REPLICA: A Solution for Next Generation IoT and Digital Twin Based Fault Diagnosis and Predictive Maintenance

Rosaria Rossini, Davide Conzon, Gianluca Prato, Claudio Pastrone
IoT and Pervasive Technology Area
LINKS Foundation
Turin, Italy
{name.surname}@linksfoundation.com

João Reis, Gil Gonçalves
SYSTEC, Research Center for Systems and Technologies
Faculty of Engineering, University of Porto
Porto, Portugal
{jpcreis , gil}@fe.up.pt

***Abstract*—Nowadays competitiveness goes through several aspects: digitalization, productivity and environmental impact. Technology is advancing fast and helping industries to obtain more and more detailed data about their processes and equipment. In fact, the possibility to monitor and control each part of the process is a strong base on which a more intelligent and focused control can be built. Technology advance brings innovation and the possibility to manage the production in terms of ”near future” through AI prediction and decision-making support. Forecasting demands and planning production, optimizing process by reducing costs and improving efficiency without corrupting the quality of the product is a big challenge at the plant level. In this paper, a flexible, scalable architecture for intelligent digital twin realization called REPLICA has been proposed to cope with such problem and help industries to advance and discover possible optimizations. This architecture sits on top of two European projects, namely CPSwarm and RECLAIM, where their contribution focus on distributed simulation and optimization, and Adaptive Sensorial Networks, correspondingly. As a validation process, a hypothetical use case is presented, detailing the key differentiating points and benefits of the proposed architecture.**

***Index Terms*—IoT, Digital Twin, AI, Fault Diagnosis, Predictive Maintenance**

I. Introduction

In the era of Industrial Internet of Things (IIoT) and Industry 4.0, complex electromechanical systems can be equipped with a variety of sensors providing new opportunities for the development of Health Monitoring and Management Systems. These new opportunities target an optimum exploitation of available information in order to maximize the performance of the machinery and optimize the process. Focusing on the increase of production reliability and safety, as well as on the reduction of costs, there is an ever increasing industrial need not only for accurate and on time online diagnostics, but also for a robust and early estimation of the Remaining Useful Life (RUL) of the defected components, within a high confidence interval, independent of the operating conditions.

For this reason, and many others, a new concept of 'interaction' with the process arose; a concept of controlling and monitoring a replica instead of the real object, a perfect virtual replica that interacts with both humans and machines: a Digital Twin. The concept of "twin" is originally derived from National Aeronautics and Space Administration (NASA)'s Apollo Project when the aircraft's twin body was a real physical system [1]. Twin models help astronauts and staffs make decisions under emergency situations. Digital twin integrates the life cycle of a machine [2], and achieves a closed loop and optimisation of the machine design, production, operation, and maintenance, etc. In Magargle et al. [3], a multi-physical twin model is built to monitor the status of the brake system through multiple angles. NASA hopes to realise the health management and residual life prediction of the aircraft by building a multi-physical, multi-scale Digital Twin model [4], furthermore serveral roles are envisioned for Digital Twin in the industry 4.0 scenario [5].

The present paper focuses on the proposition of an intelligent digital twin architecture called REclaim oPtimization and simuLatIon Cooperation in digitAl twin (REPLICA), that focuses on two important aspects: 1) plug'n'play of models on demand and 2) Workflow design to orchestrate the models used in the Digital Twin itself. Aspect (1) aims to ease the integration and removal of models into digital twins, whenever a new version of the software is available or it performs any necessary correction in the used models. The goal of the latter aspect is dedicated to the creation of a pipeline that can manage the flow of data among all the available models. These models might be from pure data processing and decision making, from sensor and actuator integration, to third party synchronization with information systems such as Manufacturing Execution Systems (MES) or Enterprise Resource Planning (ERP). Most digital twins, and in particular intelligent digital twin architectures focus on providing the best set of models that one should have to accomplish, e.g. predictive maintenance, from the type of simulation required to machine learning models that should be refined based on newly acquired data. Furthermore, REPLICA can allow a flexible and distributed deployment in such a way that both completely cloud or mixed edge/cloud solutions deployment are accepted, with respect to the need of the specific application.

However, these architectures are normally rigid and do not support changing software models or even easily set up the orchestration of those resources. Fixed data flows are generally hardcoded, meaning that if an implementation needs to be modified, changes in code are required. This is often not recommended since these changes might negatively influence the stability of the system. To this intent, an architecture that addresses these challenges is presented.

The paper is organized as follows: in Section II, the authors introduce a literature review about predictive maintenance and fault diagnosis based on digital twin. Complementary, Section III presents the core technologies of the solution presented in this paper. Section IV describes the architecture of the solution proposed and Section V presents a first prototype implementation followed by Section VI in which a possible application is described. Finally, Section VII concludes the paper by summarizing and discussing the work.

