Proceedings of the ASME 2018
International Mechanical Engineering Congress and Exposition
IMECE2018
November 9-15, 2018, Pittsburgh, PA, USA

IMECE2018-87361

DEMONSTRATION OF AN INDUSTRIAL FRAMEWORK FOR AN IMPLEMENTATION OF A PROCESS DIGITAL TWIN

Mr Jonathan M. Eyre
Advanced Manufacturing Research Centre
Sheffield, UK

Professor Tony J. Dodd
University of Sheffield
Sheffield, UK

Mr Chris Freeman
Advanced Manufacturing Research Centre
Sheffield, UK

Mr Richard Lanyon-Hogg
Advanced Manufacturing Research Centre
Sheffield, UK

Dr Aiden J. Lockwood
Advanced Manufacturing Research Centre
Sheffield, UK

Professor Rab W. Scott
Advanced Manufacturing Research Centre
Sheffield, UK

ABSTRACT

Digital twins have received a large amount of exposure around what they can offer to industry generating lots of noise, however there are few demonstrations utilizing published architectural frameworks. This has been addressed by investigating industrial publications and reports on what is the minimum essential requirements to form a digital twin and additional desirable features. From this, a generic industrial architectural framework of a digital twin has been established to utilize real-time information from a physical asset forming a monitoring digital twin. This has been expanded to incorporate a Discrete Event Simulation (DES) to form a process digital twin utilizing structured information about the process. The framework, including the DES extension, has been validated on a reconfigurable fixture utilizing an established process that has been modelled using Siemens Plant Simulation. This result forms the start of a feedback loop presenting additional value transforming a monitoring digital twin into a process digital twin. This provides a solid foundation for discussion within the industrial community about defining the core functionality required for digital twins.

INTRODUCTION

High Value Manufacturing (HVM) sits within a wider theme called Industry 4.0, also known as the fourth industrial revolution, which is the name associated with the digitalization of manufacturing industry requiring the incorporation of new technology and infra-structure. One of these is the Internet of Things (IoT) to have a network of connected manufacturing cells or nodes that exchange information without human intervention, creating a smart factory utilizing a Cyber Physical System (CPS). Part of this is having 'big data' and 'big analytics' to 'data mine' for trends reducing production costs and increasing efficiency adding value to the manufacturer. One of the central themes within visualization is monitoring systems using real time data, of which one core area is a digital twin.

Digital twin is a term originally defined by Dr. Michael Grieves' in 2003 at the University of Michigan [1], but has gained significance worldwide both within industry and academia for the virtualization of the factories. Fundamentally, a digital twin is a mimic of a real world asset displaying up to date information of what is currently happening. This notion can be extended to incorporate a vast amount of functionality, however this article focuses on the basic principle due to the complexity involved with the current functionality suggested of a fully active system. The principles can be split into three core aspects: the physical asset, the virtual counterpart and the data link between them. This article presents this by discussing in turn: the creation of a universal framework for the implementation of digital twins as a framework is not currently agreed upon, creating the visual graphics and integration of the live sensor data. These sections alone provide a monitoring digital twin, but this has been extended with the formation of a feedback loop utilizing DES to compare nominal process timing information in real-time providing extra feedback information to the user.

Digital twins are non-trivial to produce due to the intricate process and wide range of skills required, which at a high-level view requires visualization and data communications with awareness of robotic manufacturing systems for this application. In addition, conventional Computer Aided Design (CAD) software packages are unsuitable for real-time viewing of an asset due to functionality to accept a live data feed not being incorporated into this type of software. Due to this, real-time visualization software is required that has the capability of displaying content while incorporating a communication connection to external data sources. One range of software that fits these requirements is real-time engines, which are conventionally for creating games and deploying to a wide range of platforms. Using a real-time engine does cause some complications such as three-dimensional (3D) model data is required to be a triangular mesh, also known as a polygon mesh, rather than a CAD data format. Although polygon mesh formats can be produced from CAD software, these frequently contain too much information for real-time engines to handle. Hence an evaluation of this conversion process will be reviewed due to being a key issue of choosing to use real-time engine technology.

In general, this article presents the current conclusion of the literature and an implementation of a process digital twin for a single reconfigurable robotic cell. This provides not only real-time state information for a monitoring digital twin, but also additional simulated process information closing the real-time feedback loop of information to the user.

