ISSN 2619-1865

Proceedings ICPR Americas 2020

edi UNS

COLECCIÓN CIENCIA Y TECNOLOGÍA

International Conference of Production Research-Americas
(ICPR-Americas). ISSN 2619-1865

Editorial de la Universidad Nacional del Sur
Santiago del Estero 639 – B8000HZK – Bahía Blanca – Argentina
Tel.: 54–0291–4595173 / Fax: 54–0291–4562499
www.ediuns.com.ar | ediuns@uns.edu.ar

Libro Universitario Argentino

Diagramación interior y tapa: Fabián Luzi

Bahía Blanca, Argentina, marzo 2021

Modelling the Dynamics of a Smart Factory

Marisa Analía Sánchez[1][0000-0002-8291-0491] – Daniel Rossit[2][0000-0002-2381-4352] – Fernando Tohmé[3] [0000-0003-2988-4519]

[1] Dpto. de Ciencias de la Administración, Universidad Nacional del Sur, Bahía Blanca, Argentina
mas@uns.edu.ar

[2]Dpto. de Ingeniería, Universidad Nacional del Sur, INMABB-UNS-CONICET, Bahía Blanca, Argentina
daniel.rossit@uns.edu.ar

[3]Dpto. de Economía, Universidad Nacional del Sur, INMABB-UNS-CONICET, Bahía Blanca, Argentina
ftohme@criba.edu.ar

Abstract. "Digital twining" is one the main ways of establishing data channels in cyber-physical systems using both the outputs of running a virtual model and real time data collected by sensors. The purpose to this paper is to outline the digital twin of a cyber-physical *production* system. We apply the System Dynamics paradigm to the case of a shop-floor factory devoted to cloud manufacturing. The digital twin uses data from the real production line, providing assistance to maintenance procedures triggered by inconsistencies between the real and the virtual processes.

Keywords: Digital Twin, Cyber-physical Systems, System dynamics, Simulation.

1 Introduction

One of the main components of the Industry 4.0 infrastructure are "smart factories", able to adapt their production processes to demand, assigning their resources more efficiently than traditional factories. The key technologies enabling smart factories are the Internet of Things [2], Cyber-physical Systems (CPS) [3] [4], cloud computing [4] and big data [5] [6].

These technologies allow the efficient integration of different functionalities of the control specification ISA-95 into a single production system. In terms of its five levels, all the processes that can be digitalized can be thus absorbed in Industry 4.0 systems [7] [8]. This integration can be both at horizontal and vertical levels. The vertical integration extends from the physical process level (level 0 of ISA-95) to Manufacturing Execution Systems (level 3 of ISA-95) [9]. These levels, handled by CPS, translate the physical events into digital data, creating a digital twin (DT) of the production system. The digital twin provides useful information about the real workload of the production system, which can be used in planning and business management process-

es [10] [11]. Managers can simulate the inner workings of the plant under different scenarios, gaining further information contributing to make better decisions. The ultimate objective of digital twins is to improve the operation and efficiency of manufacturing assets, reducing costs by forecasting future states and supporting advanced decision-making throughout the entire manufacturing lifecycle [12].

A DT of a production system is a type of simulation that allows the real-time control of production. Ding et al. [1] use the term "digital twining" to refer to the process of building a digital twin in the cyber world of physical objects and systems, establishing data channels for cyber-physical connection and synchronization. The implication of this definition is that a DT should use both simulated values and real time data collected by sensors. The simulation has the ability to compress time running ahead the real world process since the digital twin simulation clock is set at one relevant period ahead of real time. If at time t_r a sensor reports a machine failure that was not generated in the simulation when its internal clock was at t_r, the latter should be reset at t_r updating its state with this new information. This feature enables a DT to keep the simulation running at a par with the real production line.

The aim of this work is to describe a simulation model for a digital twin cyber-physical system. In order to define the guidelines for a smart plant environment, the modelling methodology of System Dynamics can be used to handle the uncertainties and nonlinear relationships among interacting system components [13] [14]. The complexity in the definition of the model may ensue from structural or dynamic aspects. Structural complexity refers to the number of components in a system, or the number of combinations involved in making a decision. Dynamic complexities result from the nonlinear and history-dependent nature of self-organizing and adaptive systems [15]. System Dynamics addresses these two kinds of complexity by postulating that the behavior of complex systems results from an underlying structure of flows, delays, and feedback loops [16]. The emphasis in System Dynamics is not on forecasting the future, but on learning how the actions in a period of time can trigger reactions in the future [17] [18].

The paper is organized as follows. Section 2 presents concepts drawn from the literature on manufacturing processes, providing information about the system to be modelled. Section 3 presents the running example that will illustrate the main notions presented in this article. Section 4 introduces Causal Loop diagrams, the basic graphic modelling tool needed to sketch the description of the system. Section 5 presents Stocks-and-Flow diagrams, which allow developing a full model of the system. Section 6 shows how this model provides the grounds to simulate different scenarios using the digital twin to support decision-making. Finally, Section 7 concludes.

