Design, Development and Deployment of ISO 23247 Standardized Digital Twin on Machines

A dissertation submitted in partial fulfilment of the requirements for the degree of

Master of Technology

Submitted by

Roll No	Name
213100073	Ashutosh Pathak

Under the guidance of

Dr. Makarand S Kulkarni

Department of Mechanical Engineering
Indian Institute of Technology Bombay
Powai, Mumbai , India

Spring 2023

Dissertation Approval

This dissertation entitled **Design, Development and Deployment of ISO 23247 Standardized Digital Twin on Machines** by **Ashutosh Pathak**, Roll No. **213100073**, is approved for the degree of **Master of Technology**.

.....	.....
Examiner	Examiner
.....	.....
Supervisor	Chairman

Date: .....

Place: .....

Declaration

I declare that this written submission represents my ideas in my own words. Where others' ideas and words have been included, I have adequately cited and referenced the original source. I declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated, or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will cause disciplinary action by the Institute and can also evoke penal action from the source which has thus not been properly cited or from whom proper permission has not been taken when needed.

. . . . . .

Ashutosh Pathak
Roll No.: 213100073
Date:
Place: IIT Bombay

Acknowledgements

I would like to express my special thanks of gratitude to my guide Dr. Makarand S Kulkarni for guiding me throughout this project and providing me with useful directions and inputs whenever necessary.

I would also like to express my gratitude to Department of Mechanical Engineering IIT Bombay for providing me with this opportunity. I am thankful to Mudit Sand whose previous work has laid foundation for the development of my work. I am grateful to each and every member of CAM lab who helped me in my work.

I would also be thankful to all other persons who helped me during this project knowingly or unknowingly.

Abstract

In today's fast-paced global environment, access to valuable information is crucial for enterprises to gain a competitive edge. In the manufacturing industry, much of the work still relies heavily on the operator's skills, lacking autonomous decision-making capabilities. One of the primary reasons for this limitation is the absence of historical data for Machine Intelligence. However, this challenge can be overcome by digitizing manufacturing enterprises, enabling decentralized informed decision-making. Digital Twin, a recent concept in the context of Industrial Revolution 4.0, provides a solution by allowing the modeling of manufacturing processes. Each manufacturing machine comprises numerous Observable Manufacturing Elements (OMEs) that generate vast amounts of data, which can be digitally modeled and analyzed to facilitate informed decision-making.

This project focuses on the development of Digital Twin 3.0, based on ISO 23247 standards and built upon previous versions of Twin work conducted in our lab. The project also establishes an architecture for physical installation to ensure the standardized deployment of the Twin. Previous versions of the Twin required manual initialization of various agents by entering Port Numbers and IP Addresses in Python scripts, followed by script activation. To address this, two types of user interfaces have been developed: one for server initialization and launching the Digital Twin User Entity (Platform), and another for initializing Twin Agents and role agents.

Furthermore, this project extends the functionality of the Twin beyond the Operation and Maintenance Sub-system Entity. A comprehensive Application and Service Sub-system Entity has been developed, encompassing machine condition monitoring, tool condition monitoring, report generation, alarm display, and recommendation provision. Additionally, a framework for the development and deployment of custom downloadable functions on the Twin has been established. The project also includes the creation of a binary classification model and predictor module for tool failure prediction, which has been integrated into the Application and Service Sub-system Entity. Overall, this project demonstrates the potential of Digital Twin technology to enhance decision-making and operational efficiency in the manufacturing.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

The pace at which the future is unfolding exceeds our expectations. The rapid rate of change anticipated for tomorrow will make our current progress seem sluggish. The onset of Industry 4.0 has already commenced, and one crucial catalyst in this transformative era will be the utilization of Digital Twins empowered by Artificial Intelligence (AI). It is widely acknowledged that there will exist two categories of companies in the future: those that embrace and integrate digitization and AI into their operations, and those that will become obsolete.

As Elon Musk aptly stated, "Companies must strive to develop AI or they risk losing competitiveness. Essentially, if your competitor is actively pursuing AI development, they will outperform you." The coming decade will revolve around the collaboration between humans and AI, and Digital Twin technology will serve as a platform for visualization, communication, and decisive action.

1.2 Background

1.2.1 Industrial Revolution

Industrial revolution is defined as major changes and transition in manufacturing and industrial processes with new innovative technologies. The evolution of Industrial revolution with time and advent of relevant technologies can be understood from this figure.(Refer : 1.1)

Industry 1.0 (1784)	Industry 2.0 (1870)	Industry 3.0 (1969)	Industry 4.0 (2011)
Machines powered by water and steam	Electricity invention, Assembly line and mass manufacturing	Electronics, Automation and use of computers in manufacturing	Cyber Physical Systems(It connects physical world to digital world)

Figure 1.1: Industrial Revolution Evolution with time

1.2.2 Industry 4.0

Industry 4.0 is a new phase in industrial revolution that introduces intelligent networking of machines and processes for industry with the help of ICT(information and communication technologies). Some features of Industry 4.0 are:

- Machines are connected with other machines
- Factories are connected with other factories
- System and processes can be accessed from remote location for maintenance and updating software

1.2.3 Comparison between Traditional Manufacturing and Industry 4.0

Manufacturing is evolving day by day by the use of new Industry 4.0 concepts like cyber physical systems, digital twin, cellular manufacturing, Augmented Reality, Artificial Intelligence,etc. A comparison between traditional manufacturing and Industry 4.0 manufacturing can be understood from this figure.(Refer : 1.2)