LITERATURE REVIEW

Manufacturing funds £6.7tn to the global economy and is a key part of the UK economy making up 10% of Gross Value Added, 45% of UK exports and is directly employing 2.7 million people in 2017 [2]. In addition, investments from UK manufacturing are estimated to be 68% of the total Research and Development (R&D) of the UK economy, generating innovative products and manufacturing techniques [3]. This chapter introduces the topic of advanced visualizations within this investment towards Industry 4.0, of which this paper focuses on the digital twin. Existing industrial research is reviewed to: establish the functionality required from a digital twin, the value of a digital twin and the core functionality required that will ultimately be developed for the prototype.

BACKGROUND

Manufacturing is investing into new technologies, of which achieving the desirable end-to-end product lifecycle with complete lifespan management for the Industry 4.0 vision requires a range of integrations for both software and hardware. These integrations will require gradual incorporation over several years to utilize current technology such as IoT and disruptive technology such as mass commercialization of immersive technologies, for example virtual reality (VR).

Integrations are hence an important step towards Industry 4.0 including the gathering and analysis of data using communication channels possible through IoT to enhance the overall efficiency of producing products and providing services. Sensor data capture is just one of many integrations necessary for a smart factory that alone allows monitoring of the current state. Another aspect would be integrations into the business data architecture such as product lifecycle management (PLM) and enterprise resource planning (ERP) to have data flow from concepts through to end of life to maximize data usage throughout the life of all products. ERP integration would enable optimization of resources, such as maintenance records of when machines need to be serviced and logistic workforce planning. The Annual Manufacturing Report [4] highlighted that 42% of companies are making their most significant investment into ERP, with the second largest of 16% into hardware and 13% into Customer Relationship Management software. Each of these have current value to industry, however they become more effective once incorporated together to influence from conception through to the end of life.

All of the integration infrastructure required for big data and big analytics have no value if a company cannot interact with the data collected of their CPS, which is where the idea of the digital twin is formed. The concept was formed in 2003 by Dr. Michael Grieves as an extension to a PLM system [1]. Dr. Grieves discussed in this report that if enough information could be collected about a physical product as it was being produced from non-destructive sensing technologies then a physical product could be shown in a virtual space. This physical representation is not limited to the current position of parts, but also the: status, motion, properties and states of components in the system. However, this report did not discuss real-time data that is now expressed as a core feature of a CPS.

Siemens are one of the leading companies within this field stating “With a digital twin … use the power of digitalization to achieve improved efficiency and quality”. They continue to present several key themes that a digital twin would require such as: factory layout planning, validation of designs and testing configuration of machines through simulation. These themes are common across several publications or statements by several companies. GE define a digital twin as “a digital replica of any industrial asset … that is used to monitor, analyze and improve its performance” that “continuously collects sensor data on the asset and applies advanced analytics and self-learning artificial intelligence to gain unique insights about its performance and operation” [5]. This allows an insight into the inner workings of what a digital twin must achieve using analytics and artificial intelligence to facilitate the improvement of performance for manufactured goods or industrial assets.

IBM more recently provided a very similar definition of “a virtual/digital representation of a physical entity or system” [6]. The article also discusses the connected “things” generate real-time data that is analyzed in the cloud utilizing contextual data such as the environment to present suitable data to a variety of roles. The presentation of data is a key feature of a digital twin. With high potential for overwhelming amounts of data being produced, only relevant information should be displayed to users meaning each user, or in a broader term type of user, has a unique experience of using the digital twin software. For example, management may require remote understanding of its current production rates and issues affecting production, whereas shop floor staff and technical maintenance would be more interested in the physical assets maintenance history, needs and interactions.

Visualization of the mentioned information is non-trivial due to high variety in the way people digest information, even after reducing the amount each specific user needs access to. Dashboards, such as human machine interfaces (HMIs), are already common within industry presenting a series of contextualized labels and states either via stated names or a graphic representation of the physical counterpart. VR is another visualization platform recently receiving interest that provides a fully immersive 3D experience where the graphics can update from the same live connection driving dashboards. Although very different approaches, both of these meet the definition of a digital twin. However different user groups prefer different platforms, which these just being two examples escalating the high variance in people's perceptions of digital twins.

This overarching vision is achievable by a system comprising of three main components, regardless of the chosen platform(s), which have been identified by IBM through their digital twin with Watson IoT; the physical "thing", the virtual representation through the digital twin and the information link between the two allowing the real-time data connection [7]. The key aspect being the link to allow the feedback for all stages of development from early in the design analysis to optimize the proposal through to the end of life production to enhance the manufacturing process or operation with potential for improvements in the future iterations of the design.

