Generic Framework for Developing Process Digital Twin Applicable to High Value-Added Manufacturing

Ram Upadhyay, UES LLC, Schenectady, NY 12309 and
Domenico Borzacchiello and Jose Aguado, Miura Simulation SAS, 1 rue de la Noë, 44321 Nantes, France and
Umang Garg and Vivek Arora, Nagarro Inc., Gurugram, India

ABSTRACT

Our team has developed a generic modular framework to create Process Digital Twin (PDT) that can be customized for any process. Elements of this framework are (i) configuration of manufacturing process inputs as seen by user, (ii) capturing variation of manufacturing inputs (iii) build accurate and fast multi-parametric numerical solution using advanced model order reduction (MOR) techniques, (iv) Extract features for quality (FFQ) using specialized algorithms and (v) predict probability of success from quality models and (vi) capability to receive post-production process and quality data and use machine learning algorithms to automatically update various models. Irrespective of the physical and geometrical complexity of individual process, our numerical techniques assure a real time (< 10 sec) multi-parametric solution. Primary output of process digital twin (PDT) is predicting probability of success of a process based on all manufacturing inputs both material and physical. This allows us to use PDT to maximize production success by adjusting controllable inputs apriori. We have successfully used the framework to build digital twins for complex composite manufacturing processes that produce fan blades for aircraft engines and implemented it into manufacturing process. We will present some examples to demonstrate the paradigm shift of reactive to adaptive manufacturing.

1. INTRODUCTION

Manufacturing has been slow to incorporate digital/information technology tools in production processes due to perception of high skill requirements, complexity and reliability. Data collection systems as part of processing has been used for over last 3 decades has become much cheaper. However, actual usage of data in real-time decision making is still very limited and it is an exception rather than a normal practice. One of the first successful attempts at improving producibility on a continuous basis was production of composite fan blade for GE90 engines using a custom system called "eMan" described by Upadhyay and Sinha [1] which provides a historic perspective of the challenges in process design and the path to a robust manufacturing process.

Figure 1 shows typical stages of digitization also referred as Internet of Things (IoT) as digital/software technology sophistication is deployed and how it adds value to product manufacturing. Each stage draws on engineering and software technologies of data acquisition, analysis methods, six-sigma tools and control and IOT methods. Value added to a process at any stage would greatly depend on the specific challenges i.e. inherent variability, technology gap in control methods, faster change over to next generation products or handling unknown current and future variability. Investments in different stages will highly depend on complexity of process, quality needs, rates of production following the usual return on investment trade-offs. Let us look at each stage in some detail.

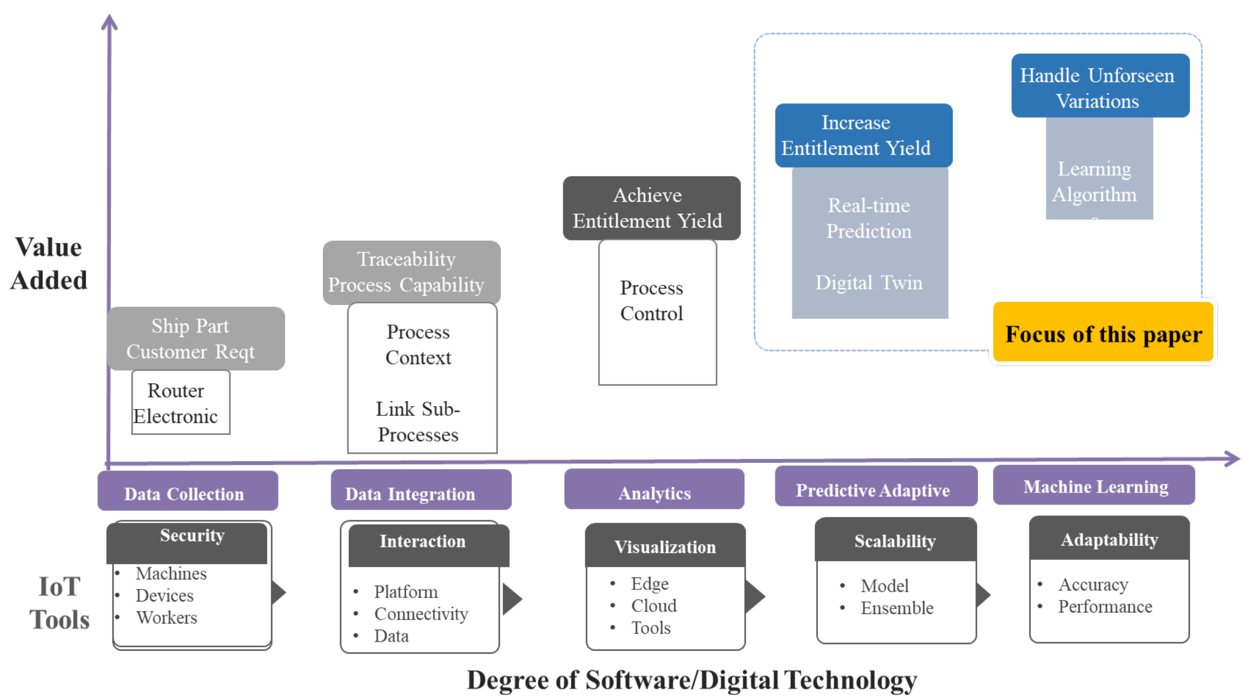

Figure 1: Digitation Journey for smarter factory showing stages of digitization, enabling technologies and added value