II. Literature Review

With the rapid advancement of Cyber-Physical Production Systems, Artificial Intelligence (AI) and IIoT, Digital Twin (DT) has gained increasing attention due to its capability to adapt and replicate the industry processes. Accordingly to these changes, many different DT architectures have been proposed to realize several use cases in an intelligent and complex production system. Industrial AI [6] brings to the processes self-aware, self-adapt, and self-configure functionalities and facilitates the integration of the DT.

In [7], the authors propose to insert an intelligent DT in the Cyber Layer architecture. The concept has been partially realized with two industrial use cases, namely a modular production system as well as a metal forming industrial process to show its potential and gains over the challenges in Cyber-Physical System (CPS), i.e., synchronization throughout the lifecycle of a cyber-physical production system; development of the DT, which can contain different models; the interaction between DT, both for the purpose of co-simulation and operation data exchange; and the active data acquisition.

In [8], the authors present a methodology for enabling DT using advanced physics-based modelling in predictive maintenance. This methodology for advanced physics-based modeling aims to enable the DT concept in predictive maintenance application and consists of two main points: digital model creation and DT enabling. Then, the user is able to define, create and utilize the digital model of a resource, as well as its DT. The integration of DT and deep learning in CPS environment has been also proposed in [9] for the development and realization of smart manufacturing.

In [10], the authors present solutions for fault diagnosis based on DT. The paper includes an experiment and interesting results obtained with the software proposed. Compared to the solution presented in this paper, this work has a limited flexibility since it is only suitable for fault diagnosis.

In [11], [12] and [13], the authors show how it is possible to build a DT of machines and systems of systems to allow autonomous smart manufacturing, but these works, while interesting, are not specifically presenting a solution for fault diagnosis and predictive maintenance.

The authors of [14] introduce a solution for predictive maintenance of computer numerical controlled machines, based on DT. They demonstrate how the exploitation of a DT for predictive maintenance can provide better results compared to more traditional approaches. Even if this work provides a good example of application of DT for predictive maintenance, it doesn't aim to present a solution that can be leveraged in other scenarios.

In this paper, the authors intend to propose a novel architecture that supports several features missing in the other solutions presented above. Specifically, the proposed solution is not based on a set of fixed components but it can integrate heterogeneous modules, in terms of Internet of Things (IoT) sensors, AI algorithms and simulation tools, easing its customization in different use-cases. Furthermore, thanks to the flexibility guaranteed by the distributed nature of the system, the setting-up of the platform can easily be adapted to the each specific industrial infrastructure, selecting the most suitable mix of edge/cloud deployed components.

Moreover, the proposed architecture can support the creation at runtime of workflows both among the AI modules as well as between the IoT sensors and the models. This drastically reduces the time needed to run and collect results from the AI algorithms. In these terms, a possible process optimization can be quickly evaluated and eventually discarded if not appropriate. Finally, all the entities (sensors, AI modules and simulators) can be substituted following a plug&play approach that ease the adaption of the system to the changes in the physical world.

III. Core technologies

A. RECLAIM platform

Following the industry 4.0 paradigm, the business models of manufacturing companies need to be transformed, resetting their strategies to improve productivity and quality. The current maintenance strategies often require the user to manually analyse data collected to extract useful information from them and, furthermore, periodic human inspection is required to assess the real condition of the assets monitored.

Currently, the lack of continuous operation and health status monitoring tools and predictive maintenance solutions lead to unpredictable situations in industry like sudden machine operation failures. In this case, the current common procedure is to ask the intervention of technicians, which then try to repair and solve the problem. This causes several problems: it is time-consuming; it leads to production delays since the machine is stopped until it is not repaired; it doesn't support resources distribution. The industry 4.0 paradigm goes in the direction to address such problems through different actions: 1) re-manufacturing systems for material and resource efficiency, 2) increased flexibility in changing machine operation purpose,

3) application of big data analytics techniques, and 4) predictive analytics and model-based forecasts and optimization procedures, based on completely data-driven processes.

These four suggestions have been the funding principles of the RE-manufaCturing and Refurbishment LArge Industrial equipMent (RECLAIM) concept definition. The main objective of the project is to increase productivity, extending the lifetime of the machines and reducing the time and cost of machinery refurbishment and/or re-manufacturing. This objective will be achieved designing and developing a set of tools supporting several activities: from the monitor of machines' health status, to the implementation of adequate recovery strategy (e.g., refurbishment, re-manufacturing, upgrade, maintenance, repair, recycle, etc.). To achieve this, the RECLAIM outcomes will include two main components: an Adaptive Sensorial Network used to collect data and a Decision Support Framework (DSF) for optimization based on different criteria. Specifically one of the technologies supporting the DSF is the proposed REPLICA where simulation and optimization is used for fault diagnosis. The Adaptive Sensorial Network is one of the key elements to be used in the proposed architecture and is seen as an entry point for the essential data to be used.

