

An integrated simulation paradigm for lifecycle-covering maintenance in the Industry 4.0 context

Chiara Cimino* Alberto Leva** Elisa Negri***
Marco Macchi***

* *PhD student at the DEIB, Politecnico di Milano, Italy*
(e-mail: chiara.cimino@polimi.it)

** *DEIB, Politecnico di Milano, Italy*
(e-mail: alberto.leva@polimi.it)

*** *DIG, Politecnico di Milano, Italy*
(e-mail: {[marco.macchi](mailto:marco.macchi@polimi.it),[elisa.negri](mailto:elisa.negri@polimi.it)}@polimi.it)

Abstract The Industry 4.0 is pushing a fast evolution in all manufacturing operations under heterogeneous aspects. Maintenance, in all its numerous declinations along the life cycle of a production asset, is no exception. A myriad of decisions need taking—often on short time horizons, involving different physical and technological domains, and with limited knowledge available. A myriad of data sources need exploiting and interpreting—often poorly structured, requiring reconciliation, and affected by uncertainty. A lot of tools are therefore used—each one conceived for a specific purpose, often leveraging knowledge drawn from different cultures, and dealing with production assets' models that are hard to integrate with one another. We conjecture that the OOMS (Object-Oriented Modelling and Simulation) paradigm can help address such a tough *scenario*. In this paper we structure and motivate the statement just made, provide high-level supporting examples, and outline future work as for both research and technology.

Copyright © 2020 The Authors. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0>)

Keywords: Modelling and simulation for maintenance; Advanced manufacturing; Cyber-Physical Systems; Dynamic models; First-principle models; Object-oriented modelling and simulation.

1. INTRODUCTION AND MOTIVATION

In modern manufacturing systems, the need for deep and fast reconfigurability is steadily increasing (Bortolini et al., 2018; Napoleone et al., 2018). At design time, it is nowadays impossible to imagine (thus to consider and optimise) all the working conditions that a production asset will be confronted with. Maintenance policies are heavily affected by this trend, not only because they need fast adaptability as well, but also and more important, because they may have to cope with operating *scenarii* that were not considered at all when the system (individual asset or system of assets) was initially built. New and tough challenges need therefore facing, along the major axes outlined below.

Time scale. In the past, the lifecycle management of a plant and its maintenance used to operate on significantly different time scales. Decisions concerning design, commissioning and possibly refurbishments could be planned in advance, and once applied, there was normally enough time to refine them with minor interventions before the subsequent heavy reconfiguration. On the other side, the operational life of the plant included decisions on maintenance interventions that were made in a limited scope of work and on a short time horizon axis. At present, the two time scales above are becoming closer and closer; reconfig-

urations can be nearly as frequent as the day-by-day (or event-generated) decisions to take about maintenance.

Data availability. A maintenance policy – be it corrective, scheduled based on time or condition-based – is usually elaborated based on the expected reliability behaviour of the system, in turn derived either from historical data, or from maintenance guidelines provided by equipment vendors and combined together according to experience about the use of such equipment (Jardine et al., 2006; Fumagalli et al., 2019). When a new system configuration is implemented, its reliability behaviour changes. It is however difficult to describe the new behaviour based on data, since there are no historical data yet. This is aggravated by the fact that the configuration may be combining the equipment in new manners, not considered during the design phase. For this reason, joining failure parameters of single equipment pieces in order to get a systemic model of the whole plant based on previous single equipment data, has in fact no solid ground.

Tool and cultural integration. As the boundary between lifecycle management and maintenance management becomes blurred for the reasons outlined above, decision aid tools have to answer questions originating from various needs (equipment choice and sizing, control design, performance evaluation, reliability forecast, and so forth) and

requiring data that are heterogeneous in both nature and source. But, most important, on the one hand these questions are formulated by professionals with very different roles, skills and backgrounds; on the other hand, given the approaching time scales just mentioned, the effects of the same questions will quickly interact with one another. To maintain control over so complicated systems, no doubt the various decision aid tools need coordinating, however there is also the necessity to establish a common jargon and culture for the involved professionals.