1. Data collection: An IoT system is designed to enable data collection across different points of interest which can be machines, actuators, sensors or workers. The data can be process, device, product or worker related parameters. The data collection process can be manual, semi-automatic or automatic. Most common method of data collection manually is by reading sensors, gauges or machine parameters and recording them in a paper router. This approach is feasible and cost effective only for small production quantities. A step towards electronic data collection starts with custom Excel based applications. Actual transformation of manufacturing process starts when a software application like an MES (Manufacturing Execution System) or a custom data collection system is deployed using databases. Final degree of digital maturity evolves to a real time data collection system when the MES or custom data collection software is integrated with machines to capture the process and quality data from machines using IOT sensors for real time decision making.

2. Data Integration: The data collected from different sources in a smart factory using different applications may be stored in variety of databases. This leads to data silos which reduces the effectiveness of using this data for analysis in subsequent stages. To manage this problem of data silos, some form of data lake can be created wherein data from various sources in their natural format enabling the integration of data not just from storage perspective but also from domain perspective. With respect to domain, this step covers developing and using specialized feature extraction algorithms on the data lake to visualize and achieve the benefits like traceability and correlation for an individual or a group of part. Integration of data is also the least recognized challenge which (i) provides context and linkage to individual part for data coming from direct measurement or indirectly thru a batch process and (ii) separates dependent (inspection and quality) and independent variables based on process knowledge.

3. Analytics and Process Control: Once we have data for set of parts and dependent/independent variable identified, there are many analytical tools and techniques such as statistical process analysis can be used to learn about the process. The first insight is the process capability of key process variables (KPV) which can identify and prioritize the areas for improvement. We also start developing correlations for yield with KPV's and allow us to set new process limits for reducing defects. Good analytics help us reach our entitlement yield (maximum possible yield by repeating the same process steps for every part). By developing yield correlations and transfer functions, we can implement useful process control methods to maintain entitlement yield. The results of the analysis can be viewed in responsive dashboards where each element of the process can be analyzed through drill down analysis.

4. Predictive and Adaptive: This is the stage of development where we not only capture all the knowledge using analytics but we actively try to adapt a priori for the incoming variation from various sources by making some changes to process for each part to maximize probability of success. This can be accomplished by building a process digital twin that can be exercised to maximize yield thru an optimizer used by manufacturing system. Primary benefits from this step leads to significant increase in yield above regular entitlement. Primary thrust of this paper is to develop, mature and formalize the concept of Process Digital Twin (PDT).

5. Machine Learning: We are in very early stages of development of Learning technologies. Learning and predicting within a defined domain has shown sign of success but learning in one domain and extrapolating to new domain is only in conceptual stage. However, there is good potential to use machine learning algorithms to continually update our quality models that predict performance. This can have profound impact on pace of maturing new processes and speedily resolving previously unseen quality issues.

Current state of the composite processing is mostly limited to first three stages. Moving to stage of predictive and adaptive manufacturing requires a paradigm change where biggest technical challenge has been real-time (<10 sec) prediction of process outcome. A process digital twin (PDT) is required to accomplish this paradigm change. The primary purpose of the current paper is to provide a theoretical foundation for process digital twin and machine learning along with a framework for seamless integration into manufacturing process without compromising the fidelity of knowledge, models and algorithms.

In order to move from a reactive to a proactive manufacturing, system needs to be able to leverage a range of digital technologies to enable faster decision making. The vision of smarter factories that would enable real time decisions is largely dependent on effectively using IoT technologies and thereafter deriving insights from the data that is collected. An IoT driven system in first stage would collect data for analysis. In the second stage this data collected would be used for taking decisions, often offline. In stage-3 some of these decisions would be made in real time. In Stage-4 the IoT sensor data can be used to make predictive analysis on machine performance or process performance and thereby enable a *virtuous cycle* of self-learning and adaption. Digital twins are best suited to enable Stage-4 of a manufacturing systems IoT journey.

2. THEORETICAL – PROCESS DIGITAL TWIN FORMALISM

Concept of digital twin has historically come from flight simulators for pilots where it was important to train without high risk. Today there are primarily two types of digital twins – product and process digital twins that have very distinct features.

Product Digital Twin predicts performance of a product based on its operating conditions and is fairly advanced in its development and usage. There are many examples like managing a wind farm based on the predicted performance of individual wind turbines, managing a fleet of aircraft engines for best life performance and predictive maintenance, managing specialized health equipment like MR system at a hospital level. In all these cases each wind turbine, aircraft engine or MR system has a digitized version which can precisely predict the performance and then be used to pro-actively manage a large system based on the overall objective.

