Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

The role of dynamics in digital twins and its problem-tailored representation

Chiara Cimino * Gianni Ferretti * Alberto Leva *

* *DEIB, Politecnico di Milano, Italy*
(e-mail: { chiara.cimino,alberto.leva,gianni.ferretti}@polimi.it)

Abstract In Smart Manufacturing, the Internet of Things technology brings about new possibilities as for the interaction of Cyber-Physical Systems with their virtual models—in Industry 4.0 jargon, Digital Twins. Besides traditional roles in engineering and control design, Digital Twins cab play innovative ones by evolving together with their cyber-physical counterpart: predictive maintenance, fault detection, or fast training of data-based decision aid tools. In this paper we focus on the relevance of dynamic models in this *scenario*, particularly as for the co-existence of continuous-time and discrete-event models wherever control and planning are involved. We argue that in such cases, models with analysis-specific levels of detail have to co-exist, interact and maintain mutual consistency. We propose a modelling approach to address the problem, and present a supporting example based on an object-oriented modelling language.

Keywords: Dynamic models; Advanced manufacturing; Production control and management; Cyber-physical systems; Object-oriented modelling and simulation.

1. INTRODUCTION AND MOTIVATION

This paper is about using Dynamic Models (DMs) in a Smart Manufacturing (SM) context, often referred to as Industry 4.0 (I4.0). We focus on the evolution that this context induces in the roles played by DMs – on which Digital Twins (DTs) are based – for the engineering of Cyber Physical Systems (CPSs). DMs serve various purposes, from design to control up to asset management. Many such uses are nowadays enabled by the connectivity boost yielded by the so called Internet of Things (IoT) technologies, not only within a plant but also involving the associated industrial processes.

The density of acronyms in the paragraph above is deliberate. As technology progresses, engineering domains once disjoint come to interact. Besides new opportunities, this brings about possible new inefficiencies due to poorly harmonised viewpoints on the same entities – and a symptom is the proliferation of names.

Some years ago, the role of DMs was not so pervasive, nor so relevant was the idea of transferring models from one phase to another through the lifetime of an asset. Different models – not all dynamic – took part in the engineering of a plant and its control, but their task more or less ended when the plant was commissioned, or control set up, or in general, at the end of the design and management phase that originated them. This was quite natural in a world where the tools used to make models and controls, as well as the instrumentation aboard the plant, were so heterogeneous – or equivalently, designed in so unrelated ways – to make their interconnection not natural.

Today this is not the case anymore, and as said, new possibilities come with new problems. One of particular interest for us, is how to exploit the enormous amount of data that modern plant instrumentation generates, *in conjunction* with the knowledge stored in the DMs used for its engineering. Said just a bit more brutally, taking decisions through data analytics is fine, but disregarding DMs for the same purpose potentially is not. Just as an example, if a DM based on physical principles is driven to off-design operation, its outcome can still be trusted as long as its validity limits, stated in physical terms as well, are obeyed; guaranteeing the same properties in a *purely* data-based context is not equally straightforward.

As will emerge from the brief literature review of Section 2, there is much on how to draw DTs from data, but much less on the role played in DTs by DMs, and even less about the integration – whatever is meant for this – of DMs and data. The above said, the contributions of this paper can be summarised as follows.

- A literature review on DTs in the SM context, showing the advantages of introducing DMs into DTs.
- A study on structuring DTs for use both in the absence and the presence of plant data, i.e., both for asset design/engineering and control/management.
- General clues to integrate DMs into DTs, specifically using Discrete Event Systems (DEVS) as computationally efficient replicas – though not totally physical – of Continuous-Time (CT) ones.
- A discussion on how the resulting modelling approach lends itself to the various situations seen in manufacturing systems, and on the opportunity of representing those systems with a mixture of CT and DEVS models.
- A realisation proposal for the said approach, formally based on a control description language (SFC) and operationally implemented by means of an Object-Oriented Modelling and Simulation (OOMS) multiphysics one (Modelica).

2. BRIEF LITERATURE REVIEW

A lot has been said in the literature about Smart Manufacturing and its enabling technologies, in particular CPSs. The CPS concept is defined by Monostori (2015) as "system of collaborating computational entities which are in intensive connection with the surrounding physical world and its on-going processes, providing and using, at the same time, data-accessing and data-processing services available on the internet".