B. CPSwarm Simulation and Optimization Environment

As indicated in [15], the CPSwarm Workbench - the set of tools released by the project for the development of CPS swarms applications - includes also a Simulation and Optimization Environment, used to evaluate the performance of a swarm solution. Such solution is composed mainly by: the Simulation and Optimization Orchestrator (SOO), which oversees the simulation and optimization tasks; a set of Simulation Managers (SMs), which provide common Application Programming Interface (API) to control heterogeneous Simulation Tools (STs); and an Optimization Tool (OT) used to perform the optimization processes. The network-based architecture is depicted in Fig. 1.

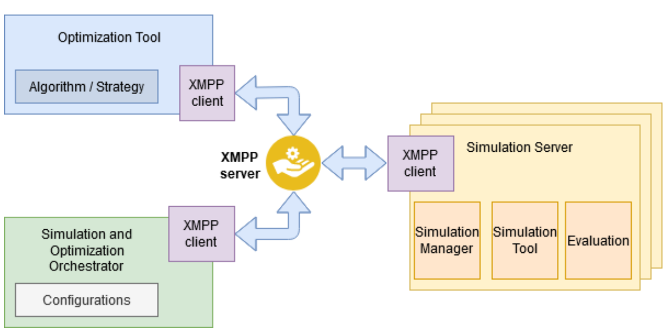

Fig. 1. Network-based Architecture from CPSwarm for Distributed Optimization and Simulation.

Such environment is useful both to simulate the behaviour of a designed swarm solution in a ST, leveraging the ST's Graphical User Interface (GUI) to evaluate its behaviour; and, on the other side, to optimize the controller parameters of algorithm/module, and possible aspects of the problem, i.e., the number of CPSs used, leveraging evolutionary design methodologies. In the latter case, candidate parameter sets are ranked based on a fitness score computed after the controller was executed with those parameters in a predefined environment. Successful parameter sets are then adapted to produce a new generation of candidates to be tested. This is a high time- and resource-consuming process, which requires a high number of simulation runs. To address this, the CPSwarm solution allows to parallelize the execution of these simulations, reducing the times required to complete an optimization. For this objective, the Simulation and Optimization Environment has a network-based architecture, allowing to parallely use a set of STs distributed on different machines [16]. This architecture has been implemented leveraging the eXtensible Messaging and Presence Protocol (XMPP) protocol, already tested executing multiple simulations on Robot Operating System (ROS)-based STs, i.e., Stage, Gazebo and Virtual Robot Experimentation Platform (V-REP). In the last release of the software (available as open-source on github[1]), a set of technologies have been integrated to improve its scalability and easy-to-use, i.e., docker and Kubernetes. Such final release has been tested, showing that it is able to scale till 128 SMs and that the time required to complete one optimization is inversely proportional to the number of STs used. Finally, a proof of concept has demonstrated the ability to deploy the controller with the optimized parameters onto CPSs.

The concept of distributed simulation and optimization is brought to the proposed REPLICA architecture by the CPSwarm results and the whole orchestration process and main building blocks are inspired by this project.

IV. Architecture

This section introduces the REclaim oPtimization and simuLatIon Cooperation in digitAl twin (REPLICA) architecture that has been designed to provide an infrastructure and be used for Digital Twin-based fault diagnostics and predictive maintenance solutions, which can be easily deployed and customize in different Industrial IoT environments.

REPLICA is composed by several modules (shown in Figure 2), mainly subdivided in two blocks: *Backend* and *Frontend*. The first one contains three main components: Artificial Intelligence (AI) Environment that hosts the AI modules, Digital Twin Orchestrator (DTO) that is used to orchestrate the operations done by the REPLICA and the Simulation Environment that is a distributed environment including several heterogeneous simulators deployed into different machines. The latter one instead contains two applications: one devoted to show the results obtained and another one for the configuration of the component. These modules will be described in the remainder of this section.