Process Digital Twin (PDT) predicts probabilistic outcome of a process based on manufacturing inputs both material and physical i.e. what is the probability of producing an acceptable part. The predictability feature now allows us to take actions a priori that will maximize the probability of success. As contrary to wide spread belief, Process Digital Twin is not just Process Models. One of the primary functions of PDT is to integrate all existing information about whole manufacturing process i.e. predictive physical models, heuristic knowledge, quality criteria, quality transfer functions. Here we define a standard for PDT with a set of feature requirements that enables a seamless integration into manufacturing environment in real-time. These features are:

1. Inputs to PDT are exactly same as actual manufacturing process
2. Always predicts outcome of the process in terms of probability of success or yield
3. Real time response – we say faster than 10 sec for it to be viable in manufacturing decision making and even faster to embed in machine controls

4. Capable of integrating in-process sensor data and physical models and extracting key features necessary for quality
5. Capable of receiving post-process quality results. This enables continuous Machine Learning to identify new quality features and improve logic to predict quality

First four features are essential for any real application. Last feature related to machine learning can evolve over time as it focuses on improving quality predictions on faster time scale. Each of the above features impose significant technical requirements and will be explored further.

2.1 Process Digital Twin Framework

Based on the technical needs from all the features, Figure 2 shows various elements and functionality of a typical Process Digital Twin with following observations:

1. The inputs are separated into four groups (i) controllable process inputs where different choices may exist, (ii) uncontrolled inputs typically coming from inherent material and process variability, (iii) live measured sensor data if it exists and (iv) material and physical model constants used for numerical solution
2. Quality prediction framework provides a real-time response regarding the outcome of the process. Key technical challenge here is to provide a real-time (<10 sec) response to any enquiry.
3. A database containing details of PDT usage and capable of receiving actual quality results
4. Models and algorithms related to Feature extraction, quality prediction and machine learning

In section 3, we will present a generic software architecture and design to build the framework in most modular, configurable and broadly applicable to many processes.

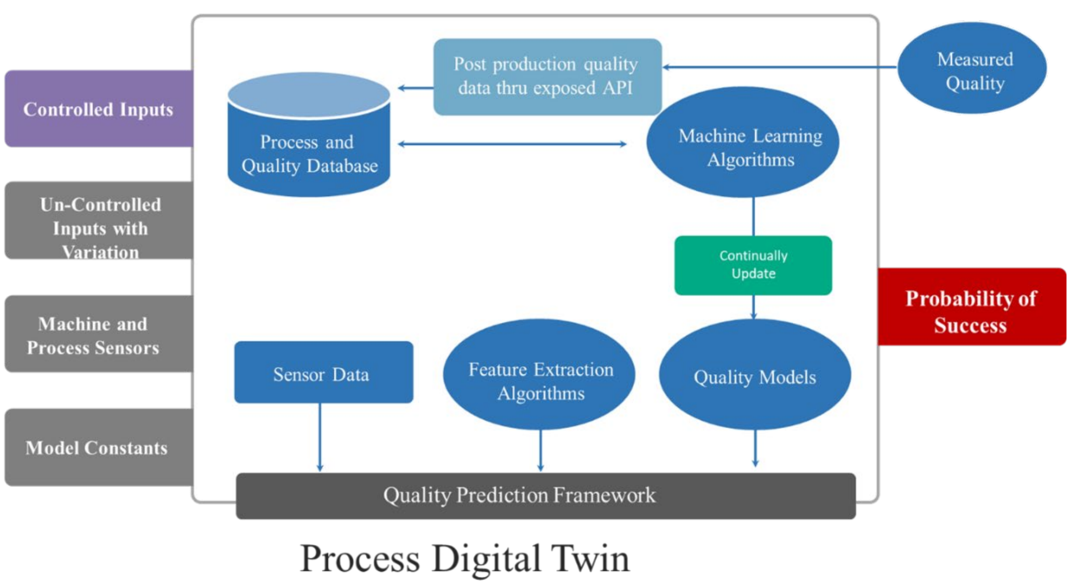

Figure 2: Key elements of a Process Digital Twin for a high performance manufacturing process

2.2 Quality Prediction Framework

Based on our requirements we have designed a generic framework which is highly configurable, modular and maintainable during manufacturing process. Figure 3 shows the base elements of the framework and transformation methods/logic operating between elements. Primary objective of the framework is to incorporate entire manufacturing process from inputs to final quality. Figure 3 shows the elements of this framework and details of each elements are described below:

