Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 38 (2019) 1095–1102

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

29th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2019), June 24-28, 2019, Limerick, Ireland.

Automatic generation of digital twin industrial system from a high level specification

Julio Garrido Campos*, Juan Sáez López, José Ignacio Armesto Quiroga, Angel Manuel Espada Seoane

Automation And System Engineering Department, Vigo University, E.E.I Campus Lagoas Marcosende, Vigo, 36200, Spain.

Abstract

A framework for the generation of industrial digital twins is presented in the paper. The framework supports industry automated systems preliminary design development, but also supports the following detailed designs implementation and final systems exploitation phases. The main problem is that requirements for first development phases are much more generic than those required for the later phases. The framework faces this problem by avoiding too detailed specifications for the digital twin generated software, but, at the same time, it takes advantage of the specific applications developed for each industrial implementation where that specificities are taken into account: the final control application and the management application. By properly linking both: the more generic digital twin and specific software applications specifically generated for the industry system, the framework may be ready to be used soon at the early development stages, but also may be used for detailed analyses at late booting and maintenance industry system phases. The system has been specialized in industrial transportation and warehouse systems. The paper presents an example of application for this kind of systems.

Keywords: Industry 4.0; Digital Twin; Simulation.

* Corresponding author. Tel.: +34-986-812610.
E-mail address: jgarri@uvigo.es

2351-9789
10.1016/j.promfg.2020.01.197

1. Introduction

Industry digitalization looks for "a smart factory model where computer-driven systems monitor physical processes, create a virtual copy of the physical world and make decentralized decisions based on self-organization mechanisms"[1][2]. This virtualization is defined as a virtual copy of the Smart Factory created by linking sensor data with virtual plant models and simulation models [3].

This virtual copy of the Smart Factory, or digital twin [1] [4], may be seen as an extension of simulation systems [5], and it has already been revealed itself as an excellent tool to improve systems design, to improve systems exploitation efficiency, as well as a maintenance support tool. However, requirements at early stages of systems development, for preliminary design analysis, differ from those for detailed systems designing, from those for maintenance support system, from those for process knowledge feedback and so worth. One of the main difference is that in the first, a simulation system is needed very early in the process to be able to evaluate different system configurations. But, for process knowledge feedback and maintenance support, detailed simulation, communication and animation capabilities are more important. It is not ease to get a good balance between a fast and easy to generate-almost from scratch-system simulation, and a more detailed and much more customized simulation with the ability of becoming a real virtual twin as well [6].

Moreover, commercial frameworks to build industrial digital twins provide the technology tools to support their implementation [7] [8], but they do not provide high level models of specific devices and facilities, which are essential for detailed simulation. Also, a complete digital industrial twin system needs the simulation of the specific mechanic and machine control by using, for instance, discrete event simulation systems [9] [10], but also the simulation of the management and planning processes [11] [12].

While one option could be having different systems (digital twin systems) for each stage, the paper presents, instead, a system which is generated from scratch in a very short time, to be successfully used as a design support at early process design stages, and as a system performance improvement tool and a maintenance aid tool during the system exploitation phase. The system has been specialized in industrial transportation and warehouse systems which, although have a finite number or building objects, they have an infinite range of final configurations, very different one form each other.

The paper evaluates in next section simulation platforms suitable to be used. Section 4 presents the digital twin industrial system generation framework. Finally, section 5 is an example of application.

2. Target systems: Industry transportation systems for internal logistics

The system has been specialized in industrial transportation and warehouse systems made of linear tracks (such as conveyors or rails) to move product containers (such as cars with hooks to hang a product, see Fig. 1 Right) between points and store areas made of several "parallel" bars where the containers are stored in FIFO queues (Fig. 1 Left). Those bars have a stope device on the exit side to control the output of the containers one by one.

Transportations tracks and storage areas, allow the automatic storage and retrieval of products. The movement of the transportation units is achieved thanks to "Powered" tracks, lift tracks (elevators) and gravity tracks. In "Powered" tracks, transportation units move through a powered puller, but they may be released (Free) from the power, stopped and accumulated in FIFO queues at any point. Lift and elevators move cars from one level to another, while in gravity (inclined) tracks, cars move freely. Systems have entry points (where products coming from outside are fitted into the transportation elements), elevator, connecting paths, deviations, and one or more exit points.