Summing up, the industry need of tools to support a rapid decision making in elaborating maintenance policies for reconfigured systems, is being strongly exacerbated. In the last decades, practitioners have developed supporting tools, tailored to efficiently solve a specific problem or decision making. In the same years, not only operations management has evolved a lot, but also mathematical modelling formalisms, and computational instruments as well, have undergone stunning improvements. The result is an incredible variety of domain- or purpose-tailored tools, impossible to list herein, among which a true integration – which entails much more than mere communication, interoperation or data sharing – at present is in fact but wishful thinking. Said otherwise to approach the scope of this paper, there is still a lack of a unique modelling and simulation tool that accompanies the assets and production systems throughout their lifecycle and that integrates different perspectives and operational objectives in such a way to be usable by different roles in industry. In recent years, the necessity for a unified conceptual framework to explain – and possibly orient – the crucial role of modelling and simulation in the industry, has given birth to ideas like that of “Digital Twin” (DT). However, in the opinion of the authors and apologising for being a bit *tranchant*, the very different meanings still given to that idea in both the literature and the applications, indicate that its introduction has nowadays gotten to decree the absolute need, not yet the realisation, of the mentioned integration.

The above said, we conjecture that Object-Oriented Modelling and Simulation (OOMS) can provide a valid methodological basis, also complemented with powerful technological platforms, to coordinately host all declinations of the DT idea – or at least, the major ones – so as to help and foster the above integration process. We here present and motivate the main pillars of the construction we are envisaging, whose ultimate scope – ambitious indeed – is to make DT not just an umbrella term for operationally separated practices, but a shared ground for the corresponding methods and tools.

The paper is organised as follows. Section 2 briefly analyses related work about modelling and simulation in Industry 4.0 – with a specific yet not exclusive maintenance-centric standpoint – to picture a highly interacting *panorama*, and further motivate our research. Section 3 introduces the basics of OOMS. Section 4 shows a few simple examples on how OOMS helps address the problems evidenced herein. Generalising, Section 5 points out the major issues to address, implicitly outlining a roadmap for the adoption of OOMS in reconfigurable manufacturing systems. Finally, Section 6 draws some conclusions and sketches out future activities.

2. RELATED WORK

The extensive use of sensors in manufacturing, and the increasing amount of data that can be acquired from them, gave rise to the idea of a Digital Twin (DT). Based on the literature (Negri et al., 2017; Tao et al., 2018), a DT may be considered in its roles of monitoring, simulating, control and decision-support in the lifecycle of the system through all its levels of details (Kritzinger et al., 2018; Macchi et al., 2018). Given the various roles attributed to a DT, we need to introduce a distinction among its functionalities. A DT can work in *offline mode* (not connected to the production field) simulating the system, analysing all the possible what-if strategies and defining some Key Performance Indicators (KPIs) or possible maintenance policies for the simulated *scenario*. It can also work in *online mode* (connected to the field) analysing the data collected from the system with respect to the KPIs computed offline to monitor possible deviations from the expected behaviour, from the current real one, from the sensor data, and to monitor the values that trigger the maintenance policies identified in the offline mode.

Simulation is already widely used in manufacturing environment for many purposes through the system lifecycle, and this enlarges the variety of analyses that a DT may contain to be as complete as possible. The use of simulation in manufacturing ranges from system and product design to operation planning and scheduling up to production performance analysis and system control (Polenghi et al., 2018). Nevertheless, the majority of DT applications use a data-driven approach (Cimino et al., 2019). This paper builds on two main pillars. One is physically centred first-principle modelling (Griffiths, 1992), that describes phenomena with systems of differential and algebraic equations (Fritzson, 2011). The other pillar is given by model-based approaches for maintenance in manufacturing (Jardine et al., 2006; Gertler, 2017; Simani et al., 2003), proposing the use of simulation, relying on Dynamical Models (DMs), to complement data-driven approaches (Wang et al., 2008; Jouin et al., 2016), and to overcome some of their limits. Indeed, DMs are getting more and more popular in the manufacturing context, as they can be used not only for maintenance but also to simulate all the possible what-if scenarios that may appear during the whole production lifecycle, to set a configuration or also to reconfigure a system. Simulation based on DMs can be used to populate the offline mode of a DT and can be really useful as a tool for decision-making support. Since it may be used to carry out a wide range of analyses, a structured methodology is needed to introduce it as a fundamental part of the asset lifecycle management. The main gap is identified by the complexity of the manufacturing systems. Different analyses can be performed on a system, and each analysis can be modelled with different models relative to the considered level of detail. As already said, a lot of tools exist to perform different analyses on manufacturing systems and the results obtained in different simulations are not integrated with one another. In conclusion, the paper proposes an OOMS-based methodology and a corresponding roadmap (in Section 5) to use simulation as a computational analysis tool for decision making and maintenance policy identification, combined with the integration of the different analyses.