2 Conceptual model of the smart factory

The digital twin concept was first devised as a rich digital representation of actual devices, being widely used in the aerospace field [19]. In 2015, the scope of the original definition was expanded, opening the possibility of using it in other fields [20]. Virtual factories were characterized as the digitalization of plants integrated with the

real systems assisting the production process along the lifecycle of each asset [21]. This development led to two different concepts of what a digital twin is: some researchers believe that it can be identified with its virtual component (i.e. the simulation process) while others emphasize on the connections between the virtual and the physical aspects [22]. Tao *et al.* [23] present a design for a smart factory including four main components: a physical shop-floor (PS), a virtual shop-floor (VS), the data generated by the digital twin (D), and a shop-floor service system (SSS).
The physical part includes entities such as human, machines, and materials. The virtual part refers to the digital twin of the physical shop-floor which supports the decision-making and control of the physical part [23]. It provides control orders for the PS and optimization strategies for the SSS. The data component integrates physical, virtual, and service data with the aim of providing consistent information. The service component embodies services like the definition of production and resource allocation plans as well as guidelines for data fusion. These services, in turn, provide information for both the physical and the virtual shop-floor. The authors propose a three stage operation of the digital twin:

- Before production, the SSS defines a production plan based on data collected from different sources (customer orders, real-time data read by sensors, outputs from the simulation, as well as business data) and transmits it to the VS for verification.
- The VS simulates the production plan and if it works correctly, it sends control orders to the PS to start production. The real-time data generated by the PS is then recorded by the VS.
- Once the production process finishes, its recorded history is analysed to extract knowledge that can be used by the VS to simulate future production plans.

In the digital twin shop-floor framework, physical and simulated data are synchronized. The operational implication is that the simulation is fed with real-time sensor data, sending back controls to the PS in a continuous interaction between the physical and virtual components. Tao et al. [23] framework is the basis for implementing the simulation model of a *digital twin cyber-physical system*. We, in turn, add on top of that the process of updating in real-time the production plan in the context of a cloud manufacturing environment. The ensuing digital twin embodies many specialized digital twins: parts DT, machines DT, and a shop-floor DT. The model assumes the use of real-time production data (shop-floor and cloud-shop floor) to support the factory operations.
While simulation has been traditionally used to identify bottlenecks in production plan, it can now be incorporated into smart machines or tools equipped with devices able to capture, process and transmit data. A simulation fed with such information becomes able to check whether the behavior of the physical factory is consistent with the "simulated" behavior. In particular, when (real) data indicates an unforeseen failure, the simulation synchronizes accordingly (data fusion), updating its behavior, and transmitting new control orders to smart devices. Hence, such simulation of the digital twin cyber-physical system helps to handle failures. Figure 1 depicts our simulation model of a digital twin cyber-physical system.

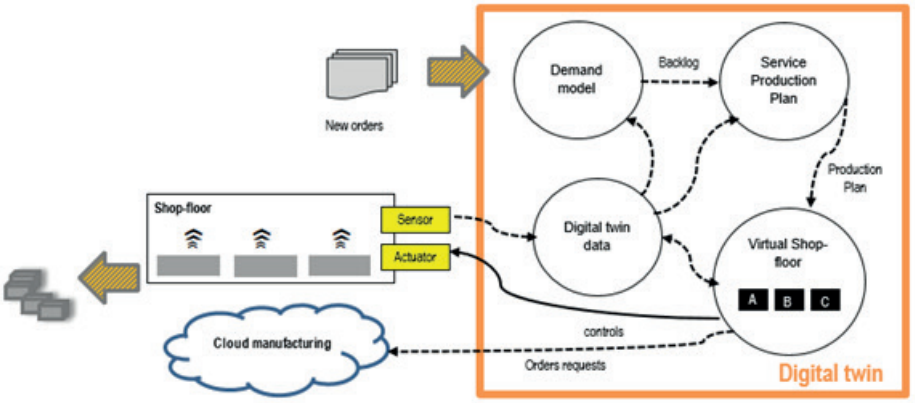

Fig. 1. Simulation model of the digital twin cyber-physical system. Dashed lines: data flow, solid lines: control flow.

3 Running example

Let us consider a smart factory consisting of two production lines A and B. Orders are scheduled on line A. If A is operating above its capacity or it is being repaired, the orders are sent to line B. If line B is operating above its capacity, then the remaining orders are assigned to cloud manufacturing suppliers. For simplicity, we assume that only line A suffers failures and the time to repair is of 3 time units. The cycle times of production lines A, B as well as that of the cloud suppliers are represented by the probability distributions detailed in Table 1. These normal distributions are assumed to have been fitted with historical data of demand.

Table 1. Probability distributions describing cycle time of production facilities.