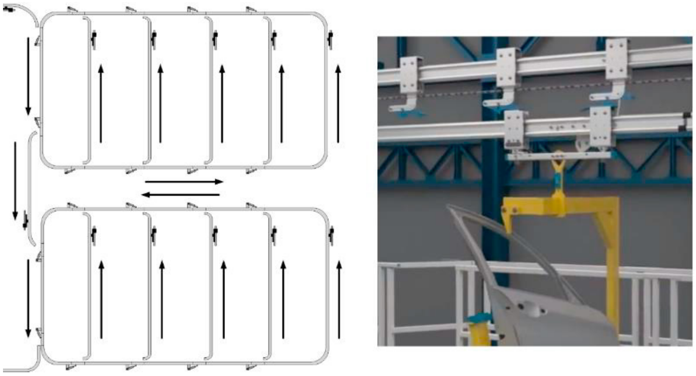

Fig. 1. (Left) storage area; (Right) transportation trolley "Power&Free".

From the machine control perspective, there are two logic levels. One low level traffic control for traffic and navigating and one high level flux control. The traffic control level is performed for each system element individually, taken into account local information, while for the flux control the knowledge of the state of the whole system is needed. Traffic control principles do not change from one industry facility to other of the same family, but flux control may be quite different in each one, although regular fluxes may be equal in all. Therefore, traffic control is more suitable to be generalized than flux control.

3. Digital twin generation framework

Different simulation systems have been evaluated in order to automate the generation of this type of facilities [13]. In [4] a review of discrete events simulation systems for flexible manufacturing systems can be found. Within the research involving this paper a number of them have been analysed to evaluate their capacity to be part of the digital twin automatic development framework.

Arena is a simulation environment with a long tradition in the industry [14]. However, it is not easy to program complex functionalities, it lacks a high level programming language, it is not easy to implement automatic generation of models and it has poor graphic capabilities to present animated results. Simulink has also checked [15]. It has a high level textual programming language. However, it is poorly object oriented, which makes it difficult to make scalable models and to automate it generation.

SIMIO environment [16], despite having a good 3D graphic environment, it is hard to program complex functionalities. It lacks of a high level programming language and it is also difficult to implement automatic generation of models (see Fig. 2).

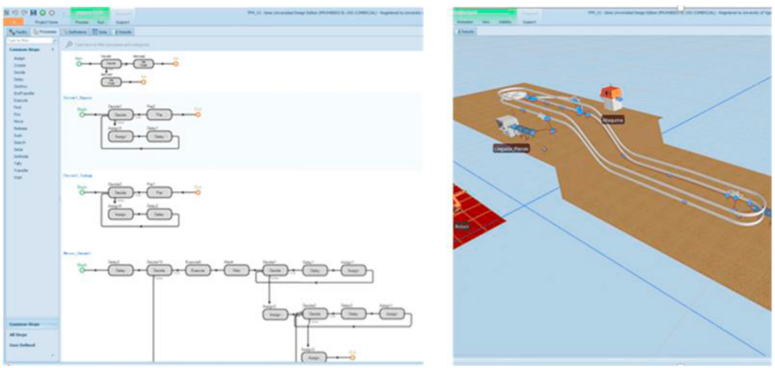

Fig. 2. Example of SIMIO environment details and system model.

PlantSimulation is one of the "finalist" systems [17]. It has all the necessary features to achieve the objectives of the project. Fig. 3 repeats an automatically generated model from a xml specification.

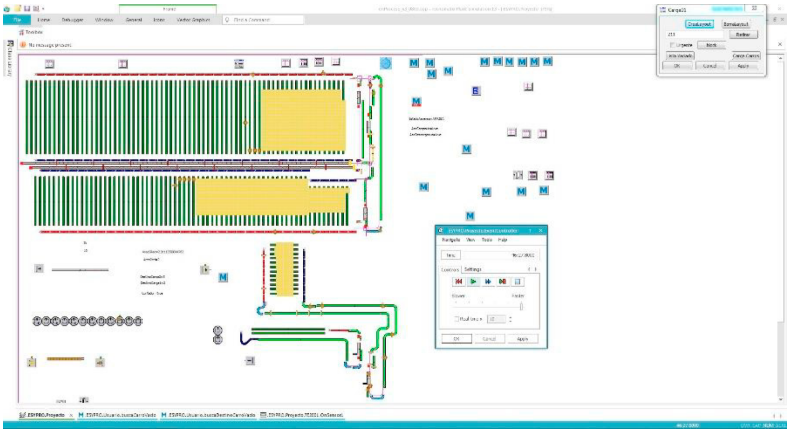

Fig. 3. Example of Plant Simulation generated model.