3. MINIMAL INTRODUCTION TO OOMS

We now point out the characteristics of OOMS Mattsson et al. (1998); Casella and Leva (2006) relevant for our proposal. Forst, OOMS crisply separates the *model*, written as a set of equations, and the *solver* used for the simulation. One can select the most efficient solver for each study without having to modify the model. When that model has to serve different purposes, this is an advantage. Models are created in a modular manner, assembling components hierarchically, and components connect to one another through *ports* that do not require orientation. The components need not to be closed (as many equations as variables). Only their compound – including connection equations – needs to be. As such, one writes component models *independently of how they will be connected to others*. Ports allow to encapsulate the behaviour of a component entirely, so that one can have interchangeable models of the same component at various levels of detail. Moreover, OOMS tools obtain the simulation code by *symbolic manipulation* of the model equations. This allows to identify characteristics and properties of the system that originate not (only) from the components but also from their interconnection. The resulting code is particularly efficient and numerically robust. Finally, OOMS permits to mix equation- and algorithm- based parts, as well as time- and event-driven ones. A single model can contain a physics-based description of the plant, block diagrams for the modulating control functions, and discrete-event models such as automata for the logic ones. Also, controls can be represented either by equations, allowing for variable-step solvers to the advantage of simulation speed, or by algorithms, down to the *replica* of how the involved computing units are programmed, for maximum fidelity. All within the same paradigm and a single tool, without requiring co-simulation.

As OOMS – here, Modelica – tools, the major ones allow for calling external code (such as fluid property computation, exact *replica* of C code aboard the plant, and so on), connection with external devices – most frequently to synchronise simulator and plant, see e.g. Henriksson and Elmqvist (2011) – and other modelling/programming tools via the FMI (Functional Mockup Interface) standard (Blochwitz et al., 2011). As an example, Figure 1 shows a multi-physics and equation/algorithm model as seen in Modelica. One can recognise, from left to right, a simple SFC¹ program for logic control governing a PI (Proportional-Integral) velocity loop composed of digital and analogue electronics to drive a DC (Direct Current) motor; this motor, through a multi-clutch transmission, operates a pump to feed fluid to an exchanging pipe, which heats a compound of two thermal capacities subject to ambient conditions (temperature and radiative flux). All in the same tool, assembled and simulated together. Summing up, OOMS is a powerful paradigm supported by solid technology, also open source, increasingly adopted in high-end domains². We now move to illustrating how it can ground the integration we aim for.

¹ SFC (Sequential Functional Chart) is a graphical programming language for industrial controls defined in the IEC 61131-3 standard.

² The companies participating into the OpenModelica Consortium are listed at <https://openmodelica.org/home/consortium>, and the tools supporting FMI at <https://fmi-standard.org/tools/>.

4. A FEW APPLICATION EXAMPLES

We now show how to model a system through OOMS, illustrating – through four examples – the steps needed to build a model-based approach for a maintenance management of reconfigured systems. For convenience we take an extremely simple system, to allow seeing immediately how each statement reflects into the system equations. Needless to say, if we can evidence problems and OOMS-centred solutions in a system with just a handful of equations, the yielded advantages will be even more evident in a case of real-life complexity.

The system we consider is a bay of mass M carrying a payload of mass m . The bay moves on a linear guide of *abscissa* x pulled by a force F , with a return spring of elastic constant k_x , and subject to friction. The rest length of the spring corresponds to $x = 0$, and the friction force is simply proportional to velocity via a coefficient h_x ; F is exerted by a permanent magnet DC motor and a transmission of inertia J through a pulley of radius R ; the transmission, of angular position θ , is subject to a friction torque, proportional to angular velocity through a coefficient h_ω . Indicating with a dot the derivative with time and assuming the friction coefficient h_x to depend on the bay position x , the system is ruled by

$$\left\{ \begin{array}{l} (M + m)\ddot{x}(t) = F(t) - k_x x(t) - h_x(x)\dot{x}(t) \\ F(t) = R\theta(t) \\ J\ddot{\theta}(t) = \tau - RF(t) - h_\omega\dot{\theta}(t) \\ \tau(t) = k_\omega I(t) \\ V(t) = RI(t) + L\dot{I}(t) + k_m\dot{\theta}(t) \end{array} \right. \quad (1)$$