Production facilities	Cycle time (in days)
A	Normal distribution (mean=2, st. deviation=2%).
B	Normal distribution (mean=4, st. deviation=2%)
Cloud manufacturing suppliers	Normal distribution (mean=2, st. deviation=2%)

4 Causal loop diagrams

The first step towards creating a dynamic model of the digital twin CPS is to draw its Causal Loop diagram [18]. In a Causal Loop diagram variables are connected by arrows denoting the causal influences among them: $x \rightarrow y$ means that the input variable x has some causal influence on the output variable y. A positive influence, denoted by +, means "a change in x, being the rest of variables unchanged, causes y to change in the same direction". In turn, a negative influence means "a change in x, being the rest of variables unchanged, causes y to change in the opposite direction". An increase in a cause variable does not necessarily mean the effect will actually increase

[15]. Since a variable may have more than one input, a change depends on the combined effect of all its input variables. The process wherein one component x initiates changes in other components, and those modifications lead to further changes in x itself is said a feedback loop [24].
According to the system of interest, the diagram may be decomposed in several subdiagrams. For the model of the digital twin cyber-physical system, we developed five diagrams. To save space we include only a partial view of the full causal loop diagram.

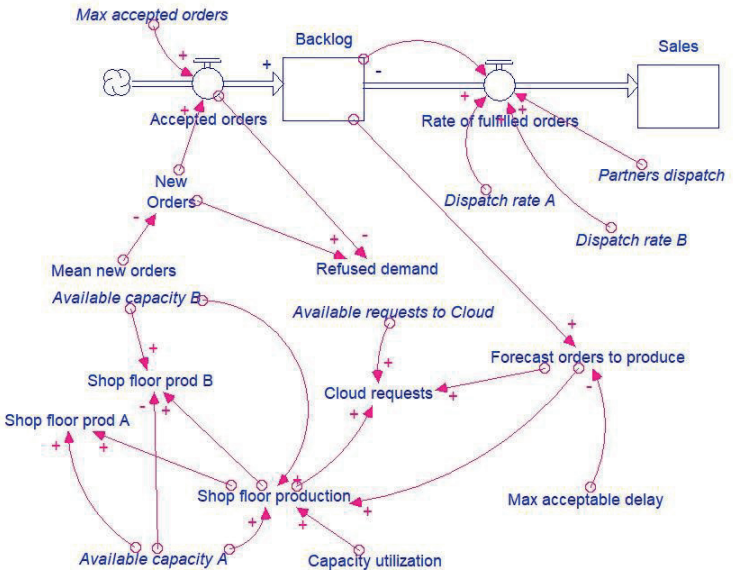

Fig. 2. Causal loop diagram of demand and the production plan. This represents a partial view of the causal loop diagram of the complete system (Generated with Stella™).

Figure 2 depicts the interaction between demand and the production plan. The diagram shows the causal links among the state of the backlog and the accepted orders rate, the dispatch rate of production lines A, B and cloud manufacturing, and the sales volume. As orders increase, the backlog grows, increasing the forecasted number of orders to produce. An increase in the capacity of the production line A adds up the number of units it produces but decreases the shop floor production of line B as well as the production in cloud manufacturing. In turn, increases in failures of production line A decreases its running time and thus its available capacity. As shop-floor production increases, the work in progress and the required dispatch rate grow in time. Finally, increases in the dispatch rate impacts positively on the rate of fulfilled orders and on sales.

The accumulation of resources in a system (stocks), their rates of change (flows) and the feedback loops are the main components of the syntax of dynamic systems models. Stocks are represented by rectangles, inflows and outflows represented by pipes pointing into or out of stocks; valves that control the flows indicate the rates of

change; and clouds represent the sources and sinks for the flows originating outside the boundary of the model. Converters representing processes that convert inputs into outputs are depicted as circles.

5 Stocks and flows diagrams

We structured the smart shop-floor simulation model in five interacting sectors. Cloud manufacturing requests are incorporated in addition to the physical shop-floor production in such a way that production is not limited by plant capacity. The overall structure of interactions between customer demand and flows of orders and products is based on the supply chain models in Sterman [15].

Demand sector. The stock Backlog is the accumulation of accepted orders less the satisfied orders (Fig. 3). It is assumed that orders cannot be changed or cancelled.

$$\text{Backlog}(t) = \text{Backlog}(t-dt) + (\text{Accepted orders} - \text{Rate of fulfilled orders}) * dt \quad (1)$$

There is a cap on the amount of production based on the capacity of the plant and the number of manufacturing orders that can be requested to other manufacturers in the cloud. This cap sets the theoretical maximum production rate that may depend on constraints such as the availability of materials, labor, equipment in the plant and equipment of partners in the cloud. The fulfilled orders rate is given by equation (2).

$$\text{Rate of fulfilled orders} = \text{Dispatch rate A} + \text{Dispatch B} + \text{Partners dispatch} \quad (2)$$

Fig. 3 depicts the Stock and flow diagram for the Demand sector. There are converters and flows defined in such a way as to capture the logic of the synchronization when the behavior of the simulation is not consistent with sensor data. The *Real time gap* is set to 10 meaning that the simulation clock is 10 time units ahead of real time (this value is arbitrarily defined only for illustration). Let the simulation clock be t_s. *Sync backlog* is a variable recording the value of *Backlog* at period $t_s - Real\ time\ gap - 1$. The initial value of converter *Reboot simulation* is 0 and is set to 1 when simulation synchronization is required. Then, if *Reboot simulation* is 1 at simulation time t_s, then the *Backlog* stock is emptied (the flow *Empty Bck* is habilitated) and set to the historical value at $t_s - Real\ time\ gap - 1$.