THE VALUE OF A DIGITAL TWIN

Although the digital twin concept is possible due to utilizing technologies such as IoT, the system must present industrial significance and value for investments to be made into it. IBM presented in March 2017 their vision of the digital twin producing three key areas of value: design, build and operation [8]. During the design phase, the digitalization of the product can allow rapid development cycles for alternative proposals utilizing current products and regulatory requirements to find the optimal design. When considered within the setting of Industry 4.0, designs can be visualized in context using Augment Reality or within fully immersive environments using VR providing extra value to the designers. Visualization of the build process can hence provide better efficiency and quality yielding better insights causing reductions in costs and scheduling. This aspect is only possible through the integrations of the management tools such as PLM and ERP. Operation improvements covers service operation incorporating demands of uptime, safety and efficiency of resources with staff time and hardware such as manufacturing cell usage. For example, implementation of prescriptive maintenance for serviceable parts rather than reactive by utilizing machine learning.

Another focus is the overall impact a digital twin has to industry, offering value through having different viewpoints available across the whole lifecycle of a product, deemed as the Digital Thread by IBM. However, having access to all information requires manipulation of the data sets to tailor specific viewports to suit different stakeholders.

While the previous information presented was IBMs stance of where value lies, this is supported by Siemens by their statement of "computer models that provide the means to design, validate and optimize a part, a product, a manufacturing process or a production facility in the virtual world" [9] displaying what features they see adds value. They continued to state their digital twin vision should "simulate, validate and optimize robotic operations before they were executed on the shop floor" continuing to discuss virtual commissioning of cells before they are even built highlighting the same three aspects that IBM explicitly stated.

Overall, a wide range of functionality could be incorporated, however the important factor is while there is no quantifiable amount of value digital twins will generate for industry, there is clear significance to the research through such functionality as: monitoring, optimization, prognostics and diagnostics. The information presented all indicate monitoring forms the core of a digital twin, requiring: the physical asset, virtual representation and data link between them. Without observation, other functionality either becomes infeasible or meaningless, therefore this was identified as core.

For the benefit of this paper and the prototype developed, a digital twin is defined to be a *virtual mimic of a physical asset utilizing real-time data*. Although other features in addition to monitoring should be incorporated into a fully developed digital twin, monitoring applications form the base which a process digital twin builds upon. The real-time aspect is achieved by utilizing sensor data outputting the information to the digital twin through a server driven data link, i.e. the digital thread.

METHODOLOGY

First, electronic journals were searched for the keywords 'digital twin' most prominently with secondary searches made for wider understanding with keywords 'cyber-physical systems', 'industry 4.0' and 'industrial internet of things'. Information deemed relevant were extracted in the literature review and presented within this article to summarize the key characteristics of a digital twin at the time of writing. Secondly, a universal system architecture was created to formalize what the system must do and how using the universal modelling language (UML) notation. Although this project focused on a single flexible fixture, the architecture presented aims to be universal due to no established framework being agreed upon.

Next, the creation of the visual aesthetics was split into three sections. Conversion of the CAD data into suitable polygon meshes was undertaken first. To do this, the provided CAD files (.stp) were converted to an initial polygon mesh using Ansys SpaceClaim [10] to then be optimized to reduce the processing requirement and hence hardware requirements of the digital twin prototype. Secondly, the visual look of the objects were formed to define how to visualize them. Finally, moveable objects from the state information were grouped into rigged components to prepare them for the real-time visualization. All of these steps were completed within software called Blender 3D [11]. This is a 3D creation suite that supports a full 3D pipeline, including: modelling, material creation, rigging and animation. Although automated tools do exist for optimization of the object representation, manual optimization still currently yields the best results.

Connection of live data from the physical flexible assembly cell to the digital thread was achieved with an industrial computer (the data collector) acting as a bridge between the local cell network and the main shop floor network. This is running an instance of Node-RED that is a flow-based programming tool to rapidly create software with the editor accessible via a web browser [12]. The digital thread for this application utilizes: message queuing telemetry transport (MQTT), a SQL database with a front-facing RESTful application programmers interface (API) and Kepware (from PTC), which is their industrial connectivity platform.