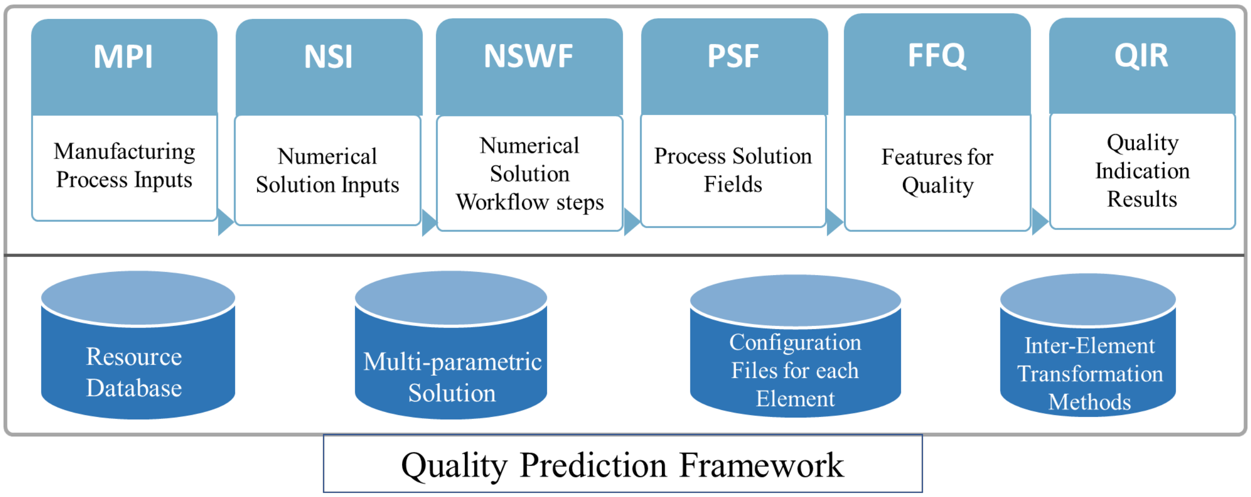

Figure 3: Key Elements of Quality Prediction Framework. Data flows from Left to right and inter-element transformation methods/logic are used to insert custom process knowledge.

Manufacturing Process Inputs (MPI) – Includes controllable, material, model and quality inputs. All inputs here are as operator or users see on manufacturing floor. MPI to NSI transformation methods are used to translate into physical quantities for next step. An example would be translating a particular autoclave location into a heat transfer coefficient.

Numerical Solution Inputs (NSI) – Includes physical quantities that are boundary conditions, initial conditions and other parameters required for building a numerical solution.

Numerical Solution Workflow (NSWF) – This element defines workflow steps that will be used for creating a multi-parametric solution – a key to providing real-time (<10 sec) solution. We will discuss details of latest numerical methodology in separate section.

Process Solution Fields (PSF) – Full field solution for individual part (based on specific inputs for the part) are then generated using the multi-parametric solution.

Features for Quality (FFQ) – Based on the process understanding and perhaps past knowledge, the solutions fields and all the other inputs are translated into a set of features that are deemed important to predict quality. This may require fairly complex algorithms to extract these feature from field solutions and potential sensor measurements

Quality Indication Results (QIR) – Here we capture key quality failure modes as indications and predict the probability of success based on quality requirements. Requirements can be in form of hard limits or heuristic logic involving one or many FFQ's. Overall probability of success is then predicted as a product of all individual indication success.
An example for adhesive bond cure process is presented in Ref [2] providing details for all inputs and variable.

2.3 Real-Time Solution Methodology

Key challenge for building a workable PDT is to build real-time simulation engines that can be seamlessly integrated in manufacturing environment. To address this challenge, computer simulation is required to meet two constraints: first, the accuracy on the predictions; and second, the speed of simulation. It is a challenge to satisy both constraints simultaneously, especially when dealing with high-performance products that require complex, high value-added and high-variability manufacturing processes. These advanced solvers are not available "off the shelf" but must be specifically trained for the problem at hand. The mathematical methods used for training are in the class of model order reduction (MOR) techniques. By analyzing a set of simulated data, these algorithms learn a reduced set of intrinsic features to represent the system in a more compact way. The governing equations are thus reformulated assuming the learned features as model variables. In this way the original complexity of the system is overcome, making the resulting algorithms run with impressive speed-ups. The methodology to create a real-time simulation engine has three main steps (i) full-order model development, (ii) model parametrization and (iii) real-time simulator training.

2.3.1 Full-order model development

A first step to build real-time simulation engines is to have a full-order model that is able to predict accurately, although not in real-time, the manufacturing process field. Full-order model must capture physics models and their couplings; constitutive relations and material models; and external interactions such as constraints, loads or contacts. In continuum mechanics, for instance, models are expressed mathematically as partial differential equations (PDE) describing conservation laws governing the evolution of the fundamental variables that describe the state of a system.
From a mathematical perspective, the solutions of physics models are seldom found in closed analytical form. In general, they must be approximated using numerical discretization techniques. Although some simplifications are possible for particular cases, the geometrical complexity may require a full three-dimensional treatment. Nonlinear physical couplings are also a critical factor for the numerical approximation of the problem solution. Indeed, the numerical schemes are required to enforce the coupling at the discrete level in order to correctly capture the essential physics features.

The specific technology to be used for the solution of the full-order model depends strongly on its nature. Finite Element (FE) analysis [ref 3, 4 and 5] is one of the most popular and robust technologies, but many others exist. Details on the specific implementations of the full-order model will be given in Section 3.

2.3.2 Model Parametrization

To achieve real-time performance without trading off accuracy, it is essential to appropriately parametrize the full-order model. Parametrization is a fundamental way to account for both the model uncertainty and the variability of operating conditions. This includes identifying prior probability density functions for the non-deterministic parameters and defining admissible values or ranges for decision/design variables.

The primary goal of model parametrization is define the space of possible operating conditions in which we want the simulation engine to run. Then, the speed of simulation of the full-order model performance will be optimized to achieve real-time, without simplifying the underlying physics nor sacrificing significantly the numerical accuracy.