In more technological terms, the definition could read "a network of computers, (some) instrumented to communicate with a physical process, and connected to the Internet". This implies that data processing capabilities are inherent to such systems, but says nothing about *what* data elaboration is envisaged. The CPS concept can be applied at various granularities, from a single component to an entire machine or "Cyber Physical Production System (CPPS)" (Weyer et al., 2015). Also, the CPS idea in automation adds the implication that the Physical (P) part should be the plant, in broad sense, and the Cyber (C) one should be (or at least contain) its control.

The idea of DT is very similar, but focuses on *replicating* the P into the C, thus becoming the part of a CPS devoted to that specific purpose. However, notwithstanding the large amount of literature on DTs, there is to date no unique definition; in general a DT is just associated to a "digital copy" of the system somehow created by the CPS (Negri et al., 2017; Tao et al., 2018). As the more technological and control-oriented counterpart, here too a DT uses the data collected by the CPS to simulate a system in a digital environment.

The literature on DTs is vast. We here focus on works about how a DT is – or should be – employed, typically classified by use. In manufacturing, a DT is mostly associated to system analysis, can be used for scheduling purposes (Zhang et al., 2019), or for fault prediction and maintenance (Kritzinger et al., 2018; Lu et al., 2020; Tao et al., 2018; Macchi et al., 2018). Also, a DT should improve performances through the whole plant lifecycle and with a different level of integration with the real system (Kritzinger et al., 2018). Each phase of that lifecycle could correspond to a set of analyses, compliant with some standards (Lu et al., 2020). Each analysis, in turn, could be executed using a collection of models that represent the same system. Overall, a DT should thus be able to perform different types of simulation along the whole lifecycle of a manufacturing system.

Focusing on an SM *scenario* made of interconnected CPSs, a DT is generally used for analyses that can be done in parallel with the system or not: a DT can be used offline (not necessarily while the real system is running) and (as the opposite case) online (Borangiu et al., 2019).

- Offline - given a model of a real system, the offline mode produces a complete simulation with the possibility of defining – depending on the case – some Key Performance Indicators (KPI).
- Online - the same KPIs of the offline mode are computed in real time. The DT uses these and/or the simulation results to understand how much the system is closer to its digital counterpart.

Talking about offline and online mode operation, this paper will refer to the above definitions, applied in different phases of the lifecycle. The literature illustrates the uses of a DT for design, production, and maintenance through purpose-specific models (Negri et al., 2017). Also, a novelty that can enhance the production efficiency and productivity, would be the introduction of a control action directly from a DT based on operation conditions (Tao et al., 2018).

There is to date no unique architecture for DTs to support all these features. There exist different purpose- or problem-tailored ways to build a DT: Cimino et al. (2019) outlines this through a systematic review on DT applications, highlighting that most DTs are built based on data, hence replicate the system dynamics as long as those data are representative. Such DTs have knowledge about the history of a system but, without the dynamic provided by physical DMs, they can deal only with the already occurred behaviours. To include control in the possible online roles of the DT – neglecting the difficulties that must be considered when talking about CPS based control, not discussed in this paper – the classical control theory has to be recalled. A model of the considered plant is needed for real time control. As shown in Figure 1, an example of DT for a feedback control system consists in a model of the plant that is fed, in parallel with the real plant, with the output of the controller.

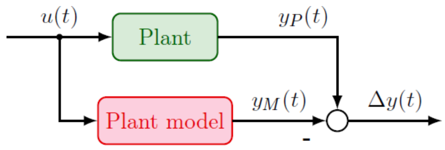

Figure 1. DT control with the use of the DM of a plant

DMs are also useful for fault prediction and maintenance. To this end, the same plant and DM can be used both offline and online. A DT can also be used for fault prediction and maintenance (online).

- Offline – through simulation-based computation of KPIs, related to maintenance and fault prediction.
- Online – The same simulation and/or KPIs of the offline mode can train the system to recognise the faults online. Once the system is installed, this should reduce the training time as exemplified in Figure 2 (Zhang et al., 2019).