The last development was the creation of the digital twin software to mimic what is happening in the real world with the physical asset utilizing the real data link established over the digital thread. This will be created within a real-time engine, of which Unity 3D [13] was selected due to familiarity and is programmed using the C# language. Unity is a cross-platform real-time engine that has a development environment for creating 2D or 3D environments with menus, animations and custom functionality through scripts to then be deployed to a wide range of platforms. A real-time engine was chosen due to CAD software being unable to be used for live links and the core aspect being real-time rendering. In addition, Unity allows the software to be developed for an immersive virtual reality (VR) world that was chosen for this project.

SYSTEM ARCHITECTURE

The presented system architecture is defined using UML to explain the dynamics of the system and a structural diagram to explain the hierarchy of the implementation. The first diagram establishes what the system will be used for through a Use Case diagram (Figure 1). This displays the first agent, the user, who has three main interactions for the current implementation of a digital twin. These are: launching a static view of the world, live update from a data feed and feedback from a live simulation comparing nominal data to the current process. The digital twin sub-system requires access to two of the three data sources hosted on the server stack. These are a SQL database, MQTT message broker and PTC's IoT connectivity platform, Kepware.

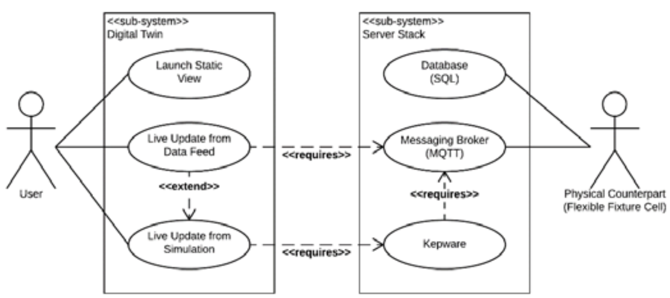

Figure 1: Use Case diagram displaying the user and physical counterpart agents interacting with the use cases within the Digital Twin and Server Stack.

The second behavioral diagram, shown in Annex A, presents how the systems will interact and in what order following a time sequence from top to bottom, known as a sequence diagram. Interactions are described using high level functions in the specified sequence that continue with each object running in parallel to allow communication. The sequence diagram states how the digital twin will be updated using the live data feed from the server stack whilst running a simulation concurrently to receive information to augment the state information from the production cell.

The final diagram defines the structure of the information flow between the systems (Figure 2). Information starts from the Programmable Logic Controller (PLC) of the flexible assembly cell into the data collector. This is setup on an industrial pc with two Ethernet ports to bridge between the local cell network and the shop floor network to push data onto the digital thread.

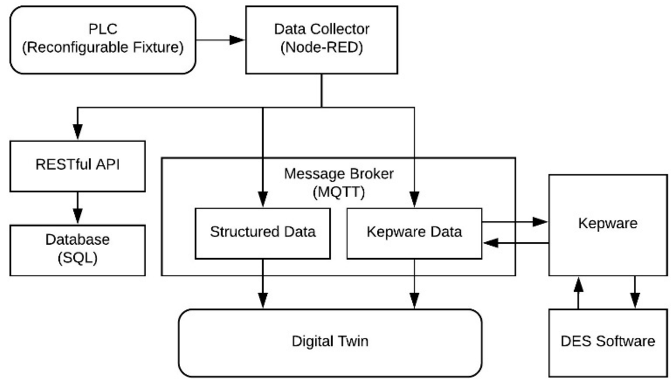

Figure 2: The developed information flow between the systems.

The data is published from the PLC as a structured byte array that is parsed and pushed to both a SQL database via a RESTful API and MQTT message broker using Node-RED. The data collector filters the incoming data feed to ensure only new data changes are pushed to the server stack minimizing data traffic. Although this is insignificant for this project, this would become substantial for a whole factory adopting this architecture. The digital twin could receive data from either the database or message broker as they contain the same data, however implementing SQL would cause unnecessary data flow if no movements or state changes are being made due to the polling required to check for changes. By utilizing MQTT, once a connection is made to the messaging broker stating which channels a subscription would like to be made to, the digital twin only receives traffic when new data is available. This means the MQTT broker is the most appropriate for live updates of the digital twin for general use. However, MQTT is unsuitable for historical viewing as it only stores the last message, which is why data is still stored in a database for historical collection. Although out of scope for this project, historical record data analysis is another theme that sits within Industry 4.0 which is now possible from this implementation.

