STP), Operator (OPE), Workstation (WST), Product (PRO), Process (LIN).

By resuming the full complexity of the problem, the model can potentially simulate either a single operation and multiple parallel processes as multi-sequences of multi-product sequences of multi-operator sequences of operations.

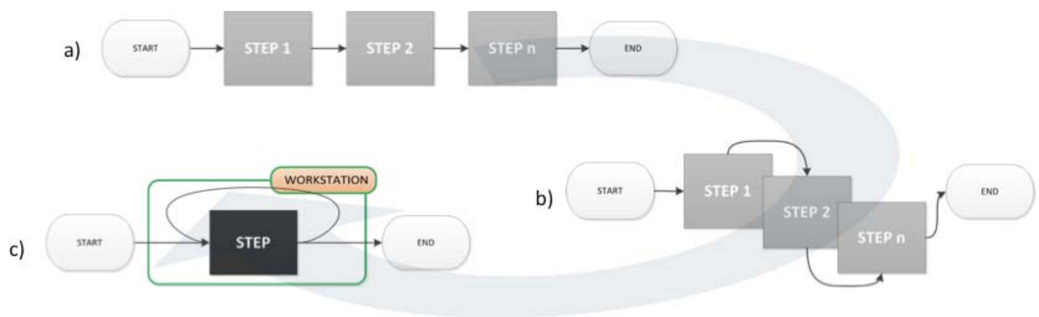


Figure 2. The idea under the model. From a horizontal to a vertical approach.

But, independently of the complexity of the model, every manufacturing process can be represented by the same vertical structure. Only the number of stacked layers varies in the model and array values univocally mark every layer. In fact, there is only one layer describing every operation "STP" made by the operator "OPE" on workstation "WST" on Line "LIN" producing the product "PRO".

The compressive logic of the model is maintained throughout every array perturbation, but a different impact of the arrays is evident: by acting on the Operation or Workstation arrays (Figure 2) the model is compressed towards a vertical structure where time-sequential events can be represented throughout a sort of spiral made by loops. Other arrays act differently even if they leave the model unaltered. Arrays Operator, Product and Process are suitable to describe events happening in parallel, not time-sequential. In designing such a model, it must be considered that some arrays vary within stacks of layers, whilst some other arrays vary between stacks only.

3. The DES model: un-parameterized elementary structure

The parametric DES model has been designed to develop a flexible working environment, enabling to effectively model and configure manufacturing cells, lines or more complex and advanced systems [17]. The model has a flexible structure able to allocate data and options needed to address a layer to a specific operation: this process has been called parameterization. The scheme of the model retraces the layer-based architecture previously introduced. First, it is possible to note that every layer is composed of functional blocks, sub-elements composed by programmed group of events and whose combination permits to cover multiple cases. Secondly, there are different types of layers. Some parameters, in fact, can be required only to characterize a sequence of operations, not every operation it is composed of.

As shown in Figure 3, the basic architecture of the model can be represented by a set of defined functional blocks. These blocks will be visited by an entity as many times as the number of layers the manufacturing system is composed of. The architecture is a generic bi-dimensional un-parameterized elementary layer. The user through the configuration process can determine the third dimension. The model is composed of three types of block, depending on their arrays. The core of the model is represented by Operation-dependent functional blocks. In order to describe an operation (stp) made on a product (pro) by an operator (ope) on a workstation (wst) as part of a process (lin), an entity has to evolve by reading the events stored in these functional blocks. Every block is composed of DES modules. Every operation specification has to be filled in by an external user accessing the model, but the model by itself has to foresee the data input.