1.2.4 Challenges of Industry 4.0

In implementing Industry 4.0 there are certain challenges which we can come across and they are needed to be addressed properly for major breakthrough in manufacturing. Some of these challenges are :

1. Huge Investment required for setup of modern machines and infrastructure
2. Need to incorporate new business models based on data generated by sensors or twins

Traditional	Industry 4.0
Mass Production	Customization as per user requirements
Large factories to produce big volume of specified products	Smart Factories with flexible production at a competitive cost
Orderly planning based on anticipation of demand (Push Type System)	Dynamic production in accordance with real time demand (Pull Type System)
Product purchase	Use of product as a service
Focus on minimization of costs	Focus on maximizing ROCE and profitability
Work Rigidity	Flexible in Work Organization (Cellular Manufacturing)

Figure 1.2: Traditional vs Industry 4.0 manufacturing

3. Resistance to change in accepting and implementing new standards
4. Reorganizing the process by forward and backward integration of value chains through pilot study
5. Standardization through ISO framework for interoperability between systems
6. Data Management
7. Data Security

1.2.5 Benefits of Industry 4.0

Some of the most important benefits which we can achieve from implementing Industry 4.0 in our process and operations are :

- Business objectives : operational excellence (enhanced efficiency, customized products, etc.) and expanded services (higher revenues through digitally refined products, access to new markets)
- Cost optimization
- Revenue gains
- Improved supply/demand from real time data

- Reduced downtime
- Quality products
- Speed of delivery through rapid prototyping involving 3D printing
- Improved customer satisfaction

1.3 Characteristics and Capabilities of an Industry 4.0 ready equipment

In this section we will discuss about the drawbacks which we face while working with legacy equipment and what makes an Industry 4.0 equipment futuristic and collaborative.

1.3.1 Legacy based euipments drawbacks

In legacy equipment the major drawbacks are listed below :

- they are not inter operable
- they have no transparency of data
- they have no common communication protocol
- they have loss of raw material, a large number of non conform defective products, delayed deliveries and non productive work hours
- they do not have autonomous response to disturbing events which do not allow them a quick reaction to changing demands

1.3.2 Key characteristics of an Industry 4.0 ready equipment

The capabilities required for an Industry 4.0 ready equipment to achieve the following characteristics are Sensors, High Speed Internet, Electricity, Semiconductors, Data storage equipment, micro controllers, processors, etc.

1. **Connected**
    - continuously pull both traditional data-sets and new sensors data-sets

- collaboration across equipments and suppliers-customers based on real time data

2. **Optimized**
    - reliable and Artificial Intelligence enabled prediction of demand capacity
    - reduced downtime and increased production efficiency
    - high automation with minimal human interaction
    - minimized cost of production
3. **Transparent**
    - live metrics and tools to support real-time decision making
    - transparent customer order tracking
4. **Proactive**
    - predictive anomaly and supplier quality issues identification and resolution
    - automated inventory management
    - real time safety monitoring
5. **Agile**
    - adaptive and flexible scheduling
    - implementation of product changes to see immpact in real time
    - configuration of equipment

1.4 Digital Twin and the value it provides in manufacturing

A digital Twin is a virtual representation of process, object or service that serves as real time digital counter part of actual physical process, object or service. Using digital twin we can track the past, have a deeper insight of present, predict and influence future. Types of digital twins can be observed from the following table:

What is twinned	Example
Process	Manufacturing Processes
Plant	Automotive assembly plant
Device	Motor drive
Product	Jet Engine, MRI machine
Production Line	Drive train Assembly
Person	Operator or Maintenance person

Digital Twins are becoming more important recently as implementation of twin is becoming easy day by day with the ongoing digital revolution. Digital Twins find value in all phases of manufacturing and this can be understood by analyzing this figure. (Refer : 1.3)

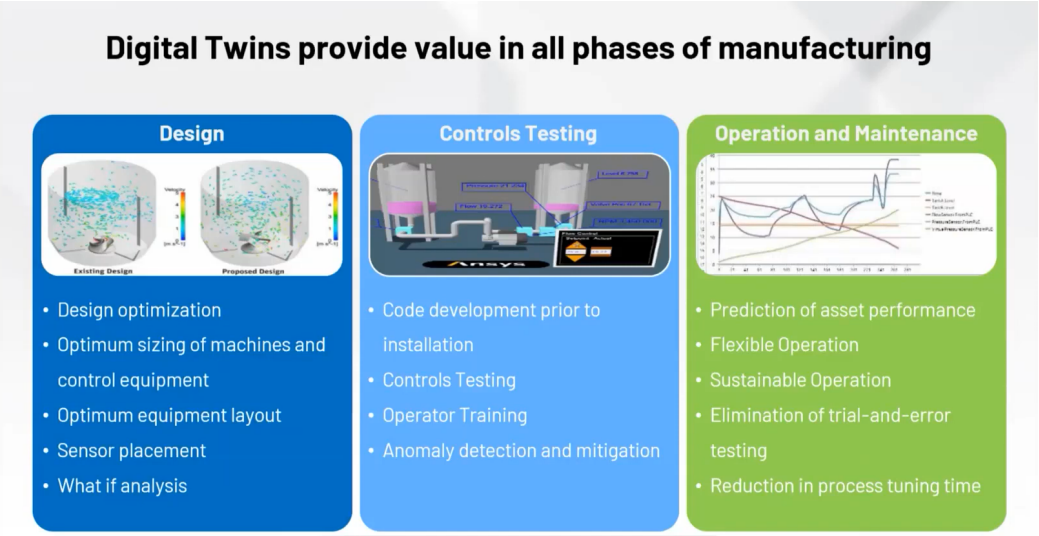

Figure 1.3: Real Value of digital twin in manufacturing ([6])