FlexSim has been the second of the systems considered appropriate for use as a simulation system within the framework of this paper [18]. As with PlantSimulation, it provides all the features necessary to achieve the project's objectives (Fig. 4).

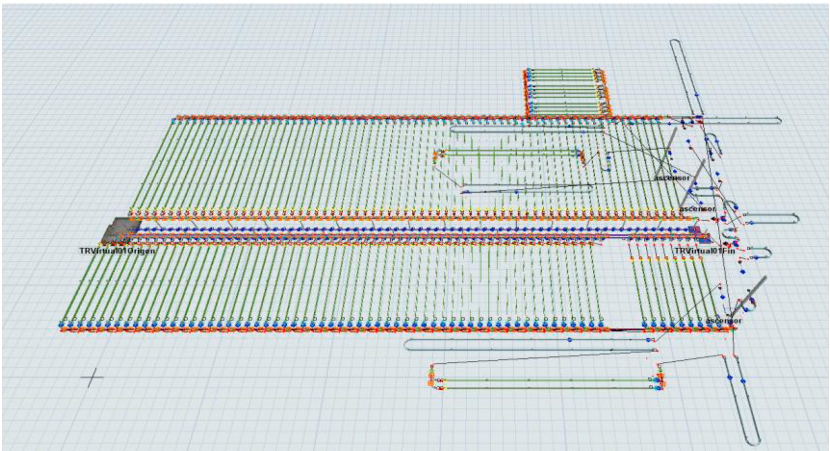

Fig. 4. Example of FlexSim generated model.

Table 1 summarizes main advantages and drawback of the main software considered.

Table 1. Advantages and drawback of the main software considered.

Software systems tested	*3D monitoring capabilities*	Object Oriented	*High level progr. languages*	*complex data struct.*	*Integration*
Arena	Low	Low	Low	Low	Low
Simio	Low	Low	Low	Low	Low
Simulink	Low	Low	High	High	Medium
PlantSimulation	High	High	High	High	High
Flexim	High	High	High	High	High

4. Digital twin generation framework

Automatic software generation frameworks are an old concept. CASE tools (Computer Aided Software Engineering) [19] [20] to support the automatic generation of simulation models are a common solution used to support the conceptual design generation process, by generating early full working simulation of systems. But also, CASE tools have been used for long to support final system implementation process, by generating the kernel or the general infrastructure of solutions from a simple specification.

The procedure is similar in all CASE tools. It begins with a basic system specification with its fundamental structure, but with enough detail to generate a first Lay-out. The CASE tool runs through this specification and instantiates the equivalent elements in the generated model. To do this, it uses pre-developed software blocks of the main elements that include all the necessary functionalities for their subsequent implementation. These blocks can be found in different formats: object libraries, code blocks, etc. This results in a first realization of the system. But doing this with complex systems, with a very wide variability, and a high degree of specificity from one project to another is not easy. In these cases, even if the main constructive and functional elements are maintained, it is necessary to add handmade code to implement these specificities.

Implementations can take different paths from this point on. One is to introduce the next handmade specific code directly on the generated model, assuming that there won't be a further re-generation. For the case that subsequent generations are required due to configuration changes, it is possible to use systems based on templates or inheritance systems, which keep the handmade code identified to be incorporated again if there is a subsequent regeneration. This solution may be suitable for systems that develop very specific and complex digital twins, which are difficult to repeatable. Specific handmade software continues to be developed and incorporated to the simulation model. But the extensions that would have to be done in the framework to be able to implement merge capabilities (to incorporate all code added to a previous generation if a new generation is performed) introduce complexity in its architecture, and make the framework difficult to be maintained.