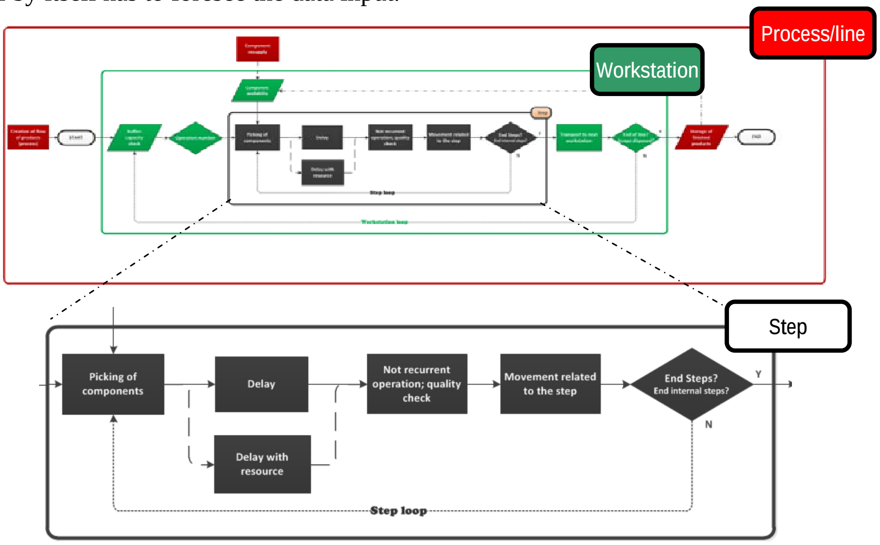


Figure 3. A functional block diagram of the parametric model and "Operation" functional blocks.

This set of functional blocks has been designed to receive all the parameters and options needed to define:

- operation description and type;
- picking of a component/subassembly;
- delay for a manual operation and semiautomatic operation;
- machining delay and motions;
- operation frequency;
- quality control.

Obviously, the model has to foresee a lot of combinations, but the user would have access only to those needs to tailor the operation. Once the entity has read all the information contained in these blocks, an opportune module controls that if there is another following operation the operator has to perform on the workstation (End steps). Additional blocks are needed to store the parameters and options used to characterize the workstation. These functional blocks are mainly referable to describe the buffer between workstations, the number of operators, the connection between workstations and the setups (Figure 3).

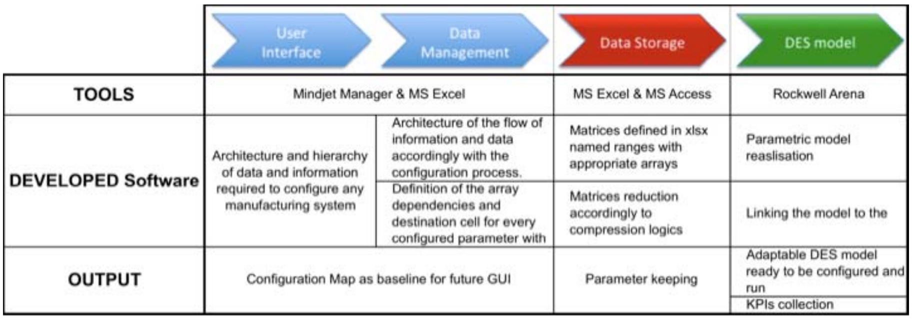
3.1 Parameterisation and arrays dependency

The parameters are the values needed to configure the model or, better, to personalize every layer the model is composed of, accordingly with the complexity introduced by the user. Every parameter depends on some of the fundamental arrays. Every array is labeled on the entity as an attribute and the combination of the attribute values permits to understand exactly on what layer the entities are lying during the simulation. Mapping the parameters and their dependencies to the arrays are a key task in realizing a flexible model. Parameters are required accordingly to options chosen when characterizing the layers. Listing the parameters not only help to design the configurability of the model, but it highlights potential integration with other software and/or data sources. This is the case of reliability & maintainability data and information. Since generally R&M parameters are difficult to estimate [18], it would be desirable to interface a DES model with dedicated tools [19]. Thanks to the parameterisation, this process can be facilitated.

3.2 Configuration process, data management and compression

The development of an adaptable model needs to face problems and difficulties that exceed the DES boundaries. Thus, not only the parametric model has been realized, but also many efforts have been necessary to design how the user can interact with it and how data can be automatically sent to the model. From a user point of view once realized and validated, traditional DES models have only to be run and results to be gathered. But, the advantages of the parametric model do not deal with simulation capabilities only, but the model has to provide a high adaptability and configurability. Consequently, the use of the model should comprehend data receiving by any user unaware of DES capabilities, opportune data management; model execution, analysis and results [20].

As described in Figure 1, the model has to foresee data receiving from external users. The user should have been guided throughout many options in order to configure its manufacturing processes with the less amount of information required. The software development does not comprehend only the DES model but even the architecture of a web based user interface and the mapping of the information and data flow to feed the model (Figure 4).



	User Interface	Data Management	Data Storage	DES model
TOOLS	Mindjet Manager & MS Excel		MS Excel & MS Access	Rockwell Arena
DEVELOPED Software	Architecture and hierarchy of data and information required to configure any manufacturing system	Architecture of the flow of information and data accordingly with the configuration process.	Matrices defined in xlsx named ranges with appropriate arrays	Parametric model reaslisation
		Definition of the array dependencies and destination cell for every configured parameter with	Matrices reduction accordingly to compression logics	Linking the model to the
OUTPUT	Configuration Map as baseline for future GUI		Parameter keeping	Adaptable DES model ready to be configured and run
				KPIs collection

Figure 4. Software development: the DES model and interfacing tools.