1.5 Digital Twin ISO 23247 framework

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees.
ISO 23247 defines framework to support the creation of Digital Twins of observable manufacturing elements(OMEs). The scope of this ISO series are

defined in four parts :

- Overview and general principles
- Reference architecture
- Digital representation of manufacturing elements
- Information exchange

1.5.1 Part 1 : Overview and general principles

This part of ISO 23247 provides overview and general principles of Digital Twin for manufacturing. Whatever in the scope of this and subsequent parts will be discussed and following are outside the scope of ISO 23247:

- selection of the implementation methods and technologies for a Digital Twin for manufacturing
- selection of the communication protocols for a Digital Twin for manufacturing
- selection of the manufacturing devices and other resources to be represented by a Digital Twin
- selection of the manufacturing processes to be represented by a Digital Twin
- selection of the manufacturing products to be represented by a Digital Twin
- design and process planning, and other non-manufacturing stages of the product life cycle

Important terms and definitions

entity : thing (physical or non-physical) having a distinct existence
Digital Twin : fit for purpose digital representation of some realized thing or process with a means to enable convergence between the realised instance and digital instance at an appropriate rate of synchronisation
presentation : manner in which information is displayed for use by a human
representation : manner in which information is stored for interpretation by a machine

Abbreviated terms

API	Application Program Interface
DCDCD	Data Collecting and Device Controlling Domain
DTME	Digital Twin of Observable Manufacturing Element
ERP	Enterprise Resource Planning
IE	Information Exchange
IPC	Inter-Process Communication
MES	Manufacturing Execution System
O and M	Operation and Management
OME	Observable Manufacturing Element

Concept of Digital Twin

A digital twin for manufacturing updates itself as its physical counter part changes to represent its status,stat of resources and any other observable conditions. The digital twin enables functionalities to synchronize its representation with its corresponding observable manufacturing elements by constantly changing operational and environmental data. Application and benefits of digital twin can be understood from this figure (Refer : 1.4) of digital twin concept.

General principles of Digital Twin Framework

Digital twin framework for manufacturing provides guidance on how to construct digital twin, specifies how to develop applications for inter operate and how to develop agents for data transfer between different sources. The following figure 1.5 shows high level concept of digital twin framework. We can see from figure that digital twin and physical world(OME) are connected and synchronized through Data Collection and Device Control. And this twin is transferring this real time synchronized data to User which can be ERP, MES or person for action.

Requirements of Digital Twin for manufacturing

1. Data acquisition - collect sensory data through sensors installed on or around equipment

2. Communication - shall transfer data and information between elements of Digital Twin

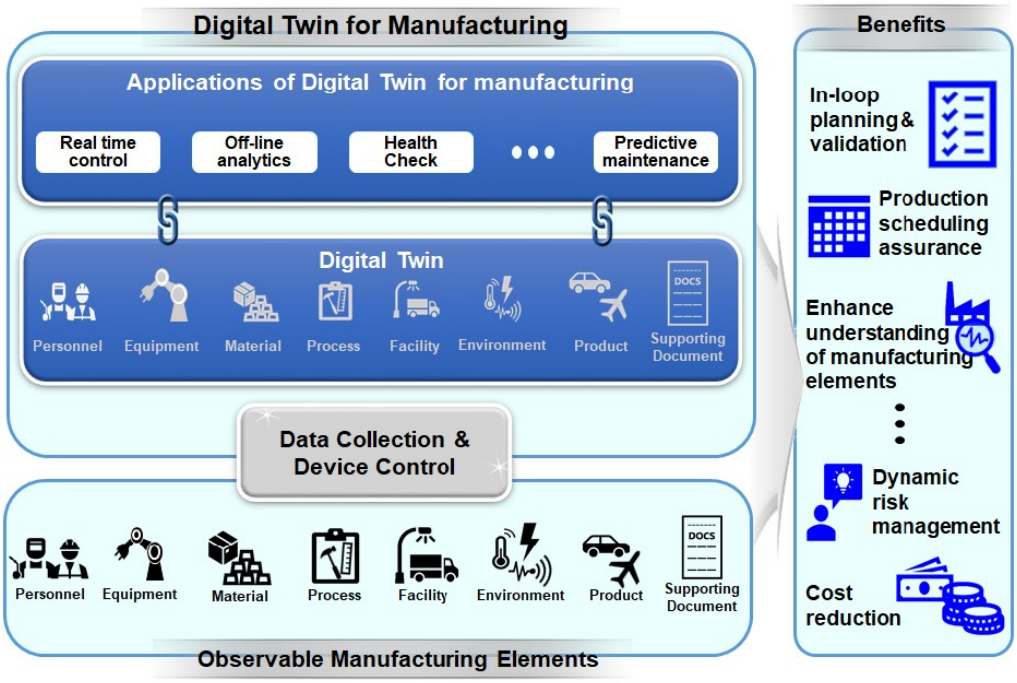

Figure 1.4: Concept of Digital Twin ([1])

3. Presentation - information shall be represented in a format human or user interface can recognize
4. Data analysis - analyze data to understand state OMEs
5. Management - Twins should be managed to optimize resources
6. Synchronization - real time synchronization should be there through DCDCE
7. Data store - a store to store data permanently or temporarily for data modelling, exchange and analysis
8. Simulation - it should simulate manufacturing elements in operation
9. Viewpoint - it should support different viewpoints for different objectives