2.3.3 Training a real-time simulation engine

To achieve real-time performance of the simulation engine, following steps are carried out:

Lower dimensional representation of Process Solution Fields (PSF): This step is achieved through a dimensionality reduction technique such as the Principal Component Analysis (PCA) [6], the Proper Orthogonal Decomposition (POD) [7]. These techniques discover the intrinsic dimensionality of the model which is expected to be considerably lower than the numerical resolution of the full order FE method that is used to discretize the governing equations.

Construction of reduced order model: This is done by reformulating the governing equation in terms of the reduced variables inferred from the previous analysis. This requires a formal definition of a change of variables from full to reduced order subspaces. Note that the new basis functions spanning the reduced order subspace are also obtained as a part of the previous step. The reduced order variables need to satisfy a reduced set of conditions that can be obtained from the original governing equations. Two possible approaches to derive the reduced order equations are:

(i) *Galerkin projection:* requiring the orthogonality of the full order residuals to a lower dimensional subspace, which might or might not coincide with the reduced order subspace where the solutions are assumed to live (in which case the method is of the Petrov-Galerkin class). In this family of methods belong the Reduced Basis method (RB) [8], the Galerkin-POD and the Proper Generalized Decomposition (PGD) [9, 10, 11 and 12].

(ii) *Residual minimization*: requiring the minimization of the full order residual in a given norm with respect to the reduced order variables. Under the precise mathematical hypothesis of affine structure of the problem, it can be shown that pre-processing, solving and post-processing the new set of equations can be performed in reduced order complexity. Indeed, all these operations do not depend on the original complexity of the full order model that is used to generate the data. Therefore, a considerable speedup can be achieved allowing the resulting simulation engine to be operated interactively.

Establish Model Accuracy: Accuracy of the full order model can be evaluated through formal error estimators. In most cases, a natural choice is to base error estimation on the residual norm.

The residual-based methods build sharp estimators for which the residual norm can also be evaluated in reduced complexity. These are used to estimate the expected error while operating the simulation engine, but also offline to decide whether the simulation engine meets the desired accuracy or the model must be updated using new data from the full order solver. Therefore, the error estimator can be used as an inexpensive indicator in order to retrieve the most information from the smallest possible re-sampling of the parametric space.

In the residual-based framework the new sampling point is chosen as the new combination of parameters that maximizes the error estimator. As an alternative, if all the parameters are independent, the parametric space has a tensor structure that can be exploited to compute integral estimators over the whole parametric space. This idea is leveraged by the PGD. In both cases the model update is performed very efficiently. Every time a new full order solution is computed, it is used to enrich the low order subspace and update both the reduced order model and the error estimator. This process is iterated until the measure of the error becomes lower than a fixed tolerance. Nonlinear terms require a specific treatment, since they can invalidate the hypothesis on the affine structure of the problem which is fundamental for fast online execution of the simulation engine. In this work, we followed the hyper-reduction approach [12], consisting in learning a subset of quadrature points and weights that can be used to evaluate.

Despite the fact that the learned features lack a precise physical meaning their relations to physical quantities of interest is formally defined by simple mathematical expressions. Relevant information is therefore accessible in short time and using low computing capabilities.

2.4 Machine Learning

Machine learning algorithms that build quality models are an integral part of the Framework. Quality models that relate FFQ's to Final result are initially built on experience and past data but machine learning process can continually improve predictive confidence. Process Digital Twin can also use the pre-trained simulator to expand the feature space by using not only real data but also simulated data. The simulator can be seen as a virtual sensor able to provide readings of variables for which sensors do not exist, at locations that cannot be accessed. This is important in the early stage of process development where very little production data exists. In general FFQ data is noisy, we can only develop quality models in a probabilistic framework. We have used some of the methods described in references [13] and [14].

2.5 Process Digital Twin Applications

PDT can be integrated into design to manufacturing lifetime of a product - part design, process development, manufacturing scale-up and autonomous process control. The richness of data content and predictive confidence levels will steadily increase through each stage. Let us take a closer look at each of the stages.

In part design phase, multiple processes may be considered for production and key objective is to choose a manufacturable design. The primary knowledge in PDT will be physical models of prospective processes and some generic criteria to evaluate quality.

In process development phase, the part design and process to manufacture has been decided and primary objective is to develop initial process window. In this case PDT would need best possible physical models based on best understanding of the process. Key experiments and process tests and sensor data would augment the knowledge and support understanding the failure modes to develop quality correlations.

In manufacturing scale-up phase, primary objective is to have high production yields. This requires adapting for inherent material and process variations typical of high performance and value-added processes. Sensor and other process data is integrated and machine learning algorithms can be used to continuously improve quality relationships to features for Quality.

In autonomous process control phase, primary objective is to automate adaptation thru machine controller as much as possible. In general terms, the feedback control loop is based on predictive quality from DPT. Real time response from DPT is the most critical factor in this phase.

3. SOFTWARE ARCHITECTURE AND DESIGN

For broadest applicability to any process, we have developed a software architecture shown in Figure 4 with design considerations shown in Table 1.