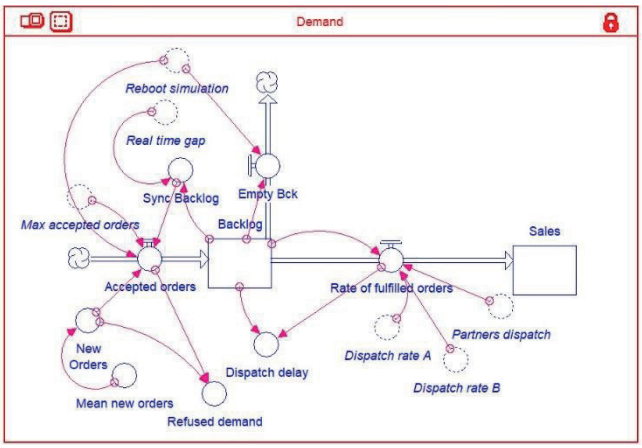

Fig. 3. Stock and flow diagram of Demand (Generated with Stella™).

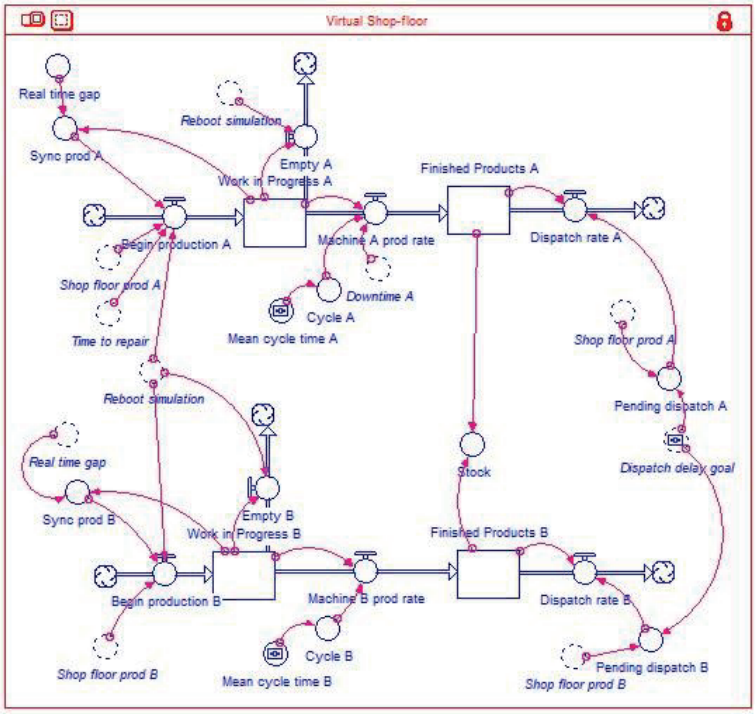

Fig. 4. Stock and flow diagram of the Virtual Shop-floor (Generated with Stella™).

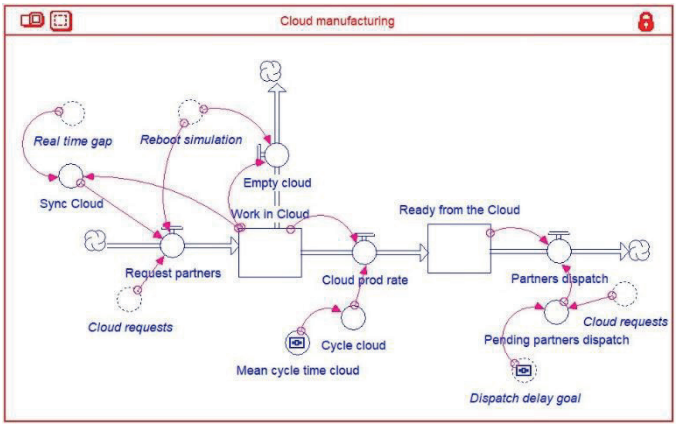

Fig. 5. Stock and flow diagram of Cloud manufacturing (Generated with Stella™).

Virtual shop-floor sector. Production is captured with a chain of two stocks: *Work in progress* and *Finished products* (Fig. 4). For the purpose of this model, all stages of the production process are aggregated together into the *Work in progress* stock as prescribed in Sterman [15].

There are two manufacturing facilities in the plant (A and B) and if necessary, requests are made to cloud manufacturing partners. As mentioned in the description of the Demand sector, there are stocks and flows defined to support synchronization.