where τ is the motor torque, k_ω and h_ω its characteristic constant and friction coefficient, R and L its resistance and inductance, I the current flowing through it and V the applied armature voltage (the system input). We assume that, owing to wear, the guide can develop a localised friction. We suppose to know the position range $[x_1, x_2]$ of this possible defect (it is a matter of mechanical design) but cannot measure its entity. We need some predictive maintenance policy, but have no data and do not want to bring into play a model involving the microscopic phenomena that ultimately cause the defect, owing to the complexity of such a model and to the difficulty of parametrising its reliability. We briefly show how model (1) can help, through four conceptually consequent examples.

Example 1: determine the amount of localised friction to cause an unacceptable behaviour of x . At the detail level required for such a question, as any specialist would say, the defect can be represented by saying that in a certain region $[x_1, x_2]$ of the guide, the friction coefficient transitions from a “low” (normal) value $h_{x,LO}$ to a “high” (defect) one $h_{x,HI}$.

In this example there is no control aboard the system, i.e., the motor is operated “in open loop” and fed with an armature voltage step. Simulations like those of Figure 2 allow to relate the value of $h_{x,HI}$ to the trajectory of $x(t)$, hence determining its maximum tolerable value. One can repeat the simulation with different behaviours of the input, each one representing for example a newly introduced system manoeuvre, and determine the corresponding acceptable defect level in advance.

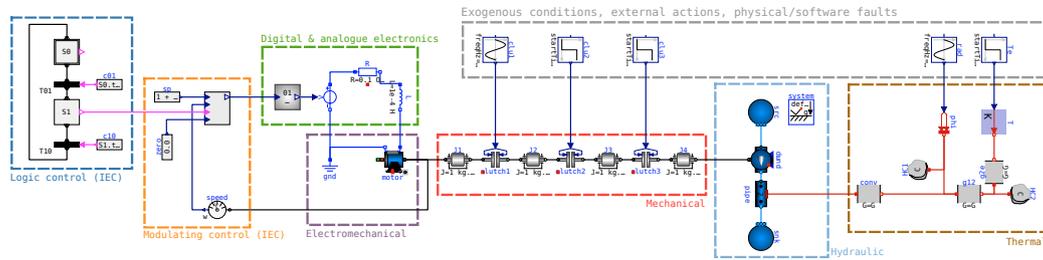


Figure 1. A multi-physics system, including modulating and logic control, as seen all together in an OOMS tool.

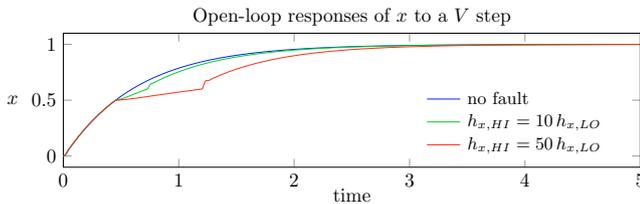


Figure 2. Determining the maximum defect level acceptable for a given operation.

Example 2: choose the best measurable indicator to forecast the fault. Condition-based (and predictive) maintenance assumes that a combination of measurable quantities can signal a forthcoming problem in advance, but the choice of these quantities should also consider that by setting thresholds we must be capable to predict abnormal behaviours. If we use the position x as predictor of a fault, we would not be able to identify a threshold surpassing which we can predict a problem (since in all cases x ranges from 0 to 1). Besides data analysis, a useful tool for the identification of the best combination of quantities to be measured is provided by OOMS. In fact, by sweeping the physical fault parameter (in our example $h_{x,HI}$) and by performing repeated simulations trying different combinations of quantities, it is generally possible to identify the best quantity (or combination of quantities) to describe and predict the forthcoming fault, and most important, to relate this choice to physical knowledge. Figure 3 shows that in our case a good candidate is the motor current I , that in relative terms is more sensitive to $h_{x,HI}$ than x , especially if the acceptable fault level is small.

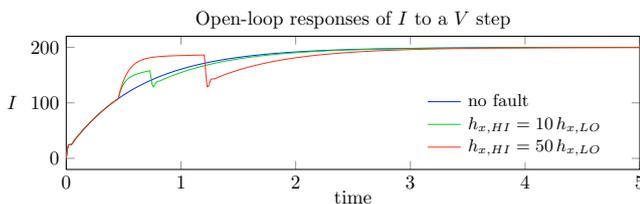


Figure 3. Determining a suitable fault prediction measurement.