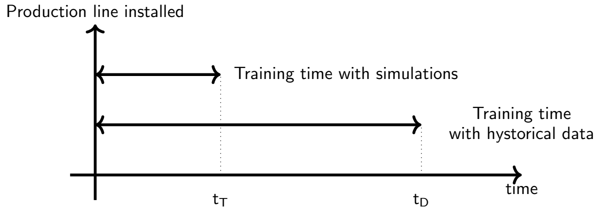

Figure 2. Training time reduction with DM simulations

In synthesis, in the literature DTs are widely mentioned in manufacturing for different purposes. Although these systems – at each level of detail – are already represented by different models, these are mostly based on data (as opposite to DMs). In this respect, the introduction of DMs in a DT could improve their use in the whole lyfecycle. DMs give the possibility of controlling the system from a DT, and can also enhance predictive maintenance and fault detection. Control through a DT with DMs is well suited for application at different levels of details and to different models, corresponding to the representation of a system in its different characterisations. The difficulty lies in managing DM-endowed simulations of each problem-tailored representation.

3. MANUFACTURING PLANT SIMULATION

We now focus on the manufacturing context. According to literature and applications, here a DT has neither a single representation nor a single role. On the contrary, it has different definitions applicable to the distinct roles that it can assume. It is highly desirable that a DT replicates the system from the beginning of its life cycle and gives information about it at several levels of detail. Then, to obtain different analyses, there is the need to represent each system throughout all its levels of detail.

The realisation of such an ambitious task involves several aspects. The processes in manufacturing are various and can be divided into simpler subsystems, each with different representations, more or less complicated in reference to the required level of detail (e.g. the production line, the single machine, the process, the tools). Moreover, this concept is also related to the dynamics to be described, including the relevant controls.

A case study on the creation of a digital copy of a real system, is described in (Cannon et al., 1991; Turvesky et al., 2007b,a) referring to a digital replica of an airplane from the beginning, starting from the geometrical properties. Design choices define the aerodynamics of the vehicle, then the various parts of the project are interconnected with one another and the simulation of each single dynamic of the components brings to the complete design of the airplane and to the final flight simulation. Then the process continues to the code building, the flight testing and the analysis of the flight data. The results obtained from the analysed case study are good tools to simulate the system allowing for an estimate of the performances and handling qualities well before any hardware is built, reducing design costs and preventing errors. This approach is also promising in the case of manufacturing systems. However it has to be well noticed that notwithstanding the success of model-based engineering in the avionic domain and similar ones, attempting to replicate the story in the manufacturing context is a hard task: systems are far more heterogeneous and complex, there is far less standardisation, production processes are continuously modified, re-designed, joined and split. In one word, after the initial design, engineering and operation continue to co-exist on a daily basis.

Considering the available space, to exemplify, we study the use case of a laboratory apparatus composed by an heater and a mechanical stirrer – that could well be part of a production line – and attempt (in small scale) to replicate the same approach of the avionics case sketched above. Systems like this are mostly represented as DEVS, namely with a model based on the events occurring during the production (Mourtzis et al., 2014; Jahangirian et al., 2010). As a possible part of a production line, the considered process can be represented as a DEVS like in Figure 3, in sequence: a generator of events that randomly decides when an entity – in this case each piece to be heated – entered the system, a buffer that accumulates a queue of pieces, a model that simulates the service time – in this case the process time – and an event terminator to visualise the heated pieces leaving the system.

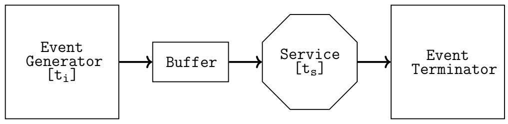

Figure 3. Generical DEVS representation

When dealing with DEVSs, event instants are generally not deterministic but based on statistical distributions, therefore DEVS models are in general stochastic. In the example above, as common in DEVS models, both the generator of the events and the time process are decided by probabilistic distributions. In practice, however, only the events that decide the part inter-arrival time – t_i in Figure 3 – is properly stochastic, while the processing (service) time t_s would be better described by the physics of the system through a DM, confining stochasticity to the physical entities and quantities that are really subject to it. The only way to connect the probabilistic distribution of the process time with the controlled system is to consider all the possible parameters of the system that can produce some variability to the process time and compute multiple simulations of the process according to that variability. Also, the DEVS system – as represented above – is poorly scalable in the level of detail. In particular, when expressing the process plant with a unique DM of the system the relation between parameters and level of detail is not so clear.