Excluding the PLC data being sent to the data collector, all information exchanged utilizes the JavaScript object notation (JSON) structure forming the commands the digital twin will use. This was chosen due to being a lightweight data-interchange text format that is a structured and language independent utilizing conventions such as brackets to break up key/value pairs and arrays structures. The platform agnostic structure allows the information to be used by the many systems across the project and is highly scalable if implemented across a full architectural solution. The following JSON message is an example that contains state information relating the cell declaring the door is open and the emergency stop is activated at this point in time. This message structure is generated on the data-collector upon parsing using the known data structure coming from the PLC to align the data set to what they mean.

{"doorClosedState":0,"emergencyStopActivated":1}

IMPLEMENTATION

From the developed system architecture, each section was developed in turn before integration of the digital twin prototype. This chapter discusses: how the virtual representation of the cell was created; the live data connection; the parallel running simulation and the digital twin application. The definition of a digital twin stated previously does not require a 3D visual, however this project chose to investigate how structured 3D model information could be used in the production of an immersive digital twin in addition to closing the loop through DES. For this reason and to achieve the immersive visual, VR was selected as the final platform to be developed of the production cell that the user can navigate around.

VIRTUAL REPRESENTATION

CAD data is stored in precise formats traditionally through a combination of precise geometry and boundary topologies to represent 3D objects. This precision comes from defining each face of the modelled object surface by an equation where edges are then formed where two equations meet allowing querying of the model to any precision. The methodology for this project utilizes a real-time engine, which require polygon mesh models where all surfaces are split into triangles stored as vertex, edge and face information. The conversion process from CAD to a tessellated model is non-trivial. For example, a conversion of a cube can be expressed with a series of six plane equations and equally expressed with a series of eight vertices, 12 edges and 12 triangular faces with no loss of information. However, a more complex example is a single sphere. With equations, the sphere can be described stating the center as a 3D co-ordinate and radius of the model. This cannot be precisely described as a tessellated object as this would require an infinite number of points and hence impossible to hold in computer memory. Hence, model precision is almost always lost when a conversion from CAD into a tessellated model takes place. The difficulty then turns into choosing enough vertices to keep enough detail for the application, but not too many to overload the target platform hardware.

A heuristic approach to produce tessellated meshes is required as there exists no golden universal rule set to produce suitable polygon meshes for all object types. A base tessellated model was created by Ansys SpaceClaim due to previous experience with the software and its capability to import the provided .stp file of the reconfigurable cell and export a mesh type, of which .obj was used. Even by exporting the polygon mesh at the lowest quality settings using a distance deviation of 2mm and angle deviation of 30°, the generated object information had a total of 5,130,620 vertices making 5,178,256 triangular faces. Other software could have been investigated, however minimal differences have been noticed between exporting from CAD software and SpaceClaim has proven to be the most reliable previously.

To optimize the tessellated objects manually would require a lot of time, therefore it was decided to utilize the scripting interface within Blender 3D to create custom toolboxes to incorporate a semi-repeatable and structured workflow before a final manual adjustment of mesh information. The workflows designed were split into several steps: scaling of the objects, surface shading, removal of duplicated vertices, decimation of similar planner surfaces and creating an edge split modifier. The discussion of their selection and application is out of scope for this discussion.

One example model is a fence panel situated around the cell originally contained 2,666 vertices, which has been reduced to 48 due to many features that were deemed unnecessary. The four main features found, which was common across many objects, were: holes within components, thickness of extruded components, joining of components that can be separated, and edge modifications such as chamfers and bevels. Although these have been removed within minimal loss of information for this application, other applications may require details of this nature to be left in. This adds further complexity to a fully automated solution, of which further discussion is omitted from this article. This approach of the workflow combined with manual optimization of the described features led to an overall reduction of vertices by 85.95%.

After the meshes were optimized, three additional steps were required to fully mimic the physical asset's process. The first of these is rigging, the name associated to the application of skeleton bone structures utilizing hierarchical importance to simplify movements changes of mesh information. This can often reduce moving a complex system in the 3D space to a series of 1D manipulations which then cascade through the hierarchic structure. For example, the rotary table that can position itself over the pogo matrix has a rig applied consisting of three additional parental structures allowing for linear positional changes for two of the axis' to move along and vertically adjust with the third being for rotation of the table.

Materials were then created, which affect how models look and ultimately determine the realism within the 3D environment. These were created from reference photographs, which ultimately reduced to 24 unique materials specifying the color pigment, the reflectiveness of the surface affecting the light response and roughness of the surface [14].

The final preparation of the model was to produce animations of any components that followed a path from a single trigger. For example, the door opening and closing is triggered through a Boolean value signifying whether is open or closed. Therefore, an animation curve was created to be used to open or close the doors whenever the door data state updates.