Other option, the one presented in this paper, is to develop a system that allows, from very early on, in the preliminary design phase, generate a complete simulation that supports the first phases of system design. But at the same time, and without the need to develop and add any subsequent handmade code, it must be a valuable support for the exploitation phase of the industrial system: for maintenance, for improvement of system operation, etc. For these phases, it is necessary to work with a very realistic digital twin of the system: faithful twin of control, faithful twin of management decisions, and faithful twin of real situations. But requirements at early stages of systems development, for preliminary design analysis, differ from those for detailed systems designs at later industry systems development stages. To get a simulation useful for these further phases where systems are already working, much more logic need to be added to the simulation model resulting from a CASE tool to cover each system specifies. The solution has been that, instead of implementing those specificities in the simulation system, take advantage of the applications developed in the implementation of the real system in which they have been necessarily taken into account: the control program and the real high level decisions and administration system.

Fig. 5 depicts the architecture of the whole system. The upper part of the system represents the automatic generation of the operating system, which is not the main subject of this article, while the lower part is the automatic generation of the simulation. Another CASE tool uses the same general system specification to generate it.

The operating systems has three levels: The low for real time control (PLC control), a medium level foe Manufacturing process control and Man Machine Interface (MES/MMI) and a high level (the business level represented as a Data Base) from where primary orders for the system came from. At development time, the CASE tool generates the main structure of the tree systems from a general system specification. These three generated main structures are generated with the communication mechanism already set. Finally, right part of the figure represents the solutions that have been particularized by adding handmade code after the automatic generation. At execution time, the three systems maintain a copy of the "transportation unit's location" (MAP). The PLC Control uses it to generate the navigating orders (for instance, to decide to open a stop device and to allow a transportation unit to move) or not; The MES/MMI uses the MAP to generate flux orders (for instance, to decide is to initiate a flux to transfer units form one area to other of the warehouse); and the high level order uses the MAP lo look for where is a specific product reference at the warehouse.

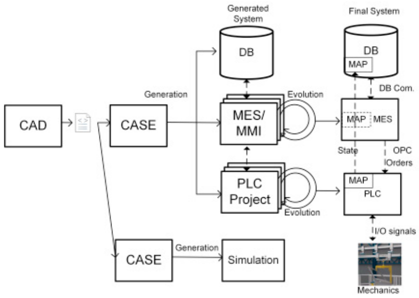

Fig. 5. System generation process. Top: Development of execution system. Bottom: Development of Simulation System.

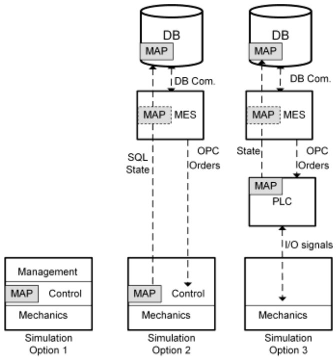

Fig. 6. Digital twin uses options. Left: Option 1, full simulation; Center: Option 2, Mechanic and Control simulation, MES real; Right: Option 3, Control and MES real and mechanic simulation.

The MAP is the key element in to implement the integration of the tree systems. The PLC Control system maintains a map of the elements of transport in real time to assist in making decisions on traffic regulation. The PLC performs the control over each basic element of the system (retention elements of the mobile transporter devices and of the deviations and elevators) based on the management of the system traffic. The orders to these elements are decided by the control system depending on the destination assigned to each mobile-car transport element. On the other hand, the flow management system maintains a copy of the map of transport elements used for the generation of the highest level decisions for which it is necessary to take into account the global state of the entire transport system and storage. For example, if the management system wants to perform a load (storage) operation, it analyses the current state of the system and the current or planned one. That is, it reviews the map of transport elements of the system (those that are stored and those that are in transit) and queues of requests to the control system that have not yet been addressed, to decide the destination that will be assigned to the transport element involved in the loading operation, which will be communicated to the control system. If, for example, a reference request (picking order) is made, the management system will search the system map for transport elements that are in storage areas and that contain the same reference, and decide which of them it sends to the exit area, and generates the corresponding orders to the control system

Figure 5 low branch represents the generation of the simulation system. This is explained in the following Fig. 6, where the three levels of development and use of the system are represented:

- Option 1: Self-contained simulation system. In this case, everything is simulated, that is, the mechanical part is simulated, control is simulated, and management is simulated.
- Option 2: Evaluation of the management system. In this case, the simulation system is limited to simulating the control program and the mechanics, but the high-level orders come from the real management system (management of high-level flow routes).