API first Approach	All internal components and external systems interact using API's
Scalable	Both hardware and software scalable. Multi-instance deployment to handle load. Multi-tenant deployment
Configurable	Parameter types encapsulating ranges, condition and other needed properties. Each element configured for process specific methods, chart/visualization Expandable to any number of elements
Easy Integration with User's manufacturing environment	Integration to receive the manufacturing process inputs and return the accept and reject back to the system requesting the information
Open Source Methodology	Open source technologies to limit the cost of ownership
Cloud Ready	Design needs to be cloud ready
Fast and Responsive (<10 sec)	State of the art technologies to give end user a fast and fluid response

Table 1: Key design considerations for the architecture of PDT application development

Digital web application includes the application web interface and set of API endpoints to interact with the system process controller. The web application offers the user a web interface to configure and run workflow executions and perform set of available visualizations for an execution of the selected workflow.

Digital twin process controller is the main core of the application, which is responsible for managing, configuring and executing workflows while interacting with the available data models. This component consists of the below defined sub-modules

Workflow template configuration manager: This component allows the user to create and manage workflows. It gives options to setup new workflow definitions using the defined json file structure or using the web application configuration screens. The user can follow the defined structure to set up new groups, sub groups and parameters for a workflow. The generic structure which is used gives user complete flexibility to represent complex parameter structure just as vectors very easily into the application. The user can easily add new parameters if needed and can easily adapt to the changes in the manufacturing process. User can set up workflow inputs for all the 6 steps (MPI, NSI, NSWF, PSF, FFQ, QIR) independently.
Workflow execution manager: This module is the core for executing the configuration for all the 6 steps defined. During the execution of the workflow the execution manager also leverages expression evaluator for any commutation configured for each step. Once the workflow is completed the final result and also for each step is pushed back.

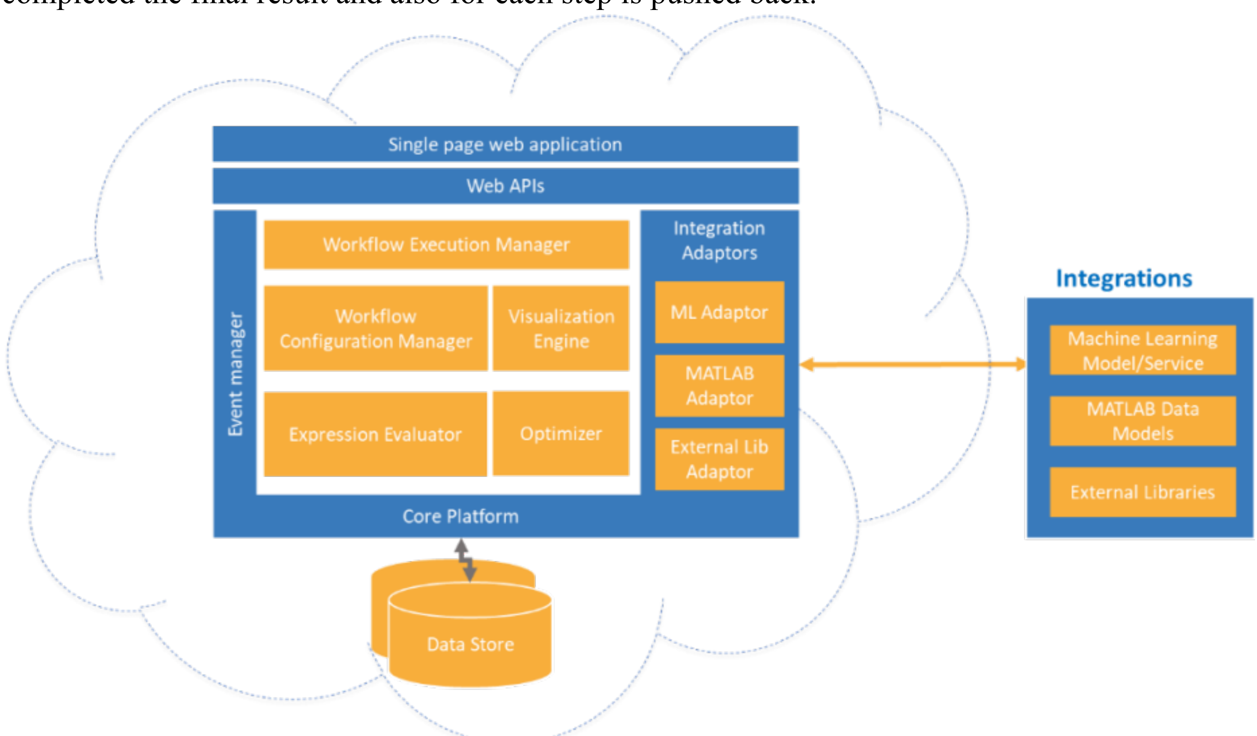

Figure 4: Process Digital Twin software architecture

Expression evaluator: The expression evaluator module is responsible for resolving expression during a workflow execution. The expression evaluator also interacts with custom methods defined by the users and calls the method during the execution. The user can use basic operators like addition, multiplication, subtraction and division directly with in the expression. For more complex operations, user always has an option available to create a custom method and upload it to the application using the web interface. The custom method is registered and then can be directly use with in an expression.