To visualise the concept, considering the laboratory apparatus use case, the DM of the plant would be represented by the transfer function 1 that relates the difference between the liquid temperature and the air temperature (T_L) to the heating power (P_h). The variation of the system is given by the parameters a, t_1, t_2.

$$P(s) = \frac{T_L}{P_h} = \frac{a}{(1 + t_1 s)(1 + t_2 s)} \tag{1}$$

In the system, t_1, t_2 are the two time constants of the process, that in turn relates to the equations of the physical process. In the mentioned example, they can derive from the variability of a heat exchange coefficient or from the variability of the contact area during the heating process. Such correlation represents the scalability in the level of details. In short:

- In general, in a DEVS representation, it is possible to decide the level of detail only changing the probability distribution related to the variability of a set of system parameters;
- The DMs provides the description of the physical processes, that relates the parameter variability of the system with the probability distribution (from simulation results).

What is missing is the definition of a standard architecture that associate the correct DM to each level of detail of the system, being manufacturing systems really various, and with deep scalability. As a consequence, with the introduction of DMs in DEVS for manufacturing environment modelling, it is possible to:

- Give a physical meaning to the probability distributions coming from DEVSs;
- Understand which are the right parameters to include in the computation of the said probability distributions (among the large set of parameters that can be found in a manufacturing system).

A consistent representation can be identified in an existent structure, the Sequential Functioning Charts (SFC); the choice will be better motivated in Section 4.

4. SFC MODEL BASED WITH OO LANGUAGE

Until now we talked about DTs in CPS-based manufacturing. The idea of introducing DMs into a DT, could enlarge its role from control to fault detection and predictive maintenance. Though the majority of manufacturing systems are simulated with DEVSs alone, a connection between DEVS and DMs could help confine stochasticty in DEVS simulation and related it to the physics of the system. As said in Section 3, such a link requires a formalism to provide (i) scalability in the level of detail, (ii) integration with the control strategy, and (iii) representing the dynamics of the system.

Thinking e.g. of the laboratory apparatus example of Section 3, the control logic is well described by models based on the IEC 61131(Controllers-Part, 2013) standard on expressing the logic control systems defining five programming languages (Otto and Hellmann, 2009). Among them, the language that follows this standard and that gives also a good representation of the system for operations, control design and simulation is the well known SFC, introduced under the french name of GRAphe Fonctionnel de Commande Etapes/Transitions (GRAFCET). A lot can be found in the literature about the description of systems using this representation.SFC is mainly based on two concepts, represented in Figure 4:

- The phase, where certain actions are performed;
- The transition, that allows to change the set of active phases when an event occurs.

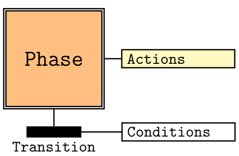

Figure 4. SFC basic elements.

The SFC for the laboratory apparatus example in Section 3 is shown in Figure 5, referring to a heated stirrer. In the initial state **1**, the system is initialised by setting no control action (AUTO=0) and the stirrer OFF (CMDOFF). State **2** becomes active when the stirrer is actually OFF. When the START button is pressed, state **3** becomes activate until the sensor detects the presence of liquid (Li). When the system reaches state **4**, the control action is activated (AUTO=1) and the liquid temperature set point is fixed