– Option 3: Finally, this system is used to evaluate the real control system. Here the simulation is limited to reproducing the mechanics according to the orders generated by the real control system, and also simulating the information -signals- with which the mechanic part feeds the control system back.

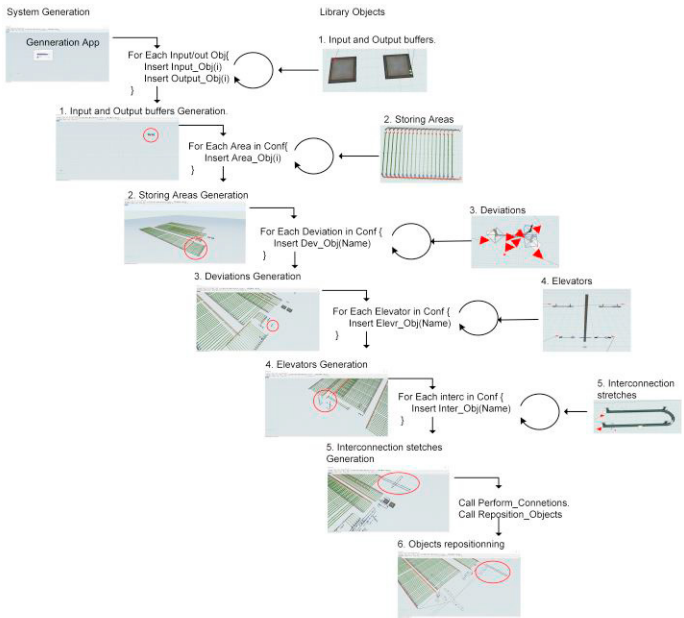

Fig. 7. Example: Digital twin automation generation sequence.

In option one (see Fig. 6), the simulation system implements the map of transport elements. It simulates both the traffic control system implemented in PLC (based on the MAP), and the management of high-level generic flows (based on the same or only MAP).

In option two, the simulation system maintains its simulated location MAP, which is communicated in the same way as the PLC, to the high level flow management systems. The "real" management system takes its decisions transparently, without knowing if the locations of the transport elements (MAP) come from a real or a simulated control system.

In option three, the simulation does not maintain any MAP of locations and it is limited to simulate the mechanical response and sensors to the orders of action of the control system, simulating equal movement of the transport elements. That is, if for example, the control system generates an order to open a transport carriage retention element, then the simulation system reacts by simulating this opening, generating a simulated signal of the activation of the element's exit sensor. In addition, it is possible to specify an initial starting situation of the system-map of the system-in all of them to reproduce real or desired starting situations.

Fig. 7 is an application example. It represents (in pseudocode) the algorithm of system generation. On the left, the individual objects involved are represented, and on the right are the different stages through which the generation of the system goes. This process takes few seconds provide the system high level specification has been done. That is a XML file with high level information of the industrial system (number of accumulation areas, number connected tracks and topologic information, etc.). It is made by hand, although an aid tool has been developed. With it, it would take 2 hours to complete one of these specification files for a complex transportations and warehousing system. The result of Fig. 7 generation, is a full working simulation of the system. It may be already used to check the system design. But, when the specific handmade control code and the specific MES where developed (both based on the kernel objects generated by the CASE tool using the same XML specification file), the digital twin (options two and there) arises.

5. Conclusions and future work

The developed framework has proven its effectiveness both in the preliminary development phases and in the operation phases. They have been contrasted through the simulation of systems already in operation. Their results have allowed to immediately observe bottlenecks and problems that occurred in their real equivalents. One of the main conclusion comes from real experience: there is a point where further customization provokes a loose of efficiency of the system. From that point it will be very difficult to regenerate the system, if there has been a substantial change in the overall configuration. After that point, code development falls on the real system: the Controller and the MES application. That point happens just when specific handmade code is started to be developed. Although the Control System and the MES would have also been simulated, it requires a big effort, which is even harder in the case of the case of industry systems with differences between one implementation and another. The solution presented on the paper avoids this extra work by using the real control and MES instead of simulate them, using the run time system state (the MAP–or real time location of transportations units–for the case of transportation and warehousing systems) as the integration mechanism. For other kind of industry systems, the corresponding MAP equivalent would be used.