ESTABLISHING THE LIVE DATA CONNECTION

The master PLC drives all the logic for the cell, which has a central data block containing the 61 states of information used in the digital twin: two for the cell, six for the robot joints, two for each of the five bridges, four for each of the two pogo sticks on each bridge and three for the rotary table. These can be seen in Figure 3. Festo motor controllers (CMMP-AS-C2-3A-M3) are used to control the pogo heights, which can be queried to obtain the current pogo heights. Clamping units, that are used to lock the pogo and bridges movements, are pneumatically controlled via valves on a Festo valve island from the PLC and hence the states are always known. The KUKA KR 270 R2700 ultra robot joints and end effector position are known through the KUKA robot controller. This allows the last set of information of the bridge and pogo positions to be calculated within the logic of the programming when they are moving by calculating the relative end effector position from set reference points of the respective pogo or bridge positions.

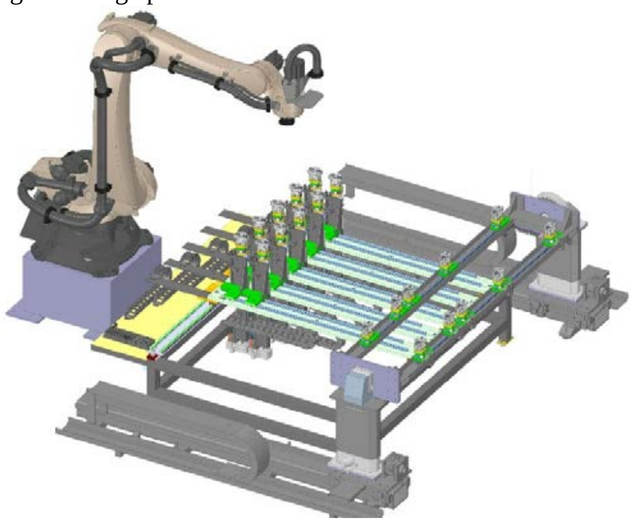

Figure 3: Reconfigurable robotic cell showing the robot, 10 pogo sticks within the pogo matrix and rotary table.

A live data connection has been implemented from the local cell network using an industrial computer acting as a bridge between the local cell network and the shop floor network. Data is sent from the central PLC as a structured byte array and sent to the computer using the TCP/IP protocol. This is received and parsed into 17 structured objects of information that is filtered to only send changes over the network. Any messages that contains significant changes to object data are sent to both an SQL database and as JSON strings of information over the shop floor MQTT message broker using a pre-determined set of topics. The message broker sends out two sets of information; one standard state information as previously described and another in a Kepware IoT Gateway specific format following the structure type below:

[{"id": "Channel1.Device1.Tag1","v": 42}, ….]

This array of a specific structure allows either partial or the full set of tags to be sent across to Kepware declaring the registered ID and new state value. The channel, device and tag information is how the tag is found inside Kepware, to then update the state to the value being passed in, which the tag and data type must already be setup before any data is transmitted. This system was implemented using Node-RED. Figure 5 shows the node layout that was implemented to follow the described implementation.

DISCRETE-EVENT SIMULATION CREATION

Siemens Plant Simulation is a discrete-event software tool that models a high level process, which includes networking connectivity to Kepware to request information via OPC/UA. Real-time state values can then be compared to the simulated data to determine current performance. A visual of the simulation is shown in Figure 4 displaying each of the key processes with overall process information including: start time, elapsed time, estimated remaining time, total process time, nominal time and expected completion time. This information is sent across to Kepware via OPC/UA, which is sent on to the MQTT broker to allow this information to be viewed in the digital twin software.

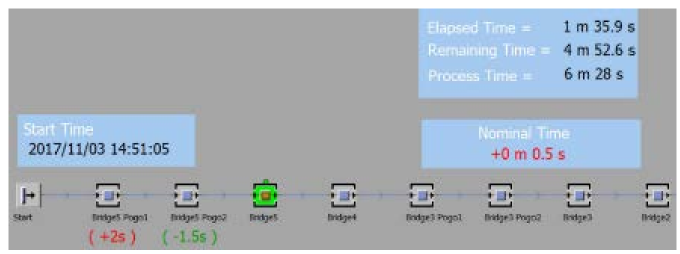

Figure 4: Extract screenshot of Siemens Plant Simulation for the Discrete Event Simulation of the process.