1.5.2 Part 2: Reference architecture

Digital Twin reference architecture for manufacturing defines reference models and architectural views. The architecture increases understanding of

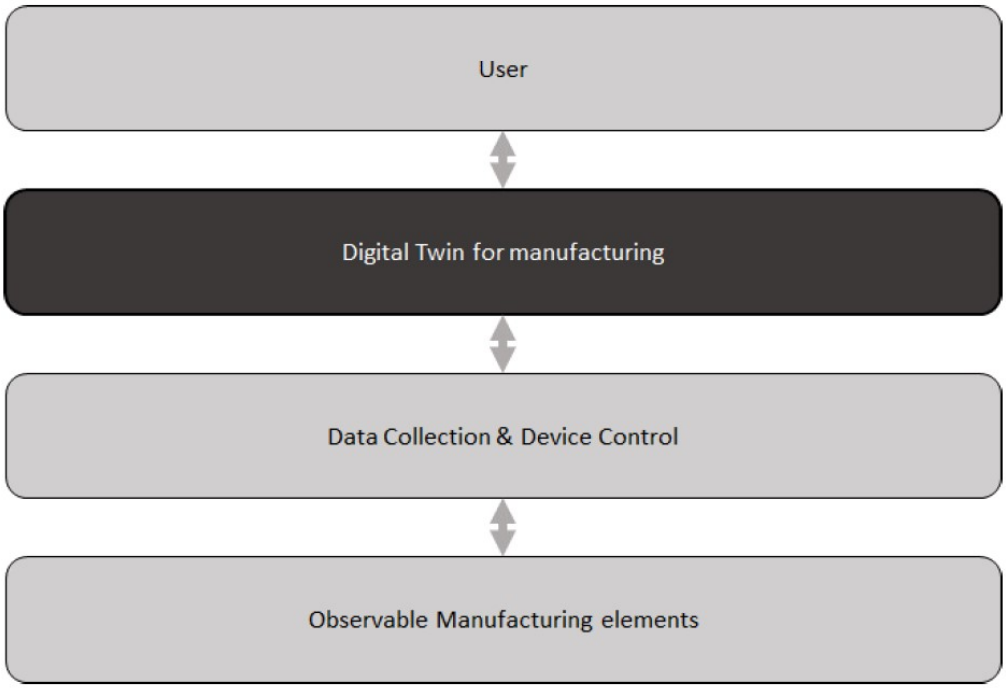

Figure 1.5: Digital Twin framework ([1])

Twins for various stakeholders. Figure 1.6 outlines reference architecture defining relevant reference models and architectural views.

Reference Model

Domain-based reference model is useful to describe various tasks that have to be performed in separate areas, by allowing a logical and sometimes physical subdivision.
Entity-based reference model breaks down Twin at system level in conjugation with domain concept in order to facilitate understanding of system composition of Digital Twin. Both model can be understood from this figure. (Refer : 1.7)

Architecture View

Two types of architectural views are defined in ISO 23247 for Digital Twin and they are functional view and networking view. Below figure 1.8 shows functional reference architecture to support requirements defined in ISO

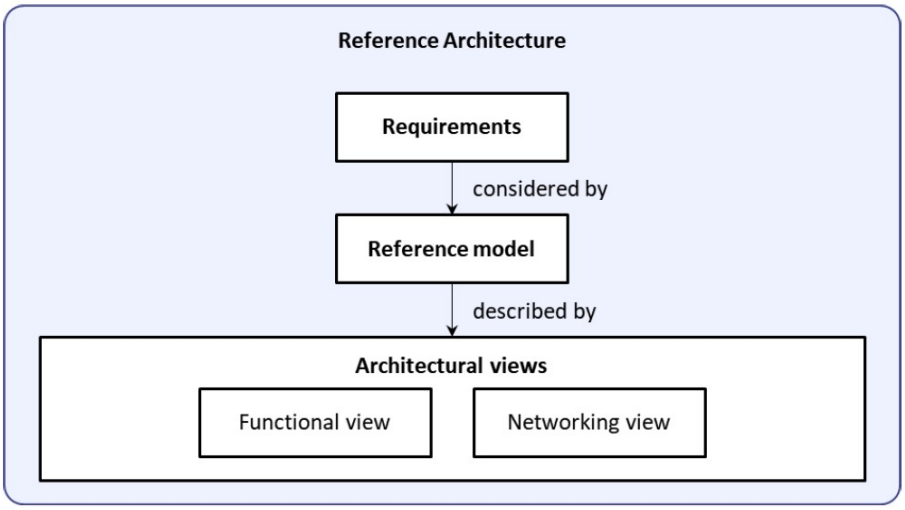

Figure 1.6: Outline of Digital Twin reference architecture for manufacturing ([1])

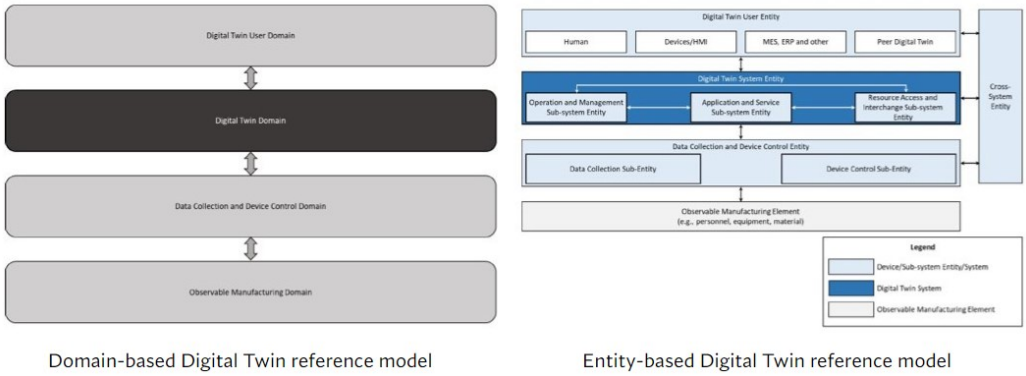

Figure 1.7: Outline of Digital Twin reference architecture for manufacturing ([1])