As explained in Section I, the DT concept concerns the integration of three main components: the data collected by IoT sensors; the realistic models of the real devices and the

[1]https://github.com/cpswarm/SimulationOrchestrator/wiki/Simulation-and-Optimization-Environment

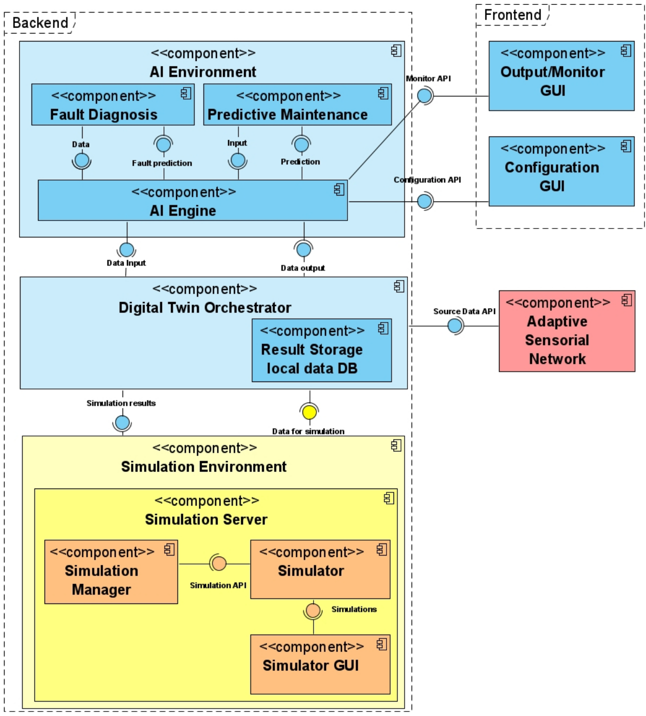

Fig. 2. REPLICA Architecture.

synchronization with those using the data collected; and a set of AI modules connected to these models. REPLICA fully supports this concept, providing the infrastructure to integrate these technologies.

In REPLICA, the Digital Twin Orchestrator (DTO) is the module in charge to manage all the IoT data flow coming from the field: machinery data, historical data and other data from legacy systems already present in the shop-floor. As described in Section III, these data are flowing through a component of the RECLAIM platform, the *Adaptive Sensorial Network*. Furthermore, the DTO is in charge to create the correct flow among the AI modules running in the AI environment and the machine models running in the Simulation Environment. Finally, the DTO oversees the storing and organization of the processed and simulated data, which are saved in a local database.

In REPLICA every machinery of interest has a corresponding realistic model running in one of the simulators integrated in the *Simulation Environment*. Specifically, the *Simulation Environment* is a distributed environment based on the one presented in Section III-B. Similar to what has been presented for the original solution about swarm intelligence, the environment supports a set of heterogeneous simulators distributed in different machines. Each of these simulators is wrapped by a *Simulation Manager*, and the role of this component is to abstract the functionalities provided by the simulators using the standard API exported by the DTO. In this way, the DTO can: 1) control the simulators to run the required simulations; 2) inject the data needed to keep the models synchronized with the real machines; 3) inject in the simulators data produced by the AI algorithms (for example to simulate failures); 4) receive from the simulators the produced results. Finally, the *Simulation Environment* supports for each integrated simulator one advanced Simulator GUI that allows to monitor the simulated device. This GUI is the one integrated in the simulator, which provides a graphical representation (also 3D) of the simulated device.

Finally, in REPLICA the AI algorithms are hosted and executed in the AI Environment. This environment allows to host and run heterogeneous algorithms for fault diagnosis and predictive maintenance. Besides the algorithms, the environment also hosts a module called AI engine. This module has the objective to orchestrate the algorithms creating the needed workflows among them. Furthermore, the AI Engine uses the API provided by the DTO to interconnect the AI modules with the machine models and the data coming from the shop-floor.

The architecture is completed by the two interfaces in the *Frontend*: the OutputMonitor GUI is used to monitor in real time the results produced by the running solutions in a user friendly interface. Instead, the Configuration GUI is leveraged by the users of the system to configure the AI engine for the needed tasks.

The proposed architecture aims to provide the following key features: 1) Allowing the integration of heterogeneous components in terms of sensors data collected from the field, AI algorithms and simulators running accurate machine models in the shop-floor; 2) Supporting the creation at runtime of workflows not only among both AI modules and the IoT sensors, but also among themselves; 3) Supporting the plug&play at runtime of the IoT sensors, the AI modules and machine models, without the need to restart the system; 4) Easing the adaptation of the digital twin to the changes in the physical world; 5) Enabling a flexible and distributed deployment: supporting both completely cloud or mixed edge/cloud solutions, based on the need of the specific application.

A first partial implementation of the proposed architecture and a set of possible future works for the part not yet implemented is presented in the next section.

V. Preliminary implementation and future prospects

This section presents the first prototype of the proposed architecture. The solution is a combination of newly developed components and the evolved version of components already developed in previous European Union (EU) projects. For the new components, this section will introduce only some possible technologies that the authors are evaluating and testing so far to leverage and implement the architecture, while for the existing components a more concrete implementation is presented. More specifically, the software already implemented is one algorithm for predictive maintenance, one for fault diagnosis and a distributed simulation environment already developed in CPSwarm, while the components yet to be implemented are the AI environment, the AI engine, the DTO, the Configuration GUI and the OutputMonitor GUI.