Orders are scheduled on production line A. If A is operating above its capacity or it is being repaired, the orders are sent to line B. If also B is operating above its capacity, the required orders are placed to cloud manufacturing providers (Figure 5-8). There exist a maximum of orders that can be requested from the cloud depending on previous agreements with partners. The cycle time is dependent on process-technology and product design (converters *Mean cycle time A*, *Mean cycle time B*). Also, plant A is subject to failure events and the rate of production is thus dependent on the availability of the plant. The stocks, inflows and outflows are formulated as

$$\text{Work in Progress A(t)} = \text{Work in Progress A (t-dt)} + (\text{Begin production A} - \text{Machine A prod rate} - \text{Empty A})*dt \quad (3)$$

$$\text{Empty A (t)} = \begin{Bmatrix} \text{Work in Progress A(t), for Reboot simulation=1} \\ \text{0, for Reboot simulation=0} \end{Bmatrix} \quad (4)$$

$$\text{Machine A prod rate} = \begin{Bmatrix} \text{(Work in Progress A/Cycle A), for Downtime A=0} \\ \text{0, for Downtime A=1} \end{Bmatrix} \quad (5)$$

Most of the time, the inflow *Begin production A* assumes the value *Shop floor production A*. However, when simulation synchronization is required at instant t_s, the simulation should reflect machine A failure at time t_s, and a downtime interval of length $t_s + Time\ to\ repair$, resetting stock *Work in Progress A* with the value at period $t_s - Real\ time\ gap - Time\ to\ repair$ (value recorded by converter *Sync prod A*). Hence, the inflow *Begin production A* is defined as

$$\text{Begin production A(t)} = \begin{Bmatrix} \text{Sync prod A(t-Time to repair), if (Reboot simulation(t)=0} \\ \text{AND Reboot simulation(t-Time to repair)=1)} \\ \text{Shop floor production A(t), otherwise} \end{Bmatrix} \quad (6)$$

We assume that products are dispatched taking into account the dispatch delay goal. For example, if the dispatch delay goal is set to be 3 days and current time is t_s, then finished products which orders were placed at time $t_s - 3$ become dispatched.

$$\text{Finished products A (t)} = \text{Finished products A (t-dt)} + (\text{Machine A prod rate-Dispatch rate A})*dt \quad (7)$$

$$\text{Dispatch rate A} = \text{Min (Pending dispatch A, Finished products A)} \quad (8)$$

$$\text{Pending dispatch A(t)} = \text{Shop floor production A(Max(0, t-Dispatch delay goal))} \quad (9)$$

The dispatch delay takes into account the time since an order is included in the backlog until it is ready for dispatch.

$$\text{Begin production B} = \begin{Bmatrix} \text{0, for Reboot simulation=1} \\ \text{Sync prod B(t-1),} \\ \text{for (Reboot simulation=0 AND Reboot simulation(t-1)=1)} \\ \text{Shop floor production B, otherwise} \end{Bmatrix} \quad (10)$$

Service production plan sector. This sector is in charge of recommending orders of production to each production line (Fig. 6). The number of orders to produce is dependent on the dispatch delay goal and on the available capacity of production lines A, B and of the cloud manufacturing providers. Calculations use data provided by (a) simulations (the available capacity of production lines A and B), and (b) production managers (capacity utilization and the dispatch delay goal).

$$\text{Max acceptable delay} = \text{Dispatch delay goal} \quad (11)$$

$$\text{Forecast orders to produce} = \text{Backlog}/\text{Max acceptable delay} \quad (12)$$

The following formulas represent the scheduling criteria for production lines A, B and cloud manufacturing. The production volume is based on forecasts, assuming that no more than the available capacity is allowed.

$$\text{Shop floor production} = \text{Min ((Available capacity A + Available capacity B)*Capacity utilization), Forescast orders to produce)} \quad (13)$$

$$\text{Shop floor production A} = \text{Min (Available capacity A, Shop floor production)} \quad (14)$$

$$\text{Shop floor production B} = \begin{cases} \text{Min(Available capacity B, Shop floor production-Available capacity A),} \\ \text{for (Shop floor production>Available capacity A)} \\ \text{0, otherwise} \end{cases} \quad (15)$$

$$\text{Cloud requests} = \text{Min(Available requests to cloud,Max (Forecast orders to produce-Shop floor production, 0))} \quad (16)$$

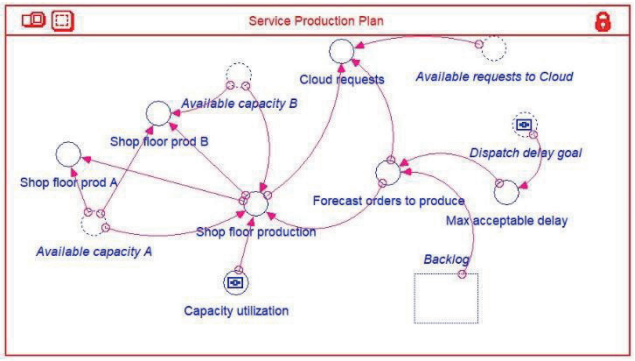

Fig. 6. Stock and flow diagram of Service Production Plan (Generated with Stella™).

Digital twin data sector. The digital twin data sector emulates (Fig. 7) the digital twin data component introduced by Tao [23]. This component is in charge of calculating measures used by the Service Production Plan and the Virtual Shop-floor. These measures are ratios between the capacity utilization of plants A and B, the number of orders that can be requested to cloud partners, and the maximum number of order that can be accepted considering the current level of utilization of plants A and B and cloud services. The dispatch performance rate compares the current dispatch delay and the dispatch goal defined by the firm's goals. The time to repair machine A is defined as constant, but it can be also represented with a probability distribution function or can be a simulation input entered by the operator.