23247. In this **functional entity of observable manufacturing element** are not in the scope of ISO 23247 but depicted here for better understanding. Any detailed architecture development has to be according to this reference architecture. Reference architecture provides us with mandatory building blocks on which detailed architecture can be developed.
The reason for standardizing these is that it is soon expected that digital twins will be relatively more common in the near future. And therefore, if

everyone builds it based on a certain reference architecture then it would be much easier to ensure that there is interoperability between these Digital Twins. So that is the reason why this architecture has been proposed.
This **entity of Observable Manufacturing Elements** consists of physical assets. These could be an entire production line or this could be a single machine or a motor inside a machine as it depends on how we are defining the physical assets. And everything leaving this OMEs functional entities in figure 1.8 is part of Digital Twin framework.

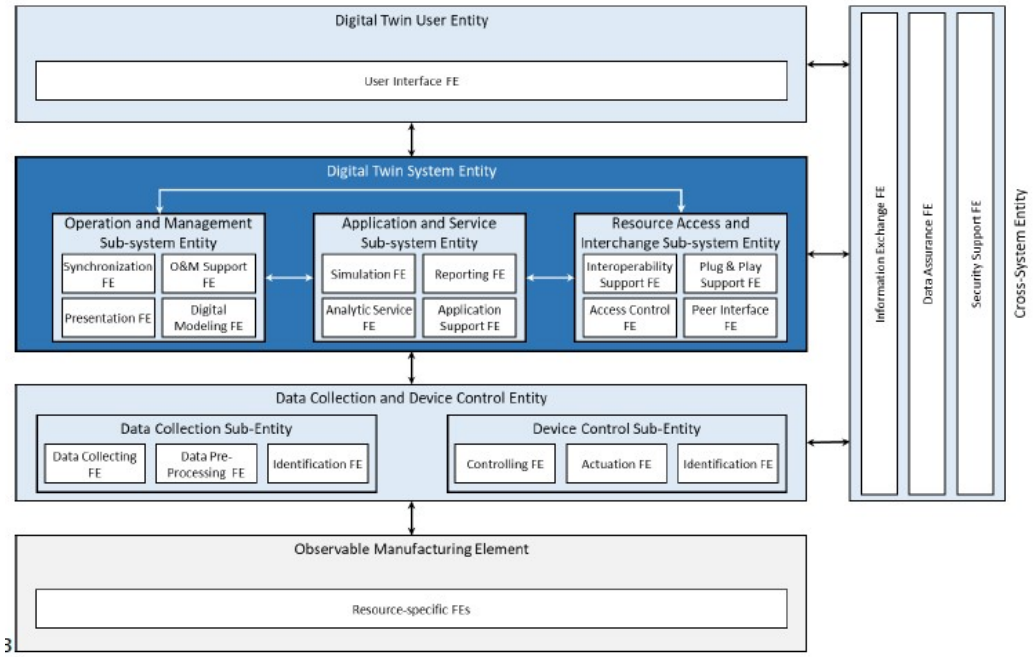

Figure 1.8: Functional reference architecture of Digital Twin ([1])

1. **Data Collection and Device Control Entity** - One of the most important thing is connection between this data collection and device control entity and OMEs functional entity. Once the connection is established the first thing our Digital Twin should be able to do is data collection and device control. In this entity device can be sensor, controller, actuator, or entire machine based on our scope of Digital Twin framework. Now there are two sub entities **data collection sub-entity** and **device control sub-entity**. **Data collection sub entity** collects data from physical assets and pass it on to the Digital Twin System Entity. Suppose there is no human involved in the Digital

Twin User Entity and Twin need to make changes in physical assets automatically based on analysis from System Entity then it used **Device Control Sub entity**. These sub entities has functional elements embedded in it and some these have been defined below.

(a) **Data collecting FE in data collection sub-entity** - it provides data collecting functionality from OMEs. For eg. data can be coming from multiple sensors and these are connected to a DAQ(Data Acquisition System) which in turn can be connected to an edge device and this edge device is connected to data collecting functional entity of Digital Twin. This entire communication has to follow some standard protocol and thus we need to design this data collection entity in such a manner that it should be able to understand multiple standard data communication protocols. These understanding of multiple data transfer protocols by data collection entity enhances interoperability of Digital Twin.

(b) **Data pre-processing FE in data collection sub-entity** - the raw data collected might need some preprocessing such as filtering and aggregation of data. For eg. suppose we are getting data of voltage every 5 seconds and we need max voltage then we need to define data pre processing entity to filter out maximum voltage before passing to Digital Twin System Entity

(c) **Controlling FE in device control sub-entity** - it provides functionality of controlling OMEs by the request from Digital Twin

(d) **Actuation FE in device control sub-entity** - it provides actuation of OMEs by the request from Digital Twin

(e) **Identification FE in data collection and device control entity** - it provides functionality for identification of OMEs and its data that is to be collected and controlled uniquely. Proper identification should be there like what data is coming from which sensors and what control command is to be passed to which actuator.