The integration with both by the simulator, in real time and pre-developed automatically, has allowed to verify, maintain and improve the algorithms executed in both. To this end, it has been very important to be able to transfer to the simulator the real state of a system situation at a given moment - through a mechanism to transfer the real system map to the map in simulation.

References

[1] Davies, R. (2015). Industry 4.0: Digitalisation for productivity and growth. EPRS | European Parliamentary Research Service.
[2] Yadav, A., and Jayswal, S.C. Modelling of flexible manufacturing system: a Review, International Journal of Production Research, (2017) 56(7).
[3] European Parliament. Directorate general for internal policies. Policy department: Economic and scientific Policy. Industry 4.0. (2016).
[4] McFarlane, D., Sarma, S., Chirn, J.L., Wong C.Y., and Ashton K. Auto ID systems and intelligent manufacturing control. Engineering Application of Artificial Intelligence, 16 (4), (2003) 365-376.
[5] Qina,J., Liua, Y., and Grosvenora, R. A Categorical Framework of Manufacturing for Industry 4.0 and Beyond. In: Procedia CIRP 52, (2016) 173–178.
[6] Krenczyk, D. Automatic Generation Method of Simulation Model for Production Planning and Simulation Systems Integration. Advanced Materials Research. 1033-1036, (2014) 825-829.
[7] Siemens. Twins with potential, The Magazines. Available at: https://www.siemens.com/customer-magazine/en/home/industry/digitalization-in-machine-building/the-digital-twin.html. [Accessed 10 Dec. 2018] (2018).
[8] Anylogic. Simulation Software Comparison: Discrete Event Simulation Competitors. Available at: https://www.anylogic.com/resources/white-papers/simulation-software-comparison/. [Accessed 10 Dec. 2018] (2018).
[9] Yadav, A., and Jayswal, S.C. Modelling of flexible manufacturing system: a review. International Journal of Production Research, 56(7), (2016) 2464-2487.
[10] Ghani, U., Monfared, R., and Harrison, R., Integration approach to virtual-driven discrete event simulation for manufacturing systems. Journal International Journal of Computer Integrated Manufacturing, (2015) 28(8).
[11] Ferro, R., Ordóñez, R.E.D., and Anholon, R. Analysis of the integration between operations management manufacturing tools with discrete event simulation. Production Engineering, Volume 11, 4(5), (2017) 467–476.
[12] Krenczyk, D. Integration of manufacturing operations management tools and discrete event simulation, IOP Conf. Series: Materials Science and Engineering (2018).
[13] ORMS. The Institute for Operations Research and the Management Sciences Simulation Software Survey, https://www.informs.org/ORMS-Today/OR-MS-Today-Software-Surveys/Simulation-Software-Survey. (2018)
[14] Arena Simulation Software. Pittsburgh. USA. Rockwell Automation. (2018)
[15] Simulink. The MathWorks, Inc. Natick, Massachusetts, USA. Available at: https://www.mathworks.com/help/matlab/internet-file-access.html . [Accessed Dec. 2018] (2018)
[16] Simio. Pittsburgh, PA, USA. Simio LLC. Available at: https://www.simio.com/resources/technical-support/index.php (2018)
[17] PlatSimulation. Plano, TX, USA. Siemens Product Lifecycle Management Software Inc. Available at: https://www.dex.siemens.com/plm/plant-simulation. [Accessed Dec. 2018] (2018)
[18] FlexSim. Orem, Utah, USA. FlexSim Software Products, Inc. Available at: https://www.flexsim.com/flexsim [Accessed Dec. 2018] (2018)
[19] Arief, L.B., and Speirs, N.A. A UML Tool for an Automatic Generation of Simulation, Programs, In: WOSP '00 Proceedings of the 2nd international workshop on Software and performance. (2000) 71-76.
[20] Bergmann, B., Strassburger, S. Challenges for the Automatic Generation of Simulation Models for Production Systems. IN: SCSC '10 Proceedings of the 2010 Summer Computer Simulation Conference, Ottawa, Ontario, Canada, (2010) 545-549.