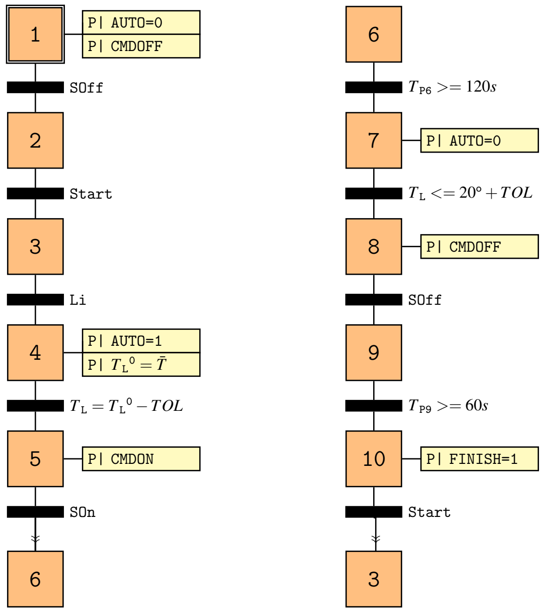

Figure 5. Example: SFC of a laboratory apparatus

($T_{L0} = \bar{T}$). State **5** fires when the measured liquid temperature T_L becomes equal to the desired temperature within a tolerance, and the stirrer is switched ON (CMDON). Once the stirrer is ON (Son) the system reaches state **6** and remains there for 2 minutes. After that, the system arrives in state **7** where the control action is deactivated (AUTO=0), and when the temperature goes below 20°C, the system goes in state **8** and the stirrer is turned OFF (CMDOFF). When the stirrer is OFF (SOff) the system goes in state **9**, stays there for 1 minute and reaches the final state **10**, send the FINISH signal and waits for another START signal to start over from state **3**. The simulation results of the described example is illustrated in Figure 6, from bottom to top: the first plot exhibits the evolution of the system through its phases, the second and the third ones are respectively the behaviours of the control action (P_h) and the controlled variable (T_L).

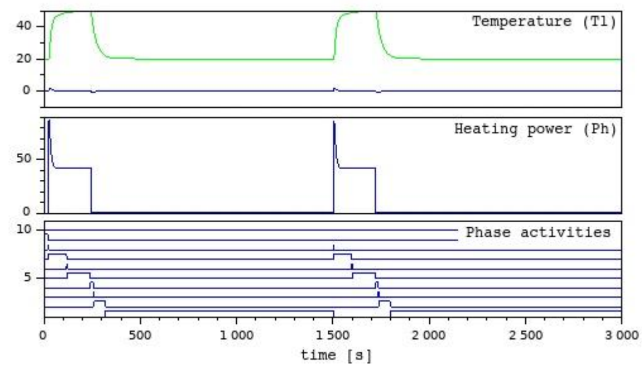

Figure 6. Example simulation with fixed parameters: $a = 0.7$, $t_1 = 40$, $t_2 = 2$

As said in Section 2, in detail, the time constants of this specific operation depend on a heat exchange coefficient, whose variations around a nominal value impact the total heating time. To prove it, the represented system was simulated 1000 times: in each simulation the heat coefficient was varied based on a Gaussian random generator. This yielded a variation of the time constant t_1 in a range $[-10, 10]$ around the nominal. The resulting heating time distribution is depicted in Figure 7.

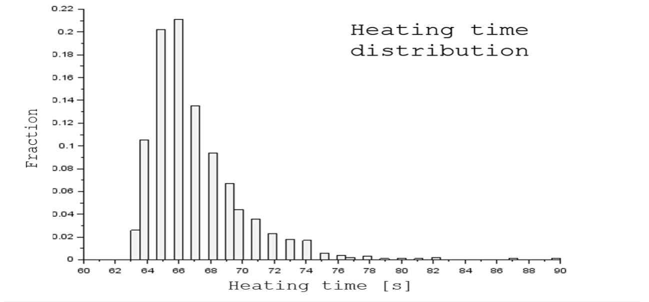

Figure 7. Heating time distribution varying the heat coefficient

Figure 7 suggests that, in general, the process times do depend on their DMs, that relies on the level of detail of its representation given by a set of parameters that may vary in the system. Also, a variation given by a Gaussian number generator does not produce a Gaussian distribution, and this is related to both the variability of physics and to the control action.

This representation permits the co-existence of continuous models with the event-based ones and can be easily extended or reduced in order to describe less or more levels of detail of the system. Above, we talked about the level of detail in reference to the set of parameters used in a DM of a process, but the level of detail can be also referred to the processes of a system that are described by a DM in a SFC. Taking again the example into consideration, if the stirrer is automated, it could be described by its control logic and then can be included in the PLC. As illustrated in Figure 8, the SFC could be extended adding a state **SD-On** where to set the automatic mode of the stirrer ($AUTO_{stirrer} = 1$) and a set point for the stirrer speed ($v_S = \bar{v}$). Consequently, a command is added to deactivate the control action of the stirrer in state **8** and then the stirrer is turned OFF (CMDOFF). In this case, not only the time constant of the on/off process of the stirrer is considered – as in the example in Figure 5 – but also its controlled dynamic, knowing the related DM of the stirrer (that relates the speed to a control variable).