The authors have chosen to base the AI environment on a docker container based solution. Each predictive maintenance and fault diagnosis algorithm will be wrapped in one container. In this way the AI environment will support the integration of AI modules based on different technologies.

Two examples of solutions currently supported in the AI environment are the fault diagnosis and predictive maintenance modules presented in Fig. 3 and 4. The fault diagnosis module is composed by techniques to find abnormal behaviors that deviate from normal process conditions to raise warnings and find root causes for the problem. This algorithm will be fed directly with sensor data (when possible and pertinent) or transformed data from the field in order to be more interpretable. Based on the analysis of data streaming, the algorithm should indicate if a warning should be sent to the key personnel to check the system. This algorithm is the first front-line of analysis from shop-floor components in order to understand machine's health.

Additionally, the predictive maintenance module is composed of 1) a component failure prediction in the future (e.g. 48h and which maintenance action should take place); 2) Optimization module for scheduling future maintenance actions based on the existing scheduling; 3) Simulation module that aims at assessing the impact of changes in the machine and shop-floor [17]. The main idea of this method is to predict what kind of maintenance and when it will be required based on the failing component in the machine. With this, it will be possible to understand what changes need to be done in order to compensate the downtime of the failing machine.

As can be seen from Figure 3, the implementation already follows a block based approach which allows a better flexibility once building the required data workflows among models. For this particular case, the Dynamic INtelligent Architecture for Software MOdular REconfiguration (DINASORE) [18] platform was used, which is a run-time environment developed in python language for the International Electrotechnical Commission (IEC) 61499 standard [19] and integrated with the Eclipse based Framework for Distrubeted Industrial Automation and Control (4DIAC) Integrated Development Environment (IDE) [20]. Moreover, this implementation does not only allow for the orchestration of models, but also the plug&play of such models in a distributed system, where software can be reconfigured on-demand. This supports both completely cloud or mixed edge/cloud systems, depending on the required application and the number of machines available for execution. Finally, since DINASORE is implemented in python, the state of the art implementations of AI can be promptly used.

For the implementation of the AI Engine that interconnects the AI modules, the authors have already evaluated several solutions. One is the possibility to run the modules in a docker environment, making them read and write from text files located in specific folders and then interconnect them through a software that allows to handle the workflow, e.g., Node-red or NiFi. Another evaluated solution is the possibility to use Acumos AI to implement the AI environment; in this case, the deployment of the containers, the interconnection of the components and the workflow will be handled by tools included in the framework. At the moment of writing, the final solution to be used is still under evaluation and the authors are investigating if Acumos AI satisfies all the needed requirements of the AI environment, particularly focusing on the possibility to add and remove AI modules at run-time and the dynamic change of their interconnections to create new workflows.

For the implementation of the DTO and the *Simulation Environment*, the *Simulation and Optimization Environment* solutions provided by CPSwarm will be leveraged and extended. Specifically, REPLICA will incorporate the communication API based on XMPP and the deployment system, based on docker and Kubernetes [21]. The use of these technologies will allow to integrate heterogeneous simulators, to simply deploy and run the simulations needed on distributed machines, add and remove at run-time the simulators running different models. More specifically, in the *Simulation Environment*, considering that the solutions proposed in CPSwarm was integrating only ROS based simulators, new types of simulators, e.g., java based simulators, will be included during the RECLAIM project. For this scope, a specific SM will be developed for the required simulators and the API will be refactored to support also these new simulators. Beside the SM, also docker containers to easily deploy such simulators will be created. Instead, for the implementation of the DTO, the authors have defined that the SOO implemented in CPSwarm will be completely refactored and extended to support the functionalities required by REPLICA. Additionally, only some of the functionalities of the SOO will be leveraged, extending them to support data storage and data analysis features. Finally the DTO will provide a set of API based on some standard technologies, e.g., Message Queue Telemetry Transport (MQTT), which will allow to collect data to be used by the algorithms and to keep the models updated and synchronized with the physical world. Thanks to these API, the DTO will be able to collect data from heterogeneous devices that, in the RECLAIM platform, are integrated through the IoT Gateway (see Section II). Also in this case, it will be possible to add and remove devices at run-time, without the need to restart the system. The new devices can be immediately used by the solution developed, just after they have registered themselves.