The most relevant part of the digital twin sector is the comparison between data retrieved by sensors and simulated data. For simplicity, we only consider a sensor for machine A (converter *Sensor A*). The sensor detects machine A failures. If this information does not agree with simulated data (converter *Failure A*) at period t_r, then the behavior of the simulation after t_r should be updated to take into account this (the converter *Reboot simulation* is set to 1). In a more realistic setting, the presence of failures should be checked on a time interval (not a point interval).

$$\text{Reboot simulation(t)} = \begin{cases} \text{1, for Sensor A AND (Failure A(t-Real time gap)=0} \\ \text{0, otherwise} \end{cases} \quad (17)$$

The current available capacity of machine A is restricted to its maximum (physical) capacity. In addition, the downtime in case of failure and the reboot of the simulation are also taken into account.

$$\text{Available capacity A} = \begin{cases} 0, \text{ for Downtime A}=0 \text{ AND } \begin{pmatrix} \text{Reboot simulation}=0 \text{ AND} \\ \text{Reboot simulation}(t\text{-Time to repair})=1 \end{pmatrix} \\ \text{Max(Initial capacity A-Work in Progress A, 0), otherwise} \end{cases} \quad (18)$$

$$\text{Available capacity B} = \begin{cases} 0, \text{ for Reboot simulation}=1 \\ \text{Max(Initial capacity B-Work in Progress B, 0), otherwise} \end{cases} \quad (19)$$

6 Scenario modelling

By using the System Dynamics diagrams as a modeling tool and its derived equations we are in conditions of computing numerical solutions. Tools such as Stella™ allow automatizing this computation. There are many techniques of numerical integration of differential equations that can be used to solve the resulting system. The most popular are the Runge-Kutta methods, although their use of finite steps and the approximation to average rates over the interval introduce errors, affecting the results.

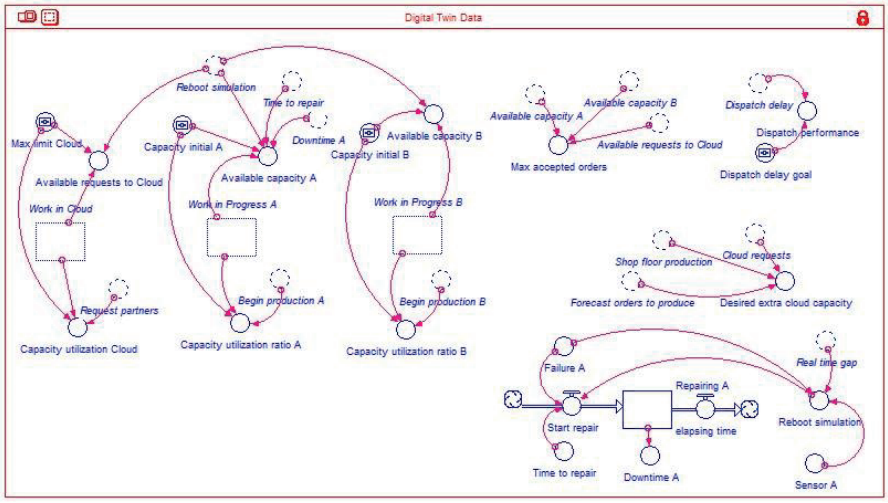

Fig. 7. Stock and flow diagram of Digital twin data. Source (Generated with Stella™).

Table 2. Parameters for the scenario simulation

Parameter description	Value
Demand (number of orders)	Normal distribution (mean=500, st. deviation=10%)
Dispatch delay goal	3 days
Initial capacity of line A	500 units
Initial capacity of line B	400 units
Available cloud capacity	1000 units

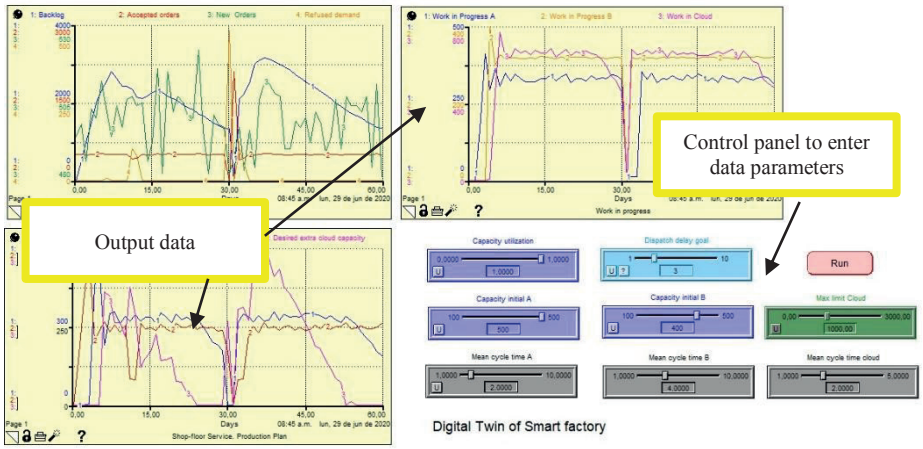

Fig. 8. Digital twin interface including graphics depicting the state of the smart factory and slider input devices to set values of capacity utilization, dispatch delay goal, initial capacity of production lines A and B, maximum limit of requests to cloud manufacturers, and mean cycle times (Generated with Stella™).