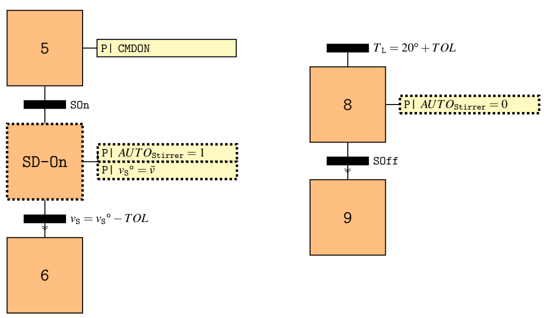

Figure 8. Changes of the SFC of the laboratory apparatus example to include the stirrer dynamic

In synthesis, a system can be identified by a series of processes, each one can be considered or not and described by its control logic, based on a DM. A first level of detail is given by the presence of a DM for each process. Each DM in turn can provide different levels of detail of each process, that correspond to a different description of it and then to several sets of parameters that can bring variability to the process. To implement such a structure in a way that can be created in future directly from a DM based system, the language selected is an Object-Oriented (OO) one. The simulation environments that use these languages can handle multi-physic systems – needed to model the larger possible range of systems in a manufacturing environment – and can relate event-based systems with continuous time ones (Mattsson et al., 1998; Fritzson, 2010).

Once chosen the environment and the language to implement the identified structure, the SFC correlates DEVSs and DMs maintaining three main features:

(1) *Scalability in the levels of detail.* As seen from the example in Figure 5 and 8, a SFC representation can be made scalable paying attention to the object abstracted in the OO language;
(2) *Integration with the control strategy.* The SFC standard by definition represents the control logic. Plant models in OO language can be interconnected with their controls, and in doing so the SFC logic control representation can be described as an input/output systems, considering (i) the outputs correspond to the actions to be performed in a specific phase, and (ii) the inputs correspond to the conditions that trigger the transition and consequently the switching from one active phase to another.
(3) *Dependency on DMs.* SFC gives the possibility to connect to the input/output representation different DMs, using different dynamic representations depending on the required level of detail.

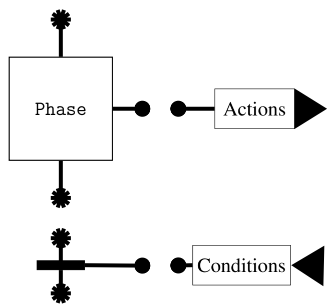

Figure 9. Possible SFC blocks in OO language

The implementation process is still in progress. A possible result is envisaged in Figure 9 as the OO language objects represent the SFC. The arrows in the “transition conditions” and in the “action to perform” are respectively inputs and outputs of the SFC representation. Notice that the Input/Output model created coincides with the PLC standards and is convenient to test the control strategies in simulation (Otto and Hellmann, 2009). The library under development is compatible with the SFC structure and the modelling purposes, and corresponds to the DT concept in a CPS-based SM environment.

5. CONCLUSION AND FUTURE WORK

A DT can assume different roles in a CPS-based SM *scenario*, from design to control and maintenance. There is however still some confusion on its operating modes, and on the models to represent a manufacturing environment accounting for its multiple products and the various required detail levels. Also, according to the literature on DTs, most of the used models are based only on data.

In this paper we considered the introduction of DMs in a DT and its possible integration with other models, in turn dependent on the dynamic of the systems, and pointed out the main reasons for the use of DMs. The resulting DT should be capable of integrating different simulation analyses and detail levels in the same digital environment, and exchange information among those analyses. This structure was discussed in association with the possible operating modes of a DT. The discussed environment where all the simulations – based also on different models – can interchange information among each others, is placed in the offline mode and can lead to other consequent offline operations, i.e. the computation of KPI or the training of a system (Figure 10). Then the offline mode can support numerous online operation, i.e. system or subsystem control, maintenance strategies, fault detection or different real-time comparisons with simulations (Figure 10).