Finally, for the implementation of the GUI included in the architecture, the presented modules are in different phases of development. Specifically, for the Configuration GUI, the authors have not yet chosen how to implement it and different solutions will be evaluated, keeping in consideration a thorough integration with the rest of the platform. Instead, for the Output/Monitor GUI, for the monitoring and assessment of the results of the algorithms, a simple implementation based on the work done in CPSwarm is already available. Specifically, this solution is based on Thingsboard, which has been used to develop two different GUI: one for process monitoring and another one for the assessment of results. The first one shows live data in a chart and allows the monitoring of the process;

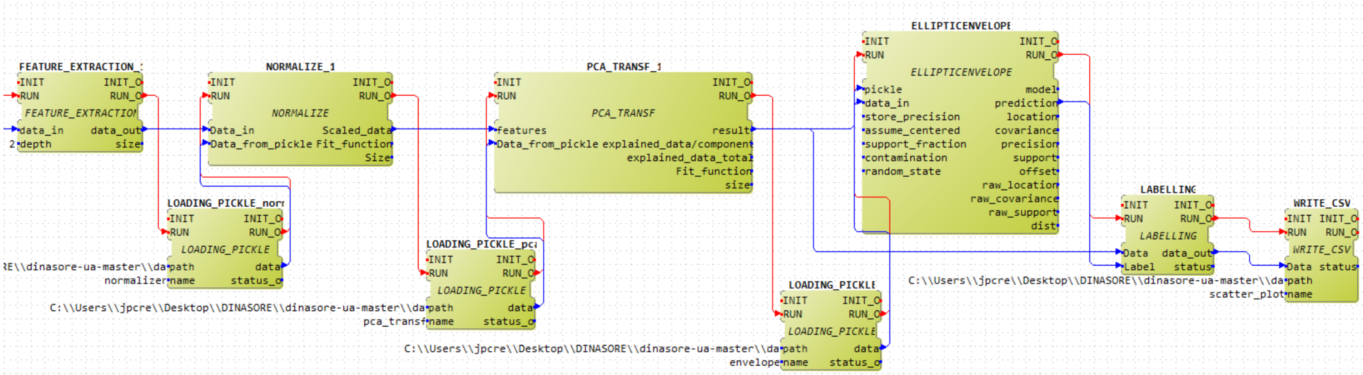

Fig. 3. Fault Diagnosis algorithm.

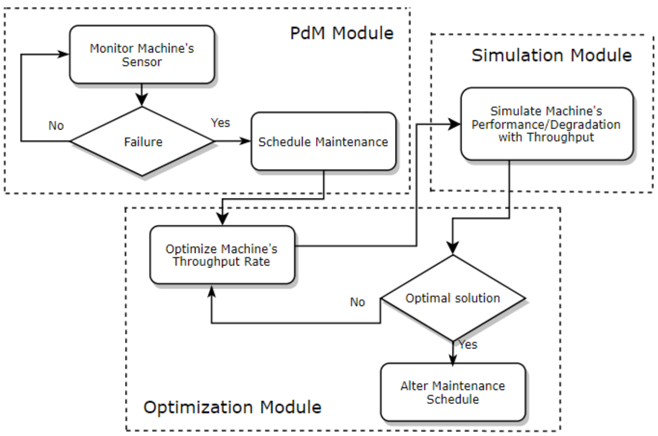

Fig. 4. Predictive Maintenance algorithm [17].

the latter one, instead, shows the data in a table, where it can be sorted by column (one column for each data). These GUIs have been used in this first implementation, but the possibility to enhance or completely replace them with something different, based on the requirements of the solution proposed, will be evaluated in the future.

VI. RECLAIM USE CASE

The aim of this section is to present a possible application of the solution presented in this paper, focusing on the predictive maintenance and refurbishment of a large Woodworking Production Line. The main objective of such use case will be to show the benefits of the adoption of advanced maintenance strategies in a large scale industrial scenario.

The selected scenario presents different challenges: firstly, the need to integrate in a single environment both heterogeneous data collected by installed sensors at the shop-floor and AI modules; together with the realistic models of the machines, enabling and easing the construction of a shop-floor's digital twin. Moreover, the proposed solution will have to optimize the use of a large industrial equipment providing novel machine learning solutions able to monitor the current system status and predict possible failures. In particular, the exploitation of fault diagnosis and predictive maintenance techniques based on the use of digital twin will increase the efficiency of the maintenance activity with respect to the performance obtained with the traditional methods based on a fixed schedule and a simple telemetry analysis.

The flexibility and adaptability of REPLICA can be well demonstrated both during the system setup in the industrial site and in case of replacement of one machine in the Woodworking Production Line.

In the first case, when the platform is going to be deployed, the data collected by the *Adaptive Sensorial Network* and a set of AI algorithms for fault diagnosis and predictive maintenance developed by different analysts have to be integrated. Furthermore, to allow the simulation of different operative scenarios, the realistic models of different machines have to be imported and executed in one simulator.