2. **Digital Twin System Entity** - It is a kind of brain of Digital Twin framework where all analysis, simulation, report generation and management is done. It is second core layer of Digital Twin framework where all our models, analysis involving Machine learning, applications, etc. resides. In this architecture it is assumed that there is single digital twin but it can have multiple digital twins also. This Digital

Twin System Entity has following three Sub system entities with their functional entities.

(a) **Operation and Management Sub-system Entity** - This sub system entity defines how a digital twin need to be operated and managed as per management requirements like digital twins need to evolve with time through digital modelling, visualization and synchronization.

   i. **Synchronization FE** - it provides functionality of synchronizing the status of the visualized digital entity with the status of the observable manufacturing element, or vice versa. This means that if there is a change in status or configuration of physical asset that should reflect in Digital Twin also.

   ii. **Presentation FE** - it provides functionality of presenting observable manufacturing element as digital entity in conjunction with digital modelling FE

   iii. **Digital modelling FE** - it provides functionality of interpreting information of observable manufacturing element to understand its physical properties, status, etc. For eg. a CAD(Computer Aided Design) model

   iv. **O and M support FE** - it provides functionalities of operating and managing Digital Twin

(b) **Application and Service Sub-system Entity** - Operations and Management sub system entity makes use of some services for analysis and decision making which is provided by this sub system entity. The reason these applications and services are not hard coded in Operation and Management sub system entity is that we can always improve our service. Suppose we need to upgrade simulation model then we can easily do that by replacing simulation functional entity.

   i. **Simulation FE** - it provides simulation functionality. For eg. FEA (Finite Element Analysis)

   ii. **Analytic service FE** - provides functionality of analysing data collected from OMEs and as a result of simulation. This will include all Artificial Intelligence and Machine Learning models for data analysis and decision making.

   iii. **Reporting FE** - it provides functionality of report generation of production results and analysis of simulation. For eg. Digital Twin can raise alarms based on reports.

iv. **Application support FE** - provides functionality of hosting platform for implementing predictive, reactive, open and closed loop applications. It provides a gateway of analysis using pre developed custom applications.

(c) **Resource Access and Interchange Sub-system Entity** - This is the third sub system entity in Digital Twin System Entity which allows plug and play, access controls, peer interfaces for user interface functional entity. Suppose this Digital Twin has another Digital Twin as User Entity then this sub system entity defines how interchange of information through peer interfaces takes place.

i. **Interoperability support FE** - it provides functionality of inter working with other Twin system using peer interface FE

ii. **Access control FE** - it provides functionality of controlling access of Digital Twin user entity to OMEs(Observable Manufacturing Elements) in collaboration with security support Functional Entity in **Cross System Entity**. It also defines access levels based on access identity.

iii. **Plug and play support FE** - dynamic involvement of OMEs in conjugation with O and M support FE

iv. **Peer interface FE** - interfacing with other Digital Twins

3. **Digital Twin User Entity** - This User Entity can MES (Manufacturing Execution System), ERP (Enterprise Resource Planning Software) or any third party software (production planning etc.) Interoperability is required to transfer data to user interface functional element of User Entity from Digital Twin System Entity.

(a) **User interface FE** - interfacing of Digital Twin user entity with Digital Twin system entity

These functional modules are there to perform some generic functions like Operation and Management sub system entity is to do digital modelling, create visualization of digital model and enable real time synchronization of status and conditions of model with physical asset. Application and Service sub system entity is to do simulation of digital model and generate results data, analytic service to do Machine Learning based analysis of resulted simulation data and reporting these results and analysis. Report Access and Interchange sub system entity is there to provide both side interoperability to the Digital Twin User Interface as well as physical assets.

4. **Cross-system entity** - Once this Digital Twin System Entity is built it felt that there should be some cross system communication. Cross system entity means entities that resides outside of digital twin network. It enables communication between Digital Twin System Entity and outside environment.

   (a) **Information exchange FE** - exchanging information among entities of Digital Twin by appropriate networking protocols and responding to the requests based on communication.

   (b) **Data assurance FE** - provides accuracy and integrity of data in collaboration with security support FE

   (c) **Security support FE** - functionality of securing using authentication, authorization, confidentiality, integrity, etc.

This completes the loop which started from data collection from Observable Manufacturing Elements(OEMs). Then data collected is transferred to Digital Twin System Entity after pre processing with identification. Digital Modelling based on data, simulation, analysis and report generation takes place in Digital Twin System Entity for decision making. Then these reports are transferred to Digital Twin User Entity where it can be displayed using an interface. Now based on analysis decision will be taken and passed on to Device Control Sub entity for necessary changes in physical assets.

Networking View

It describes communication network that has been involved in Digital Twin for manufacturing. Main role of communication network is to provide a means to exchange information between entities across the different domains and can be visualized from the figure 1.9

1.5.3 Part 3: Digital representation of manufacturing elements

One of the most important function of Digital Twins is to have digital representation of observable manufacturing elements(OMEs). Figure 1.10 shows how Digital Twin entity manages observable manufacturing elements through a digital representation.
Digital representation of OMEs should include both static and dynamic information (for eg. static information is machine number and dynamic information is amount of material in manufacturing).