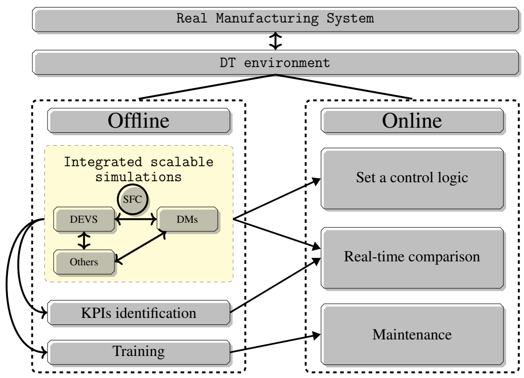

Figure 10. DT uses in a Manufacturing environment

As a result of the above considerations, and also of the reported minimal analysis about the role of stochasticity and poor scalability in the level of detail that one experiences when representing the dynamics of a system through non problem-tailored DMs and thus DTs, we devised a modelling approach based on the use of SFC models, and targeting an OOMS language. The SFC-OOMS structure implementation is still in progress – we reported a preliminary example – but it is a good starting point for a research path to create the environment discussed above. Future work will be focused in the implementation of a library that can describe the SFC in OO language. Then, the research will be dedicated more on obtaining the SFC representation directly from the correspondent DM for the required level of detail.

REFERENCES

Borangiu, T., Oltean, E., Răileanu, S., Anton, F., Anton, S., and Iacob, I. (2019). Embedded digital twin for arti-type control of semi-continuous production processes. In *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*, 113–133. Springer.

Cannon, M., Gabbard, M., Meyer, T., Morrison, S., Skocik, M., and Woods, D. (1991). Swineworks d-200 sky hogg design proposal. *AIAA®/General Dynamics Corporation Team Aircraft Design Competition*, 1992.

Cimino, C., Negri, E., and Fumagalli, L. (2019). Review of digital twin applications in manufacturing. *Computers in Industry*, 113, 103130.

Controllers-Part, P. (2013). 3: Programming languages, IEC 61131-3: 2013. *International Electrotechnical Commission.*

Fritzson, P. (2010). *Principles of object-oriented modeling and simulation with Modelica 2.1.* John Wiley & Sons.

Jahangirian, M., Eldabi, T., Naseer, A., Stergioulas, L.K., and Young, T. (2010). Simulation in manufacturing and business: A review. *European Journal of Operational Research*, 203(1), 1–13.

Kritzinger, W., Karner, M., Traar, G., Henjes, J., and Sihn, W. (2018). Digital twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine*, 51(11), 1016–1022.

Lu, Y., Liu, C., Kevin, I., Wang, K., Huang, H., and Xu, X. (2020). Digital twin-driven smart manufacturing: Connotation, reference model, applications and research issues. *Robotics and Computer-Integrated Manufacturing*, 61, 101837.

Macchi, M., Roda, I., Negri, E., and Fumagalli, L. (2018). Exploring the role of digital twin for asset lifecycle management. *IFAC-PapersOnLine*, 51(11), 790–795.

Mattsson, S.E., Elmqvist, H., and Otter, M. (1998). Physical system modeling with Modelica. *Control Engineering Practice*, 6(4), 501–510.

Monostori, L. (2015). Cyber-physical production systems: Roots from manufacturing science and technology. *AT-Automatisierungstechnik*, 63(10), 766–776.

Mourtzis, D., Doukas, M., and Bernidaki, D. (2014). Simulation in manufacturing: Review and challenges. *Procedia CIRP*, 25, 213–229.

Negri, E., Fumagalli, L., and Macchi, M. (2017). A review of the roles of digital twin in cps-based production systems. *Procedia Manufacturing*, 11, 939–948.

Otto, A. and Hellmann, K. (2009). IEC 61131: A general overview and emerging trends. *IEEE Industrial Electronics Magazine*, 3(4), 27–31.

Tao, F., Zhang, H., Liu, A., and Nee, A.Y. (2018). Digital twin in industry: state-of-the-art. *IEEE Transactions on Industrial Informatics*, 15(4), 2405–2415.

Turvesky, A., Gage, S., and Buhr, C. (2007a). Accelerating flight vehicle design. *MATLAB® Digest, January.*

Turvesky, A., Gage, S., and Buhr, C. (2007b). Model-based design of a new light-weight aircraft. In *AIAA Modeling and Simulation Technologies Conference and Exhibit*, 6371.

Weyer, S., Schmitt, M., Ohmer, M., and Gorecky, D. (2015). Towards industry 4.0-standardization as the crucial challenge for highly modular, multi-vendor production systems. *Ifac-Papersonline*, 48(3), 579–584.

Zhang, J., Ding, G., Zou, Y., Qin, S., and Fu, J. (2019). Review of job shop scheduling research and its new perspectives under industry 4.0. *Journal of Intelligent Manufacturing*, 30(4), 1809–1830.