Running our system we find that during the first five days, the system is unstable and dominated by the details of the initial conditions. So this period is discarded for analytical purposes. There is a simulated failure of production line A at time 10 and the line is down for 3 days. There is an increase in line B and cloud production (Fig. 9).

There is a failure of line A (detected by sensor data) at real time 20 (the simulation clock is set 10 units ahead of real time). Since the simulation behavior did not consider a failure at time instant 20, then production lines A, B and cloud are set to 0 and re-initialized with real values recorded at simulation time 20 (Fig. 9). In addition, line A is down for 3 days. Simulation forecasts are "corrected" by incorporating real time data when available. Hence, a synchronization of the physical and virtual factory takes place. The available cloud capacity is not enough (notice that *Desired cloud capacity* in Fig. 10 increases after the simulated failure at period 10 and after synchronization due to a real failure of A). This suggests that during downtime plant periods (or an abrupt increase in demand), improving the access to more cloud providers would be recommended. The maximum acquired cloud capacity is not enough to satisfy the desired dispatch delay of 3 days.

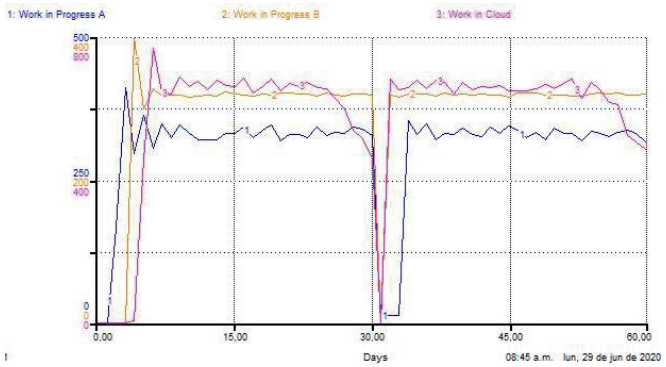

Fig. 9. Production of lines A, B and cloud partners (Generated with Stella™).

In [25] the authors use sensitivity analysis to evaluate the way that behavior patterns vary with different parameters values. We do this to assess the impact of different levels of cloud manufacturing capacity. An increase in that variable would reduce the number of rejected orders. At the same time, such increment entails higher costs incurred in partner agreements.

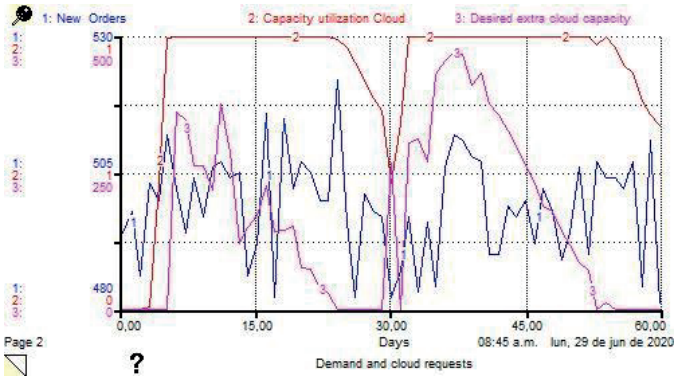

Fig. 10. Demand represented by the *New orders* line; cloud manufacturing capacity utilization (between 0 and 1); and desired extra cloud capacity (defined as the difference between forecasted production orders and available shop-floor and cloud capacity) (Generated with Stella™).

7 Conclusions

The purpose to this paper was to lay down the guidelines for a digital twin of a cyber-physical production system based on a System Dynamics approach. Although we did not intend to cover all systems that may interact with the digital twin, we described how to synchronize the simulation when an inconsistency is detected when

data comes from different sources (data retrieved from sensors and simulated data). It is possible to extend the model to integrate the digital twin with different databases of the organization.

Due to the anticipatory nature of simulation runs, different production plans can be examined to choose the most convenient one according to criteria like optimizing cost, dispatch rate or rejected orders. This information may be useful to support human decision-making or to control a physical machine. This enhances the value of a digital twin (simulation) by supporting real-time production control.

Other simulation paradigms can be used to model and implement a digital twin. For example, discrete event simulation [26] is adequate to model the shop-floor production lines. However, a digital twin of a cyber-physical production system is a complex system (many components or subsystems interact and the knowledge of the impact of a change in a variable over another is incomplete). To model problems with such levels of uncertainty an iterative learning process and modeling tools able to deal with incomplete information are required [18]. The model presented in this work serves as a basis to add interactions with other subsystems belonging to different tiers of the supply chain.