The backbone of the user interface (UI) has been described by developing a mind-map. The map summaries how a web application for parametric model configuration should be and, appearing like a tree diagram, it shows how any option can lead the user to different set of parameters. Logically, the map has to be recursively covered by the user, until he has completed the configuration process and every datum has been sent to the DES model. There is a specific hierarchy in mapping the configuration process. The architecture is quite complex and the map has a big size. Whilst the model acts on a bottom-up logic (from operations to processes), the configuration must be top-down: first the process has to be defined and then described by populating the parameters governing it at the desired level of detail (not necessarily the single operation one). Note that Figure 4 shows the macro level configuration process, hiding the full complexity of the map. The configuration has to go in detail and a lot of information would be needed to complete the task.

3.3 Data Storage and DES Model

Both parameters and attributes coming from the configuration have to be stored in databases ready to be used by the DES model. Currently, MS Excel sheets have been chosen to file data and information, because of its easiness of use and potential integration with other data sources from other software tools. Nevertheless other DB solutions can be adopted. First the matrices have to be defined as unique named ranges. Then the flow of information from the configuration process to the sheets must be detailed in the map, since an user interface needs it to be properly addressed. Every time the configuration process needs to store a datum, the path is described in the map with additional details (Equation 1).

$$\text{Type: } \mathbf{matrix\,nr(Row,Column)}[\text{Variability,Uncertainty,Unit}] \quad (1)$$

Where:

- Type is M when the program foresees the user would insert data manually; S if an automatic calculation is expected on branch selection. Note that S type includes the equations for updating or calculating variables with or without storage of data;
- matrix nr (Row, Column) defines, when needed, the specific matrix (named range) where datum has to be stored. Row and column univocally define the coordinates of the storage cell. Row and column are commonly expressed as an array or combination of arrays;
- Variability index takes into account if the user can insert deterministic values (D) or a perturbation is allowed (V);
- Uncertainty index defines if the user is confident with the datum inserted (D) or not (U);
- Unit index permits to personalize the Measurement Unit of the datum (UM) or not (D).

In order to understand the compression logic of matrices an example is proposed. It is supposed a simple manual line (only one operator for every workstation) composed of three workstations (WST attribute 1, 2 and 3 respectively), Figure 5. Product A (Attribute PRO equal to 1) process is 1-2-3-Exit (Process 1), whilst Product B process is 1-3-Exit (Process 2). Suppose two operations for every workstation and the parameter operation delay (i.e. cycle time) equal to 10 seconds in any case (Figure 5).

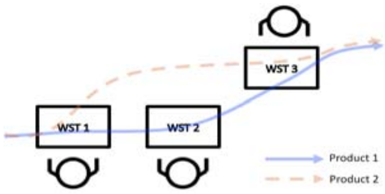


Figure 5. Layout scheme of a manufacturing process.

After the completion of the configuration process the following key matrices are filled in (see Figure 6) and dimensioned as follows:

- Matrix 50 (WST,PRO) reduced size is 3x2. The matrix records how many operators are willed for every product on every workstation;
- Matrix 38 (LVL,LIN) reduced size is 3x2. The matrix links the workstation identifier to the level of every process
- Matrix 39 (LVL,PRO) reduced size is 3x2. The matrix records the cumulated starting position of the operation set for every workstation (here level) a product will visit
- Matrix 3 (LVL,PRO) reduced size is 3x2. The matrix records the maximum number of recurrent operations required by any operators on a workstation for every product.
- Matrix 9 (POS,PRO) size is 6x2. The matrix records the operation delays.

The DES model has been designed to receive these matrices accordingly with compression rules. Thus the DES language is ready to decrypt the rules of compression and the model is able to read data opportunely depending on the layer an entity is lying on. For the example above, the delay time of a generic layer is stored in Matrix 9 in the cell:

$$(\text{Matrix39}(LVL,PRO) + (OPE-1)*\text{Matrix3}(LVL,PRO)+(STP-1), PRO) \qquad (2)$$

In the proposed parametric model, the manual delays are managed in specific modules (*delays area*, Figure 7). In particular the entities entering in the module *STP parallel delay* read the values of the manual delay associated to the layer. The value is stored in Arena as a 2D Array Variable linked to the MS Excel recorded "STP_delayparVA_WOSPU", that is the dimension of the Matrix 9 at the end of the configuration (6x2 matrix). By the entities entering the module is delayed by the value (it could even be 0) written in a cell of the 2D variable. The exact cell where the value is stored depends on the decompression rules (see Equation 1). Figure 7 also shows the logic that the model uses to receive data from external sources and to characterize the parameters (here the Operation delay, accordingly to the example of Figure 5). Now, the simulation can be performed. The model has been tested and validated considering different scenarios (single manufacturing cells, assembly workstations, portion of assembly lines, lines with sublines), comparing the simulation results with the production system indicators (i.e. lead time, throughput, value added activities, …).