Visualization Engine: Once the workflow is executed, the user can run a set of defined visualization on the generated data. The visualization engine interacts with the data model through the adapter to fetch the required data. The system currently allows three types of visualizations: Line Charts, Mesh Charts and Mesh charts over time. User can define multiple visualization charts while configuring the workflow definition.

Configuration Optimizer: Configuration optimizer is a new component which is being developed to optimize the workflow configuration settings for adaptive manufacturing. The PDT will interact with an external Machine Learning (ML) system which includes a set of models which are executed on actual and predicted results. The configuration optimizer will use the result of the ML system to update the values of quality model parameters after passing a set of validations.

To Integrate PDT application to users environment, we have developed a general framework DiVAM (Digital Value Added Manufacturing) shown in Figure 5.

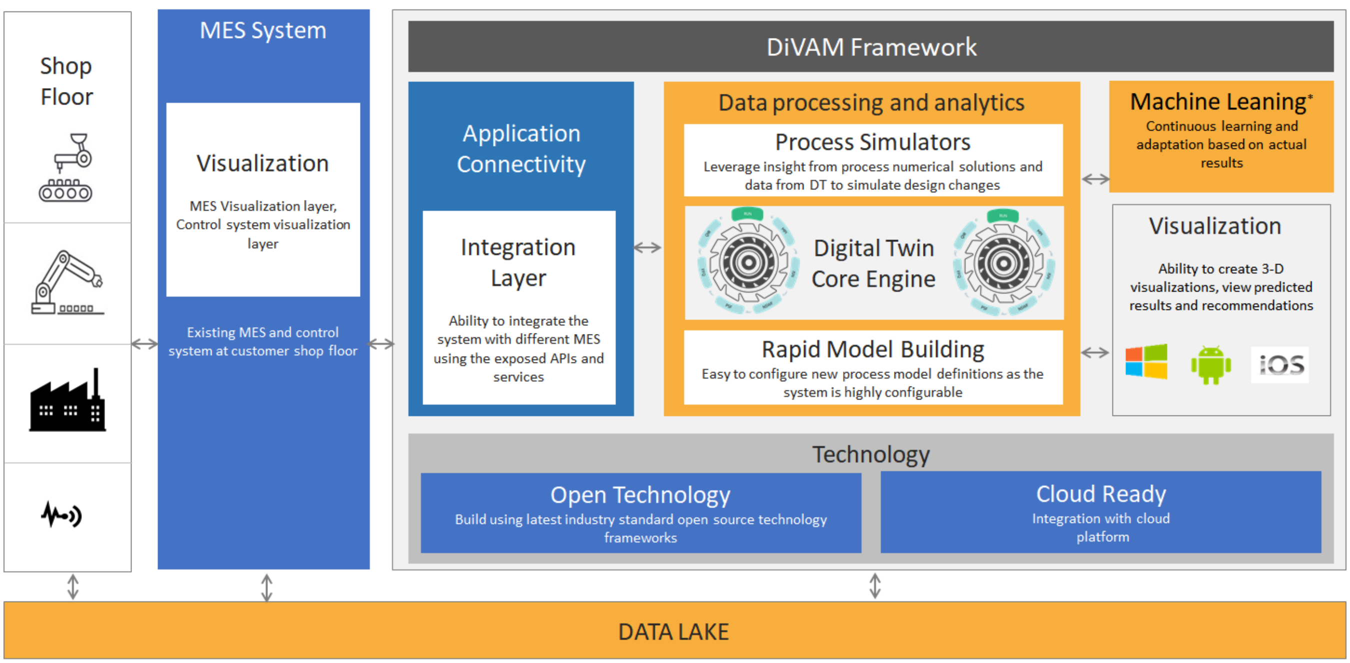

Figure 5: Framework for Digital Twin Integration into a manufacturing environment

4. IMPLEMENTATION EXAMPLE

We have applied the PDT technology to an adhesive bonding process for a premium composite fan blade manufacturer and Lednicky, Upadhyay and Baer[2] provide full implementation details in a separate paper in the current CAMX 2019 conference. Figure 6 shows the implementation of an optimizer that maximizes the yield of a batch of parts using process digital twin as a predictive function by adjusting controllable inputs.

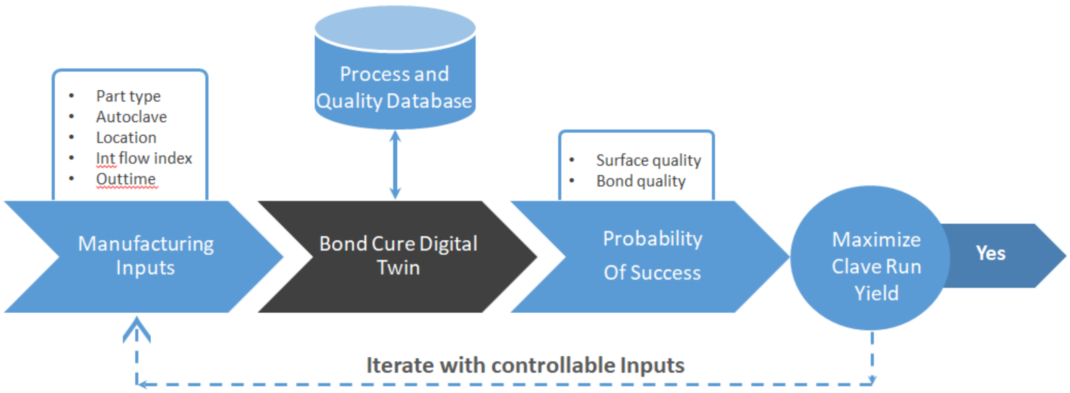

Figure 6: Example of implementation of process digital twin to adaptively handle incoming adhesive flow variation.