Example 3: foresee changes to the “normal behaviour” induced by controls. A lot of manufacturing system come with their embedded control systems. Then, let us now assume that a PI controller is added to govern the bay position $x(t)$ acting V . This means that (1) is complemented with the controller equations

$$\begin{cases} \dot{x}_C(t) = \frac{K}{T_i}(x^\circ(t) - x(t)) \\ V(t) = x_C(t) + K(x^\circ(t) - x(t)) \end{cases} \quad (2)$$

where x_C is the PI state, $x^\circ(t)$ the reference trajectory, K and T_i respectively the PI gain and integral time. In this new system V is decided by the PI, and the input is $x^\circ(t)$.

Quite intuitively, also in the absence of faults, if one does not change the desired trajectory $x^\circ(t)$ but modifies the parameters of the PI, the behaviour of $x(t)$ – the “normal” one, as there is no fault – changes. In the presence of controls, there is no unique “normal behaviour”. There is one for each value of the control parameters, and these behaviours can differ significantly. Apparently, this means that when controls are retuned, maintenance policies may need retuning as well. To illustrate the role of OOMS in such a case, the system (1) together with the controller (2) can be simulated for various values of K – the other parameters staying unchanged and in particular no fault being introduced – and give as a result the so called closed-loop transients of x , shown in Figure 4. As can be seen, different controllers do give different trajectory behaviours. The maintenance policy needs reconfiguring so as to take as “normal” the one corresponding to the actually applied controller, and this one is given by OOMS.

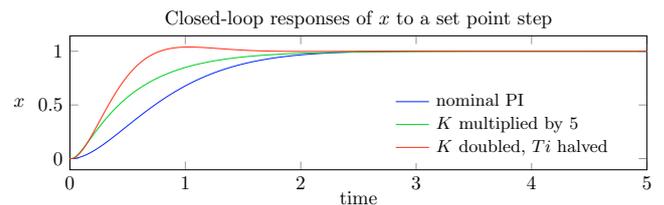


Figure 4. Observing changes in the “normal” system behaviour depending on the parameters of the applied controllers.

Example 4: re-check maintenance-related indicators when controls are reconfigured. As a deeper consequence of retuning controls, and even more of changing the structure of the control system, some maintenance-related indicators can loose significance while some others can preserve their efficacy. Figure 5 shows how x and I are affected by retuning the PI gain K , in the presence of the same fault. Note that changing the PI tuning the fault effect on x is reduced, while that on I is enhanced.

It is evident that the same maintenance policy on I can not be employed for each control strategy. Simulating the system with the right control parameters – that could change for any reconfiguration – allows to identify a proper retune of the related maintenance policies.

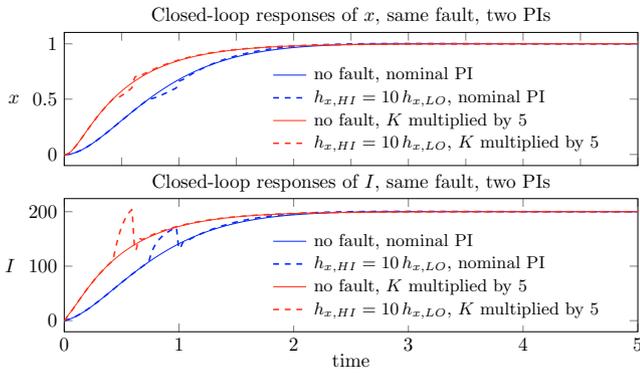


Figure 5. Effect of control modifications on the efficacy of maintenance-related indicators.

5. GENERALISATION AND OPEN PROBLEMS

Abstracting from the examples of Section 4, we now outline a roadmap for the introduction of OOMS as enabling technology for the maintenance of highly reconfigurable production systems. Along the distinction of Section 2, the simulating and modelling competencies of OOMS first cover the offline mode of a DT, and then provide online monitoring for all the maintenance decisions taken.