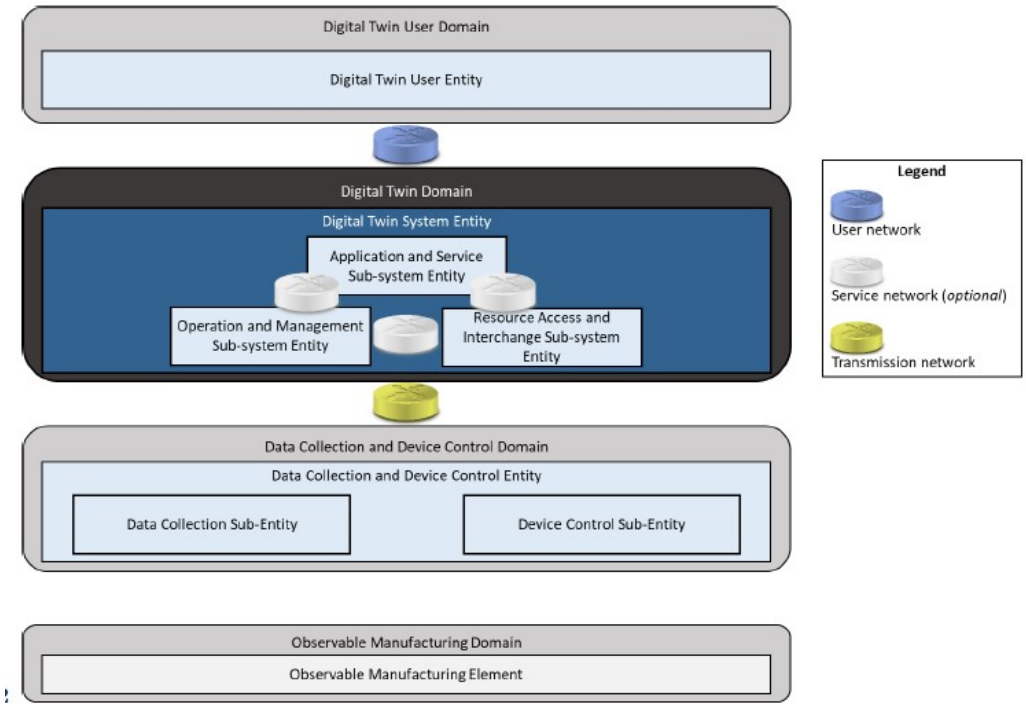

Figure 1.9: Networking view of Digital Twin reference architecture ([1])

Information attributes of observable manufacturing elements

An example of information attributes has been described in this figure 1.11 to illustrate the kind of information that must be represented with Digital Twin.
An example of static and dynamic information of equipment has also been shown in figure 1.12 for understanding.

1.5.4 Part 4: Information exchange

Each entities and Observable Manufacturing Elements(OMEs) are peers of information exchange. This document specifies four types of information exchange between entities and observable manufacturing element.(Refer : 1.13)

These four types of information exchange interface and data transfer protocol can be understood from this table :

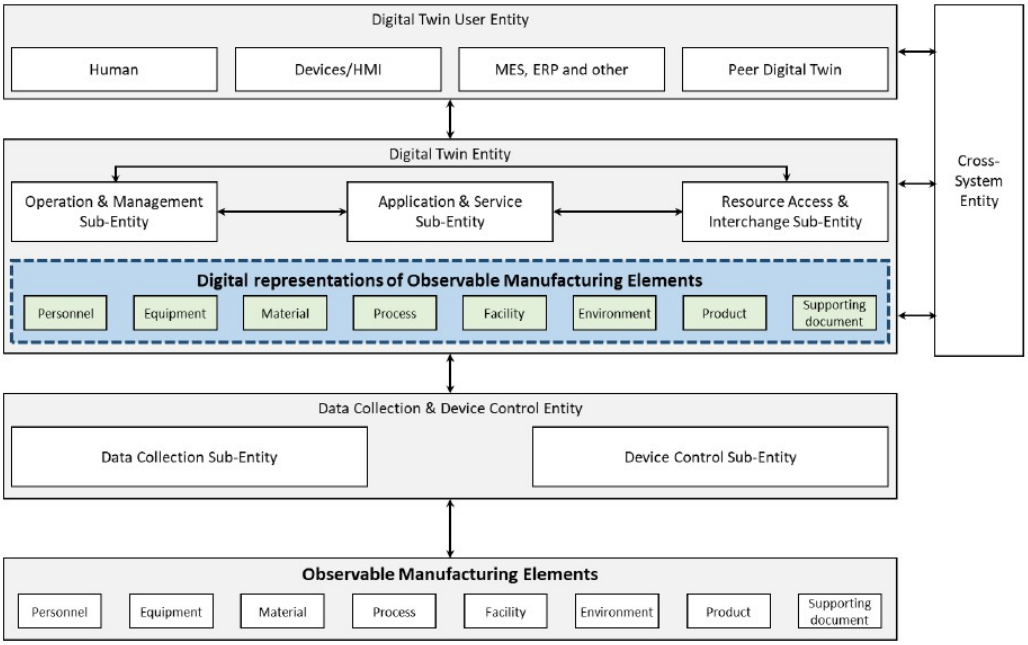

Figure 1.10: Digital representations of observable manufacturing elements in reference architecture ([1])

Network Interface(Communicating Entities)	Data Transfer protocol
IE-A(Twin User Entity and Twin Entity)	support visualization
IE-Bs(among sub-entities within Twin entity)	support presentation
IE-C(Digital Twin entity and DCDCE)	standardized protocol
IE-D(DCDCE and OMEs)	local network

Example of an agent based communication between Machines

Exchange of data should take place following some standards and protocols and it should not be random. When two twins interact for data exchange they first query what data format is supported by other twin. Suppose if first twin can share data in 8 formats and out of which only 2 formats are understood by other twin. Then data exchange can happen in any of the two common formats available with them.
In manufacturing one of the most popular open source standard is MT Connect(i.e. machine tool connect). It exchanges data in XML format and this format contains data in form of some standard tags of parameters such as