Using REPLICA, the effort to make all these components provided by heterogeneous vendors working together is significantly reduced, allowing their integration by simply exposing their inputs/outputs through the REPLICA defined interfaces. In particular, for the machine simulation, if the simulator used to execute it is already supported by REPLICA, no further developments are needed; otherwise only the SM for that simulator needs to be developed to allow its integration with the rest of the solution. The same advantage can be considered also for the AI algorithms: if the ones already integrated in the AI environment are suitable for the specific case, no developments are necessary and the platform should be just configured to enable the correct flow of data among different components. Otherwise, to integrate a new algorithms, the only requirement is to implement the inputs/outputs API defined in REPLICA.

Once all the components are connected, the AI modules can be trained using the data coming from the *Adaptive Sensorial Network* (as it is usually done), but also with data produced by the simulated machines. The integration of this secondary source of data not only allows to speed up the training process (more data available means less time to learn) but also to add the possibility of using data that are generally more difficult to collect, such as the one associated with specific failures that, for obvious reasons, are not so common in a real industrial plant.

The advantages of the REPLICA solution can be further demonstrated taking into consideration the scenario of a machines part replacement in the production line which is already monitored by the system. In this case, the administrator of the platform needs to update the components used for the fault diagnosis and predictive maintenance to reflect the new situation on the field. In a traditional system, this is a process that requires a complete shutdown of the system in order to setup and reconfigure it; instead, by using REPLICA the process is fast and mainly automatic. Indeed, for the replacement of the simulation models, this can be done just removing the old simulator and instantiating a new one with the updated model, taking advantage of the integration of the solution with Kubernetes. Once the updated simulator is instantiated, the same process can be applied to the AI algorithms, which can replace the previous ones. All these updates can be done without the need to interrupt the execution of the system. Obviously, if this change requires the integration of some new AI modules or the development of a new SM, these have to be implemented in advance. Also the workflow of the components need to be updated to interconnect the new ones, and can be done simply by using the tools provided by AI engine and DTO. These will automatically update the workflow to reflect the status of the components available in the system and that allow the administrator to easily create the new workflows. Finally as for the previous use-case, REPLICA can also be used to speed up the training time needed to have the new algorithms ready to be used, supporting the use of simulated data, instead of using the actual devices.

VII. CONCLUSION

The paper has presented REPLICA, a solution that enables the application of innovative fault diagnosis and predictive maintenance techniques based on DT. The software proposed is part of a more complex platform developed by the RECLAIM project that provides a complete software solution for both extending machinery lifetime while also improving productivity and performance.

The paper shows the main key innovations introduced by REPLICA in terms of fault diagnosis and predictive maintenance techniques based on DT. Specifically, one of the basic concepts of REPLICA is to build the solution not as unique suite of technologies, but as a run-time environment with APIs that allow the integration of heterogeneous AI modules and simulators. In this way, the solution can be easily customized to be used in different industrial sites, integrating components provided by different vendors, without requiring the development of new components, but just adapting the existing ones. Furthermore, using REPLICA the user can integrate and replace AI modules and simulation models at run-time, without the necessity to stop the system and reconfigure it. REPLICA provides tools to interconnect among each other the different modules with the data collected from the field. This allows the creation and the modification at run-time of the workflows needed for fault diagnosis and predictive maintenance, adding/removing/replacing entities to reflect the situation of the components available in the system.

In this work, the authors introduced the details of the first implementation of the proposed architecture; since the development currently is only in a preliminary phase, Section V presents the implementations of the modules that were evolved from previous EU projects' outcomes. Instead, for the new components only some initial design choices are presented. In particular, the AI environment and the DTO have been just designed and will be developed in next phases of the project, while the *Simulation Environment* is already available and only some SMs for new simulators will be developed. In the same way, a dashboard for monitoring and results assessment is ready, while the GUI for configuration has still to be implemented.

Finally, the authors have provided in Section VI two use cases based on one realistic industrial scenario, which show the advantages of using the proposed solution to apply fault diagnosis and predictive maintenance techniques based on digital twin.

ACKNOWLEDGMENT

The work presented here was part of the project ”RECLAIM- RE-manufaCturing and Refurbishment LArge Industrial equipMent” and received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 869884.

REFERENCES

[1] R. Rosen, G. [von Wichert], G. Lo, and K. D. Bettenhausen, “About the importance of autonomy and digital twins for the future of manufacturing,” *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 567 – 572, 2015, 15th IFAC Symposium onInformation Control Problems inManufacturing.

[2] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui, “Digital twin-driven product design, manufacturing and service with big data,” *The International Journal of Advanced Manufacturing Technology*, vol. 94, 02 2018.