General ideas from example 1: Physical description better predicts fault propagation. As long as a system model (DTs included) is grounded on general physical laws, its parameters have direct physical interpretations. Moreover, as in OOMS component models are written independently of their interconnections, parameters just pertain to their individual behaviour, and most often can be drawn from data sheets. There is no need to “figure out” how a new compound of components will behave, this just emerges by simulating them together. Moreover, deciding the level of model detail required for a given study is the responsibility of the expert, and no software tool can replace his/her culture. However, simulating and checking the results on the plant floor is a means to validate such choices, and most important, the so gathered knowledge is secured into the realised models for subsequent experts to build upon. Finally, in OOMS physical models faults are injected as physical facts as well (for example, there is a friction coefficient that takes an abnormal value, or a resistance going to zero to emulate a short) and not just as boolean or lexical variables saying “this fault is happening”. This has (at least) two important consequences. First, the propagation of faults emerges from the model itself in the offline mode of the DT (there is a short here, hence *the model equations say that* some power increases, thus another component overheats, a thermal switch is triggered, and so forth): faults possibly generate other faults with the timing dictated by physics, thus allowing to forecast consequences and trigger maintenance or production stops timely. Second, should one want to train some ML/AI (Machine Learning/Artificial Intelligence) predictive maintenance tool on a new system configuration, its model can produce lots of training data useful in the online mode of a DT consistent with the new system’s physics, before the system is set into operation. Using OOMS this way can dramatically reduce the setup time for a new maintenance decisions.

General ideas from example 2: Physical descriptions allow for wiser ML/AI. Connected to the point above, it is common practice to determine fault forecast indicators by seeking correlations between data, be these measured on the real system or synthetically generated by a model. If ML/AI tools are used for this purpose, without introducing any physical awareness, a known risk is that a huge amount of data leads to forecast faults by correlations that are there essentially by chance in *those* data, and may lead to erratic decisions when applied to another data set. Using OOMS models is a viable way to exploit such physical insight, helping the analyst decide which indicators are better or possibly refusing some proposed by mere data analysis without real physical reason for it; the proposed indicators may be resulting from a peculiar situation, i.e. two manoeuvres occurring simultaneously. Further simulations can then be used to confirm or disprove such ideas. Also, OOMS tools allow for a sensitivity analysis of simulation results to parameters and inputs, which helps finding good fault forecast indicators.

General ideas from example 3: In the presence of controllers, OOMS allows easier system maintenance reconfiguration. When a system is reconfigured, controls often change as well (e.g., different settling times are needed for position loops). Hence, also the “normal” behaviour of the system can change, making some component alteration previously unacceptable now tolerable, or *vice versa*. This can be detected by simulation. Also, OOMS can generate lots of synthetic data before the reconfigured system is set into operation, allowing to pre-check maintenance policies also in the face of upcoming control requirements.

General ideas from example 4: OOMS helps find maintenance-related indicators based on control signals. Control can conceal equipment malfunctions. For example, a position control can cope with a progressively hardening friction, by just injecting more current. If this is not monitored, the system will eventually reach the condition in which not enough current is available. Looking only at the position one will thus see everything good, and at some point in time an abrupt malfunction. Not only a new configuration, but even just a control retuning, can make maintenance-related indicators more or less meaningful about the facts they are meant to sense. Once again, OOMS allows to examine such situations in advance, and take the convenient countermeasures.

5.1 Considerations and open issues for a roadmap

It should be established that OOMS needs integrating into both lifecycle management and maintenance, as the ability of simulating at various scales and detail levels allows for both studies that are local in scope and/or time, and whole-system ones concerning the entire asset lifetime. Also, though models may need specialising to one or another purpose, OOMS allows to embrace all such activities into one framework, and often one tool. This is another plus, as the alternative for multi-physics problems – the majority – is co-simulation, that is more complex and inflexible for the available time and the required agility to provide useful decision clues. To make OOMS unleash its potential, some problems need addressing. An OOMS engine is the core of the envisaged decision aid DT-based

toolbox, but further components are required. Filling the gaps summarised below is at present the envisaged research roadmap. From the operational viewpoint, first a database is needed to host component models, simulators, simulation runs, documentation and the like. This makes it natural to foresee an integration between OOMS and BIM (Building Information Modeling), but the BIM paradigm itself needs extending to also host information about the pool of solution engines to choose based on the properties of each study, as well as about the model analysis tools to be used. Furthermore, “role-specific” pre- and post-processors will need creating, to help each specialist input and see information the way he/she finds easiest and most informative. It will also be necessary to integrate with data-centric (e.g. ML or AI) tools. From the methodological standpoint, it will be necessary to define modelling practices to write component models in such a way to allow composing them with high-level tools – like the mentioned pre-processors – also by people who are not OOMS experts.