The system dynamics paradigm is adequate to model uncertainty and study the evolution of the system over time (for example, the interactions among demand, logistic disruptions, and machine fatigue). System dynamics shows its potential in forecasting or understanding the far future. Hence, a promising approach is to use an hybrid method based on discrete event simulation to handle short term phenomena (machine failure) and system dynamics comprising all subsystems and assess the far future.

References

1. K. Ding, F. Chan, X. Zhang, G. Zhou and F. Zhang, "Defining a Digital Twin-based Cyber-physical Production system for autonomous manufacturing in smart shop floors," *International Journal of Production Research,* pp. 6315-6334, 2019.
2. X. Yao, J. Zhou, Y. Lin, H. Yu and Y. Liu, "Smart manufacturing based on cyber-physical systems and beyond," *Journal of Intelligent Manufacturing,* pp. 1-13, 2017.
3. E. Lee, "Cyber physical systems: Design challenges," in *11th IEEE International Symposium Object Oriented Real-time Distributed Computing*, Florida, 2008.
4. L. Wang and X. Wang, Cloud-Based Cyber-Physical Systems in Manufacturing, London: Springer, 2018.
5. J. Li, F. Tao, Y. Cheng and L. Zhao, "Big data in product lifecycle management," *The International Journal of Advanced Manufacturing Technology,* vol. 81, no. 4, pp. 667-684, 2015.
6. F. Tao, Q. Qi, A. Liu and A. Kusiak, "Data-driven smart manufacturing," *Journal of Manufacturing Systems,* vol. 48, pp. 157-169, 2018.
7. L. Monostori, "Cyber-physical production systems: Roots, expectations and R&D challenges," *Procedia Cirp,* vol. 17, pp. 9-13, 2014.
8. D. Rossit and F. Tohmé, "Scheduling research contributions to Smart manufactur-

ing," *Manufacturing Letters,* vol. 15, pp. 111-114, 2018.

9. D. Rossit, F. Tohmé and M. Frutos, "Industry 4.0: Smart Scheduling," *International Journal of Production Research,* pp. 1-12, 2018.
10. M. Parsanejad and H. Matsukawa, "Work-in-process analysis in a production system using a control engineering approach," *Journal of Japan Industrial Management Association,* vol. 67, no. 2, pp. 106-113, 2016.
11. D. Rossit, F. Tohmé and M. Frutos, "An Industry 4.0 approach to assembly line resequencing," *The International Journal of Advanced Manufacturing Technology,* pp. 1-12, 2019.
12. V. Damjanovic-Behrendt and W. Behrendt, "An open source approach to the design and implementation of Digital Twins for Smart Manufacturing," *International Journal of Computer Integrated Manufacturing,* pp. 1-19, 2019.
13. Y. Barlas, "Formal aspects of model validity and validation in system dynamics," *System Dynamics Review,* vol. 12, no. 3, pp. 183-210, 1996.
14. J. Morecroft, Strategic Modelling and Business Dynamics. A feedback systems approach, Chichester: John Wiley & Sons Inc., 2007.
15. J. Sterman, Business Dynamics: Systems Thinking and Modeling for a Complex World, New York: McGraw-Hill, 2000.
16. J. Forrester, Industrial Dynamics, Massachutses: Pegasus Communications, 1961.
17. P. Senge, The fifth discipline: the art and practice of the learning organization, New York: Doubleday/Curency, 1990.
18. M. Sánchez, "Modeling for System´s Understanding," in *Formal Languages for Computer Simulation: Transdisciplinary Models and Applications*, P. Fonseca i Casas, Ed., Hershey, IGI Global, 2013, pp. 38-61.
19. M. Grieves, *Digital twin: Manufacturing excellence through virtual factory replication,* 2014, pp. 1-7.
20. J. Rios, M. Oliva and F. Mas, "Product Avatar as Digital Counterpart of a Physical Individual Product: Literature REview and Implications in an Aircraft," in *22nd ISPE Inc. International Conference on Concurrent Engineering*, Delft, 2015.
21. M. Sacco, P. Pedrazzoli and W. Terkaj, "VFF: Virtual Factory Framework," in *2010 IEEE International Technology Management Conference*, Lugano, 2010.
22. F. Tao, H. Zhang, A. Liu and A. Nee, "Digital Twin in Industry: State-of-the-Art," *IEEE Transactions on Industrial Informatics,* vol. 15, no. 4, pp. 2405-2415, 2019.
23. F. Tao and M. Zhang, "Digital Twin Shop-Floor: A New Shop-Floor Paradigm Towards Smart Manufacturing," *IEEE Access,* vol. 5, pp. 20418-20427, 2017.
24. B. McGarvey and B. Hannon, Modeling Dynamic Systems, Springer, 2004.
25. M. Hekimoglu and Y. Barlas, "Sensitivity Analysis of System Dynamics Models by behavior Pattern Measures," in *Proceedings of the 28th International Conference of the System Dynamics Society*, Seul, 2010.
26. J. Banks, J. Carson, B. Nelson and D. Nicol, Discrete-Event System Simulation, Upper Saddle River, New Jersey: Prentice Hall, 2001.