5. CONCLUSIONS

Process Digital Twin technology is changing the manufacturing paradigm from reactively using data to pro-actively managing production quality. With proper specifications and definition of key elements, PDT will seamlessly blend into product design to manufacturing life cycle. Since it cumulatively adds to the process knowledge thru initial part/process design to automated manufacturing, it accelerates new part introduction and help avoids the "valley of death" very common while scaling up processes to full production. Within next decade, every high value-added manufacturing process would not be able to compete without Process Digital Twin technology. We are currently going thru a paradigm shift thru introduction of PDT technology.

Our team (authors of this paper) collaborated to build a generic modular framework and software architecture to build PDT that can be customized for any individual process. Also, a rigorous mathematical background has been developed for accuracy assessment and error estimation of real-time numerical solutions. Working with our customer CFAN, PDT technology has been successfully applied to real-time manufacturing processes.

6. ACKNOWLEDGMENTS

We acknowledge support of CFAN to apply the PDT framework for bond cure and blade cure process to successfully demonstrate PDT technology in manufacturing environment.

7. REFERENCES

1. Upadhyay, R.K. and Sinha, S., "3.6 GE-90 and Derivative Fan Blade Manufacturing Design", *Comprehensive Composite Materials II*, Volume 3 doi:10.1016/B978-0-12-803581-8.10075-X180, Elsevier Ltd. , 2018
2. Lednicky, T., Upadhyay, R.K., and Baer, M., "Paradigm Shift in Composite Fan Blade Manufacturing using Process Digital Twins", paper to be presented at SAMPE 2020 conference, Seattle, 2020
3. Bathe, K. J. (2006. *Finite Element Procedures*. Civil engineering series. Prentice-Hall, Englewood Cliffs, NJ., 2006
4. Zienkiewicz, O. C. and Taylor, R. L., *The finite element method. Vol. 1 The basis*. Butterworth Heinemann, Oxford, 5th edition, 2000
5. Zienkiewicz, O. C. and Taylor, R. L., *The finite element method. Vol. 2 Solid mechanics*. Butterworth Heinemann, Oxford, 5th edition, 2000
6. Jolliffe, I., *Principal Component Analysis. Springer Series in Statistics*. Springer-Verlag, New York, 2nd edition., 2002
7. Białecki, R., Kassab, A., and Fic, A., "Proper Orthogonal Decomposition and modal analysis for acceleration of transient FEM thermal analysis". *Int. J. Numer. Meth. Engng.*, 62(6):774–797, 2005
8. Quarteroni, A., Manzoni, A., Negri, F.: *Reduced Basis Methods for Partial Differential Equations: An Introduction, 1st edn. Modeling and Simulation in Science, Engineering and Technology*. Springer, Basel, 2015. https://doi.org/10.1007/978-3-319-15431-2
9. Ghnatios, C., Masson, F., Huerta, A., Leygue, A., Cueto, E., Chinesta, F.: "Proper generalized decomposition based dynamic data-driven control of thermal processes". *Comput. Methods Appl. Mech. Eng.* 213–216, 29–41, 2012
10. Borzacchiello, D., Aguado, J., Chinesta, F.: "Reduced order modelling for efficient optimization of a hot-wall chemical vapour deposition reactor". *Int. J. Numer. Method Heat Fluid Flow* 27(4), 1602–1622, 2017. https://doi.org/10.1108/HFF-04-2016-0153
11. Aguado, J., Borzacchiello, D., Ghnatios, C., Lebel, F., Upadhyay, R., Binetruy, C., Chinesta, F.: "A Simulation App based on reduced order modeling for manufacturing optimization of composite outlet guide vanes". *Adv. Model. Simul. Eng. Sci.* 4(1), 1–26, 2017. https://doi.org/10.1186/s40323-017-0087-y
12. Hernández, J., Caicedo, M., Ferrer, A.: "Dimensional hyper-reduction of nonlinear finite element models via empirical cubature". *Comput. Methods Appl. Mech. Eng.* 313, 687–722, 2017. https://doi.org/10. 1016/j.cma.2016.10.022
13. Gamerman, D., and Lopes, H. F.: *Markov chain Monte Carlo: stochastic simulation for Bayesian inference*. Chapman and Hall/CRC, 2006.

14. Cheeseman, P.C., and Stutz, J.C.: "Bayesian classification (AutoClass): theory and results. Advances in knowledge discovery and data mining" 180, 153-180, 1996.