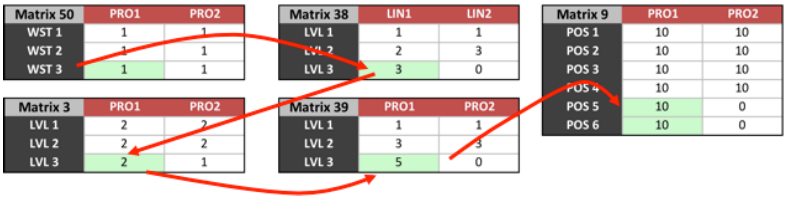


Figure 6. The matrices of the example filled in with data and reduced.

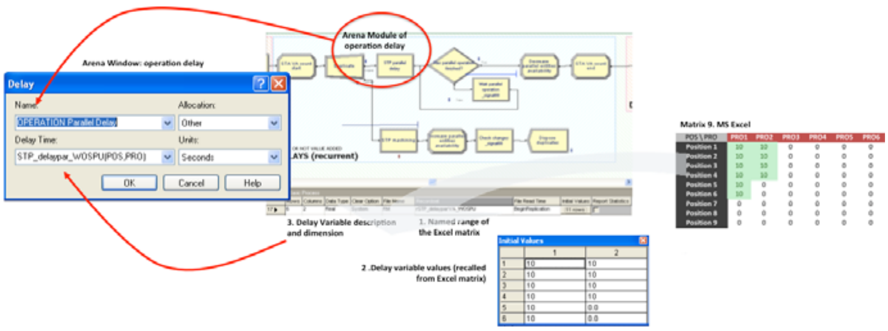


Figure 7. The DES modules composing the “delays area”, with focus on the Operation delay module.

4. Conclusions

This paper described a novel logic and approach in modeling and simulating manufacturing and transactional systems using DES tools. An adaptable and flexible DES-based parametric model is developed and ready to be used. The model is able to represent different industrial scenarios and it can be managed by end users without specific expertise in simulation tool applications. The study shows the whole roadmap and logic of the novel approach from the end user’s data capture to simulation run.

The novel approach is based on elementary structures with a vertical multilayer architecture that permits entities to access and loop in multiple times. The logic under the parametric DES model uses elementary structures, achieving the goal of adaptability not through a redundant design, but thanks to an iterative reading of information. Both parameters and attributes coming from the configuration have to be stored in databases ready to be used by the DES model. Currently, MS Excel sheets have been chosen to file data and information. Data and information classified in matrices are uploaded in the DES model in order to run the simulation.

The main achievable benefits are the application of DES tools without holding specific knowledge on simulation tools, saving time and avoiding costs related to training, modeling efforts and analysis.

The parametric model has been designed to develop a flexible working environment, enabling to effectively represent and configure manufacturing systems. The complexity of data collection and classification has required many efforts to design how the user can interact with the system and how data can be automatically sent to the model. In fact, the model has to foresee data receiving from external users nevertheless the user should have been guided throughout many options in order to configure its manufacturing processes with the less amount of required information.

The model has been designed to cover many layouts. Currently, the model has been configured to simulate single manufacturing cells, assembly workstations, portion of assembly lines, lines with sublines and other complex variants of lines served by conveyors. The next steps of the study will consist on further validations on more complex systems (i.e. considering material handling supply) and layouts in order to understand full potentialities and constraints.

5. Acknowledgements

The authors acknowledge the support of the European Commission through the 7th Framework Programme under NMP-2008-3.4-1 Rapid Design and Virtual Prototyping of Factories (COPERNICO: contract number 229025-2).