DIGITAL TWIN IMPLEMENTATION

All previous parts form together to create the digital twin prototype in the real-time engine, Unity3D. The software approach follows the three-tiered approach separating data, logic and user interaction. To fully mimic the cell movements, even though the base models and materials looked correct, custom functionality was needed to be programmed for objects that required movement. Due to this, linear kinematic equations were implemented for positional and rotational changes to transition objects from their current state to the new state utilizing their maximum velocity and acceleration to match the real-system. This requires more processing power but leads to a more realistic result.

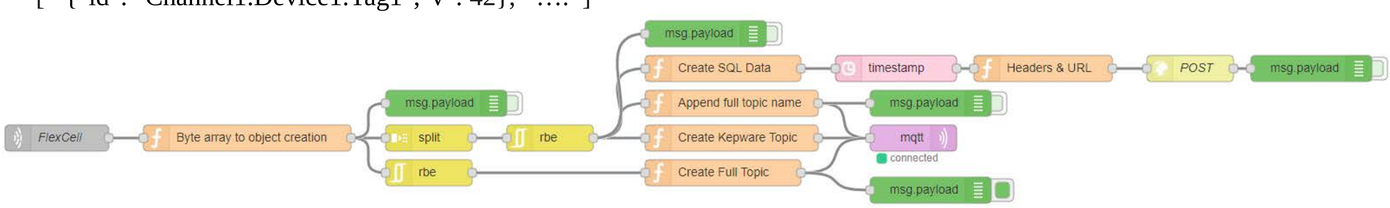

Figure 5: Node-RED implementation to read TCP/IP data, extract state values, send historical records to a SQL database and messages over the MQTT broker.

Not all state value changes require movement or lend themselves to use linear kinematics so other systems have been utilized to make these visual changes. For custom animations, such as the door sliding, an animator controller can be created that is a finite state machine with configurable transitions designed for animation control. Each state is only able to blend to the next animation when the condition is evaluated to valid, which for the door triggers are setup to change between the open and closed states (Figure 6).

Figure 6: Screenshot of the door animator containing the animations of opening and closing the doors.

The last type of display implemented are graphical user interfaces (GUIs) that have been used to show the current state of the locks, e.g. the bridges and process information that is available from the DES simulation. This technique could be implemented for all the available information, however a toggle state for the locks is deemed intuitive and position changes are better visualized by moving the fixture position in the space.

Figure 7 shows the final implementation of the digital twin with the real cell, DES simulation and digital twin software mimicking the process. The real cell is shown on the left with an overall view (top left) and a top down view from the robot end effector (bottom left). The discrete event simulation software (middle) is displaying: the current state, start time, elapsed time, remaining time, process time, nominal time and expected completion time. This information is made available through the MQTT broker, via Kepware, shown on a GUI within the digital twin providing additional information for the user to the current physical state of the system. This forms the feedback loop within the process digital twin providing real-time process information alongside the state information. The implemented real-time simulation information allows a comparison from the current status of the cell to the expected, forming a process digital twin.

CONCLUSIONS AND FUTURE WORK

A shift within the manufacturing industry is arising towards smarter factories utilizing technologies across multiple sectors. IoT is connecting networks of manufacturing cells together enabling the collection of “big data” for “big analytics”, driving data visualizations, of which one of the most significant is a digital twin. This project aimed to develop and implement an industrial framework to demonstrate how a process digital twin can be made of a reconfigurable factory environment with the focus of a single reconfigurable robotic cell.

The value shown within this paper of the implementation of a closed feedback loop real-time process digital twin is not purely in the feasibility, but more critically in the architectural approach aiming towards the creation of an industrial scalable solution of a framework that could be implemented up to a full factory / global operations including the full supply chain.

Figure 7: Final implementation of the digital twin showing the real cell (left), the discrete event simulation (middle) and an immersive digital twin within VR (right).

The architecture presented within this paper is one developed approach that has proved to be successful for a single robotic cell due to no universal approach being adopted by industry, but is agnostic to be suitable for implementation on a wide range of uses and platforms, including VR. This is the main requirement of any proposed architecture as the data inputs and visualization outputs will change over time, yet re-work of the central data systems should not be required if a suitable framework has been implemented.