6. CONCLUSIONS AND FUTURE WORK

We discussed the use of OOMS in manufacturing, highlighting that (i) there are many tools tailored to specific problems, but (ii) no unique one suitable for decision support through a whole asset lifecycle—a setting even complicated by the introduction of DTs. We focused on DM-based techniques as fundamental decisions aids when many what-if *scenarios* need exploring without using data from the real system, rather only physical knowledge available at design time (a relevant feature when introducing innovative solutions to be assessed before implementation). Recognizing the value of having a DT in a manufacturing environment, we proposed OOMS for the offline mode of a DT, as a framework to collect all the DMs for each analysis – for each level of detail – that can be performed, in order to host in a unique platform the models - based on the physics of the systems - on which a DT is based. Used this way, OOMS supports the lifecycle vocation of a DT, in the initial configuration phase, during operations, and to support all the encountered reconfigurations. Future work is focused on applying the identified roadmap to a real case study in manufacturing, investigating also the integration of DMs within different levels of detail representation.

ACKNOWLEDGEMENTS

Support for this research work came from the Lombardia regional research and innovation project AD-COM, “Advanced COsmetic Manufacturing” (ID 214632).

REFERENCES

- Blochwitz, T., Otter, M., et al. (2011). The functional mockup interface for tool independent exchange of simulation models. In *Proc. 8th International Modelica Conference*, 063, 105–114. Dresden, Germany.
- Bortolini, M., Galizia, F.G., and Mora, C. (2018). Reconfigurable manufacturing systems: Literature review and research trend. *Journal of manufacturing systems*, 49, 93–106.
- Casella, F. and Leva, A. (2006). Modelling of thermo-hydraulic power generation processes using Modelica. *Mathematical and Computer Modelling of Dynamical Systems*, 12(1), 19–33.
- Cimino, C., Negri, E., and Fumagalli, L. (2019). Review of digital twin applications in manufacturing. *Computers in Industry*, 113, 103130.
- Fritzson, P. (2011). *Introduction to modeling and simulation of technical and physical systems with Modelica*. John Wiley & Sons.
- Fumagalli, L., Cattaneo, L., Roda, I., Macchi, M., and Rondi, M. (2019). Data-driven CBM tool for risk-informed decision-making in an electric arc furnace. *The International Journal of Advanced Manufacturing Technology*, 105(1-4), 595–608.
- Gertler, J. (2017). *Fault detection and diagnosis in engineering systems*. Routledge.
- Griffiths, G.W. (1992). *Process Dynamic Simulation: An Introduction to the Fundamental Equations*. AspenTech Technical Bulletin.
- Henriksson, D. and Elmquist, H. (2011). Cyber-physical systems modeling and simulation with Modelica. In *.e 8th International Modelica Conference*, 063, 502–509. Dresden, Germany.
- Jardine, A.K., Lin, D., and Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical systems and signal processing*, 20(7), 1483–1510.
- Jouin, M., Gouriveau, R., Hissel, D., Péra, M.C., and Zerhouni, N. (2016). Particle filter-based prognostics: Review, discussion and perspectives. *Mechanical Systems and Signal Processing*, 72, 2–31.
- 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.
- 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., Elmquist, H., and Otter, M. (1998). Physical system modeling with modelica. *Control Engineering Practice*, 6(4), 501–510.
- Napoleone, A., Pozzetti, A., and Macchi, M. (2018). A framework to manage reconfigurability in manufacturing. *International Journal of Production Research*, 56(11), 3815–3837.
- 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.
- Polenghi, A., Fumagalli, L., and Roda, I. (2018). Role of simulation in industrial engineering: focus on manufacturing systems. *IFAC-PapersOnLine*, 51(11), 496–501.
- Simani, S., Fantuzzi, C., and Patton, R.J. (2003). Model-based fault diagnosis techniques. In *Model-based Fault Diagnosis in Dynamic Systems Using Identification Techniques*, 19–60.
- 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.
- Wang, T., Yu, J., Siegel, D., and Lee, J. (2008). A similarity-based prognostics approach for remaining useful life estimation of engineered systems. In *2008 international conference on prognostics and health management*, 1–6. IEEE.