[3] R. Magargle, L. Johnson, P. Mandloi, P. Davoudabadi, O. Kesarkar, S. Krishnaswamy, J. Batteh, and A. Pitchaikani, “A simulation-based digital twin for model-driven health monitoring and predictive maintenance of an automotive braking system,” in *Proceedings of the 12th International Modelica Conference, Prague, Czech Republic, May 15-17, 2017*, 07 2017, pp. 35–46.

[4] E. Negri, L. Fumagalli, and M. Macchi, “A review of the roles of digital twin in cps-based production systems,” *Procedia Manufacturing*, vol. 11, pp. 939–948, 12 2017.

[5] ——, “A review of the roles of digital twin in cps-based production systems,” *Procedia Manufacturing*, vol. 11, pp. 939–948, 12 2017.

[6] J. Lee, H. Davari, J. Singh, and V. Pandhare, “Industrial artificial intelligence for industry 4.0-based manufacturing systems,” *Manufacturing Letters*, vol. 18, pp. 20 – 23, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2213846318301081

[7] B. A. Talkhestani, T. Jung, B. Lindemann, N. Sahlab, N. Jazdi, W. Schloegl, and M. Weyrich, “An architecture of an intelligent digital twin in a cyber-physical production system,” *at-Automatisierungstechnik*, vol. 67, no. 9, pp. 762–782, 2019.

[8] P. Aivaliotis, K. Georgoulias, Z. Arkouli, and S. Makris, “Methodology for enabling digital twin using advanced physics-based modelling in predictive maintenance,” *Procedia CIRP*, vol. 81, pp. 417–422, 2019.

[9] J. Lee, M. Azamfar, J. Singh, and S. Siahpour, “Integration of digital twin and deep learning in cyber-physical systems: towards smart manufacturing,” *IET Collaborative Intelligent Manufacturing*, vol. 2, no. 1, pp. 34–36, 2020.

[10] Y. Xu, Y. Sun, X. Liu, and Y. Zheng, "A digital-twin-assisted fault diagnosis using deep transfer learning," *IEEE Access*, vol. PP, pp. 1–1, 01 2019.

[11] K. Ding, F. T. 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*, vol. 57, no. 20, pp. 6315–6334, 2019. [Online]. Available: https://doi.org/10.1080/00207543.2019.1566661

[12] G. Zhou, Z. Chao, L. Zi, K. Ding, and C. Wang, "Knowledge-driven digital twin manufacturing cell towards intelligent manufacturing," *International Journal of Production Research*, 04 2019.

[13] M. Borth, J. Verriet, and G. Muller, "Digital twin strategies for sos 4 challenges and 4 architecture setups for digital twins of sos," 05 2019, pp. 164–169.

[14] W. Luo, T. Hu, Y. Ye, C. Zhang, and Y. Wei, "A hybrid predictive maintenance approach for cnc machine tool driven by digital twin," *Robotics and Computer-Integrated Manufacturing*, vol. 65, p. 101974, 10 2020.

[15] M. Jdeed, M. Schranz, A. Bagnato, S. Suleri, G. Prato, D. Conzon, M. Sende, E. Brosse, C. Pastrone, and W. Elmenreich, "The cpswarm technology for designing swarms of cyber-physical systems," in *Proc. Int. Conf. on The Research Project Showcase of Software Technologies: Applications and Foundations (STAF)*, Jul. 2019.

[16] M. Rappaport, D. Conzon, M. Jdeed, M. Schranz, E. Ferrera, and W. Elmenreich, "Distributed simulation for evolutionary design of swarms of cyber-physical systems," in *Proc. Int. Conf. on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE)*. IARIA, Feb. 2018, pp. 60–65, ISBN: 978-1-61208-610-1.

[17] L. Antão, J. Reis, and G. Gonçalves, "Continuous maintenance system for optimal scheduling based on real-time machine monitoring," in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1. IEEE, 2018, pp. 410–417.

[18] Digital and I. I. Lab. Dinasore - dynamic intelligent architecture for software and modular reconfiguration. [Online]. Available: https://digi2-feup.github.io/dinasore/

[19] G. Cengic, O. Ljungkrantz, and K. Akesson, "Formal modeling of function block applications running in iec 61499 execution runtime," in *2006 IEEE Conference on Emerging Technologies and Factory Automation*, Sep. 2006, pp. 1269–1276.

[20] T. Strasser, M. Rooker, G. Ebenhofer, A. Zoitl, C. Sunder, A. Valentini, and A. Martel, "Framework for distributed industrial automation and control (4diac)," in *2008 6th IEEE International Conference on Industrial Informatics*, July 2008, pp. 283–288.

[21] C. Consortium, "Finalintegration of external simulators, d6.7," Deliverable of the CPSwarm Project, 2020.