6. References

[1] W. Wen, Y.H. Chen, I.C. Chen, "A knowledge-based decision support system for measuring enterprise performance", Knowledge-Based Systems, vol. 21, no. 2, pp.148–163, 2008.
[2] Hong-Youl Lee, "Finding Semantic Errors in the Rule-base of Production Systems, and Reasoning with Insufficient Input Data? Petri-net-based Approach", JCIT: Journal of Convergence Information Technology, vol. 3, no. 4, pp. 59 - 64, 2008.
[3] F. Aggogeri, G. Barbato, E.M. Barini, G. Genta, R. Levi, "Measurement uncertainty assessment of Coordinate Measuring Machines by simulation and planned experimentation", CIRP Journal of Manufacturing Science and Technology, vol. 4, no. 1, pp. 51-56, 2011.
[4] Chun-Hsiung Lan, Kuo-Torng Lan, Tzu-Ying Hung, "The Layout of Limited Factory Space by Hybrid Methods", IJACT: International Journal of Advancements in Computing Technology, vol. 4, No. 10, pp. 54 - 60, 2012.
[5] E. Gentili, F. Aggogeri, M. Mazzola, "The effectiveness of the quality function deployment in managing manufacturing and transactional processes", ASME International Mechanical Engineering Congress and Exposition, Proceedings, vol. 3, pp. 237-246, 2008.
[6] Kunpeng Yu, Yu Yang, Jingbo Guo, Tao Yang, "Research on Production Plant Layout Optimization Based on Improved Genetic Annealing Algorithm", IJACT: International Journal of Advancements in Computing Technology, vol. 4, no. 5, pp. 329-336, 2012.
[7] A.A. Tako, S. Robinson, "The application of discrete event simulation and system dynamics in the logistics and supply chain context", Decision Support Systems, vol. 52, pp. 802–815, 2012.
[8] A. Borboni, F. Aggogeri, N. Pellegrini, R. Faglia "Precision Point Design of a Cam Indexing Mechanism" Advanced Materials Research, vol. 590, pp 399-404, 2012.
[9] A.A. Tako, S. Robinson, "The application of discrete event simulation and system dynamics in the logistics and supply chain context", Decision Support Systems, vol. 52, pp. 802–815, 2012.
[10] S. Robinson, "Discrete-event simulation: From the pioneers to the present, what next?", Journal of the Operational Research Society, vol. 56, no. 6, pp. 619–629, 2005.
[11] A.A. Tako, S. Robinson, "Model development in discrete-event simulation and system dynamics: An empirical study of expert modellers" European Journal of Operational Research, vol. 207, pp. 784–794, 2010.
[12] S. Myeong-Jo, K. Tae-wan, "Maneuvering control simulation of underwater vehicle based on combined discrete-event and discrete-time modeling", Expert Systems with Applications, doi: 10.1016/j.eswa.2012.05.099, 2012.
[13] S.R. Nidumolu, N.M. Menon, B.P. Zeigler, "Object-oriented business process modeling and simulation: A discrete event system specification framework", Simulation Practice and Theory, vol. 6, pp. 533–571, 1998.
[14] J.W. Herrmann, E. Lin, B. Ram, S. Satin, "Adaptable simulation models for manufacturing", In Proceedings of the 10th International Conference on Flexible Automation and Intelligent Manufacturing, College Park, Maryland, June 26-28, vol. 2, , pp. 989-995, 2000.
[15] M. Mazzola, E. Gentili, F. Aggogeri, "Improvement through process integration using a simulative, dynamic method", Int. Journal of Manufacturing Technology and Management vol. 14, no. 3/4, pp. 396-409, 2008.
[16] D.J. Sheskin, D.J., Handbook of Parametric and Nonparametric Statistical Procedures, fourth ed. Chapman and Hall/CRC, Boca Raton, 2007.
[17] W.D. Kelton, R.P. Sadowsky, D.T. Sturrock, Simulation with arena. 4th ed. New York: McGraw-Hill, 2007.
[18] M. Tiboni, A. Borboni, M. Mor, D. Pomi, "An innovative pneumatic mini-valve actuated by SMA Ni-Ti wires: design and analysis" in Proceedings of The Institution ff Mechanical Engineers Part I-Journal of Systems and Control Engineering, vol. 225 (I3), pp. 443-451, 2011.
[19] F. Aggogeri, F. Al-Bender, B. Brunner, M. Elsaid, M. Mazzola, A. Merlo, D. Ricciardi, M. de la O Rodriguez, E. Salvi, "Design of piezo-based AVC system for machine tool applications" Mechanical Systems and Signal Processing, doi: 10.1016/j.ymssp.2011.06.012, 2011.
[20] J. Schryver, J. Nutaro, M.J. Haire, "Metrics for availability analysis using a discrete event simulation method", Simulation Modelling Practice and Theory , vol. 21, pp.114–122, 2012.
[21] M. Gyimesi, "Web Services with generic simulation models for discrete event simulation", Mathematics and Computers in Simulation, vol. 79, pp.964–971, 2008.