Chapter two presented the investigation into the core functionality required of a digital twin from current industrial papers. Although there was some discrepancy, the prototype developed defined as a virtual mimic of a physical asset in the real-world utilizing real-time data suits the extracted commonalities. Chapter three described the methodology and the implemented universal system architecture. Chapter four discussed the design and implementation of the system architecture, including: setup and optimization of the 3D representation that achieved a vertex reduction of 85.95%; the data connection between the real cell and digital twin; the digital twin prototype that utilized a real-time discrete event simulation in parallel to drive the closed-loop digital twin providing the user extra information. This presented a use case with value beyond a simple mimic to monitor the real world.

Overall, the importance of the digital twin prototype is in the accomplished functionality displaying that one of the core topics within Industry 4.0 is possible. With further integrations from readily available technologies there exists the potential for significant savings, added value and improvements to the manufacturing sector. The work completed within this project achieved the creation of a process digital twin that closed the simulation feedback loop in real-time. Further developments would include: investigation into how the feedback loop can be automated utilizing standard operation procedures; investigation into how artificial intelligence can be used to examine historical trending of process nominal times and further investigation into the value of digital twins to drive the generation of new functionality with industrial case-studies to present business cases for investment. The creation process for the digital twin was a manual process with little investigation to how the generation could be automated, which should be explored to develop a turnkey solution of going from structured information about an area of interest to a deployable digital twin.

NOMENCLATURE

API – Application Programming Interface
CAD – Computer aided design
CPS – Cyber Physical System
DES – Discrete Event Simulation
ERP - Enterprise resource planning
GUI – Graphical User Interface
IoT – Internet of Things
MQTT - Message Queuing Telemetry Transport
PLC – Programmable Logic Controller
PLM - Product lifecycle management
VR – Virtual Reality

REFERENCES

[1] D. M. Grieves, "Digital Twin: Manufacturing Excellence through Virtual Factory Replication," 2014. [Online]. Available: http://innovate.fit.edu/plm/documents/doc_mgr/912/1411.0_Digital_Twin_White_Paper_Dr_Grieves.pdf. [Accessed 3 December 2016].

[2] The Manufacturer, "UK Manufacturing Statistics," Hennik Research, 2017. [Online]. Available: http://www.themanufacturer.com/uk-manufacturing-statistics/. [Accessed 26 April 2017].

[3] EEF, The Manufacturer's Organisation, 2017. [Online]. Available: https://www.eef.org.uk/campaigning/campaigns-and-issues/manufacturing-facts-and-figures. [Accessed 26 April 2017].

[4] The Manufacturer, "Annual Manufacturing Report," Hennik Research, London, 2017.

[5] GE, "What is a Digital Twin?," GE, 2017. [Online]. Available: https://www.ge.com/digital/power-digital-twin. [Accessed 26 April 2017].

[6] L. Slowey, "What is a digital twin," IBM, 16 March 2017. [Online]. Available: https://www.ibm.com/blogs/internet-of-things/digital-twin/. [Accessed 26 April 2017].

[7] K. Lewis, "Digital Twin with Watson," IBM, 27 April 2017. [Online]. Available: https://www.ibm.com/blogs/internet-of-things/digital-twin-watson-iot/. [Accessed 29 April 2017].

[8] L. Slowey, "What is Digital Twin?," IBM, 16 March 2017. [Online]. Available: https://www.ibm.com/blogs/internet-of-things/digital-twin/. [Accessed 27 March 2017].

[9] Z. Feuer and Z. Weissman, "The value of the digital twin," SIEMENS, 16 January 2017. [Online]. Available: https://community.plm.automation.siemens.com/t5/Digital-Transformations/The-value-of-the-digital-twin/ba-p/385812. [Accessed 25 April 2017].

[10] ANSYS, "ANSYS Discovery SpaceClaim," ANSYS, 2 April 2018. [Online]. Available: https://www.ansys.com/en-gb/products/3d-design/ansys-spaceclaim. [Accessed 2 April 2018].

[11] Blender, "About," 2017. [Online]. Available: https://www.blender.org/about/. [Accessed 15 April 2017].

[12] "Node-RED," JS Foundation, 2017. [Online]. Available: https://nodered.org/. [Accessed 4 December 2017].

[13] Unity Technologies, "Unity," Unity Technologies, 2017. [Online]. Available: https://unity3d.com/. [Accessed 15 April 2017].

[14] Unity, "Material parameters," Unity, 2017. [Online]. Available: https://docs.unity3d.com/Manual/StandardShaderMaterialParameters.html. [Accessed 20 April 2017].

ANNEX A

SEQUENCE DIAGRAM DISPLAYING HOW THE CORE SYSTEMS INTERACT WITH EACH OTHER.