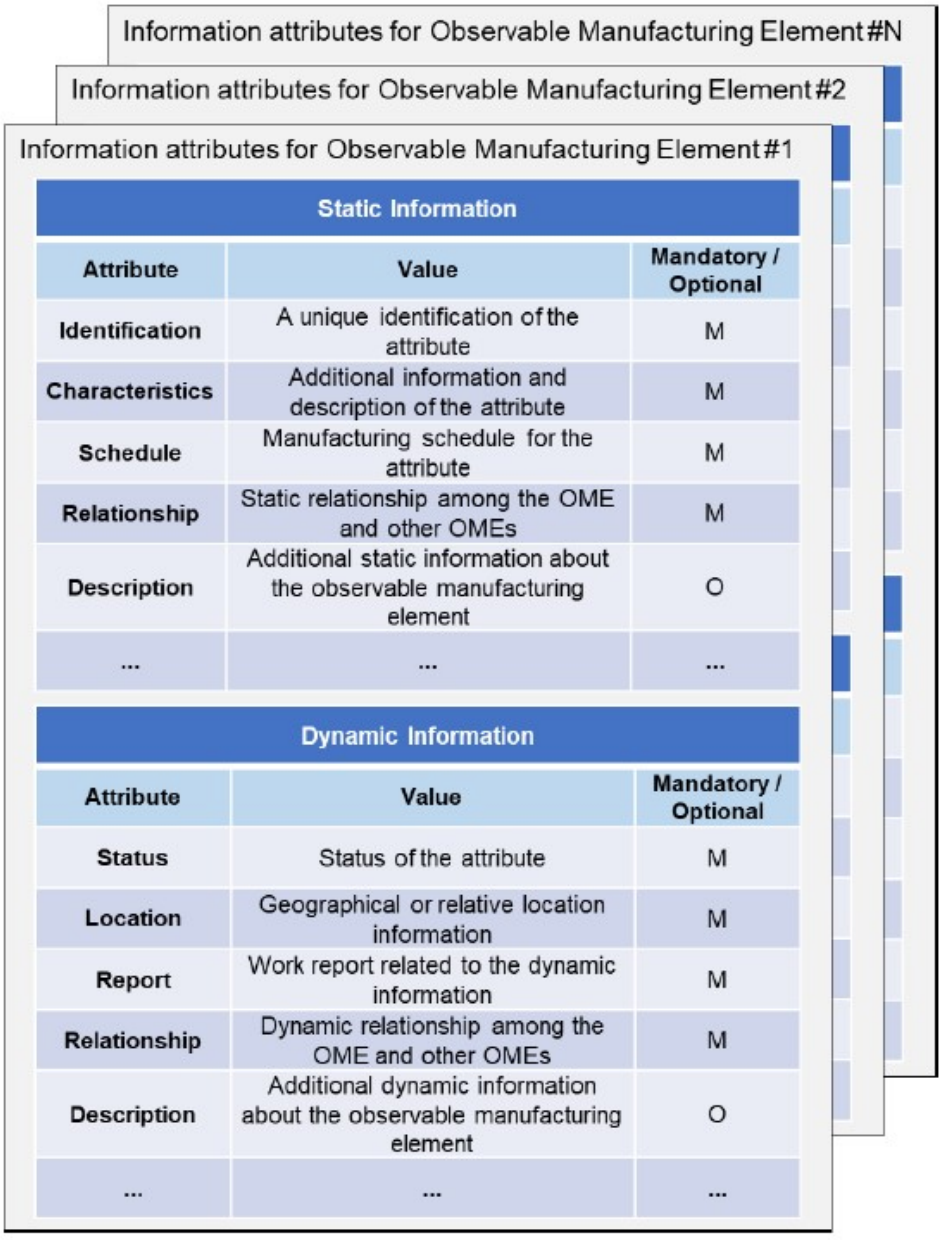

Figure 1.11: Information attributes for observable manufacturing elements ([1])

temperature, vibration level, etc. Tags are generally standardized but if some unique sensor data is collected and transmitted we can also define tag name.

Attribute	Description	Examples
Identification	Information to identify equipment	· serial number
Characteristics	Classification of equipment	· milling · turning · grinding · pressing
Schedule	Working schedule for equipment	· working schedule · maintenance schedule
Relationship	Static relationship for equipment and other manufacturing elements	· Machine #1 operates with Material #2.
Description	Additional information and explanation about the static information of equipment	· general information about equipment

(a) static

Attribute	Description	Examples
Status	Status of equipment	· on / off · working / breakdown · performance (energy usage, output) · temperature, pressure, sound / noise
Location	Location information (geographical / relative location)	· Machine #2: WorkUnit #2 in Room #3
Report	Work report related to equipment	· May 14th, 2019 9 AM to 6 PM: Regular Maintenance · May 14th, 2019 11 AM: Machine #1 reports high temperature.
Relationship	Dynamic relationship for equipment and other manufacturing elements	· Machine #1 is operated by Person #2 in WorkCenter #5.
Description	Additional information and explanation about the dynamic information of equipment	· dynamic information of equipment changing during manufacturing processes

(b) dynamic

Figure 1.12: Information of Equipment ([1])

There is an external standard data storage standard file and XML file is needed to be sent to the second entity. Now, when you do this there is a certain way in which MT Connect is expected to work and this helps us in understanding how interoperability exists.
Suppose there is person one(P1) and there is person two(P2). So person one understands Bengali and person two understands Punjabi and English.So if person one and person two want to interact, it is not possible because there

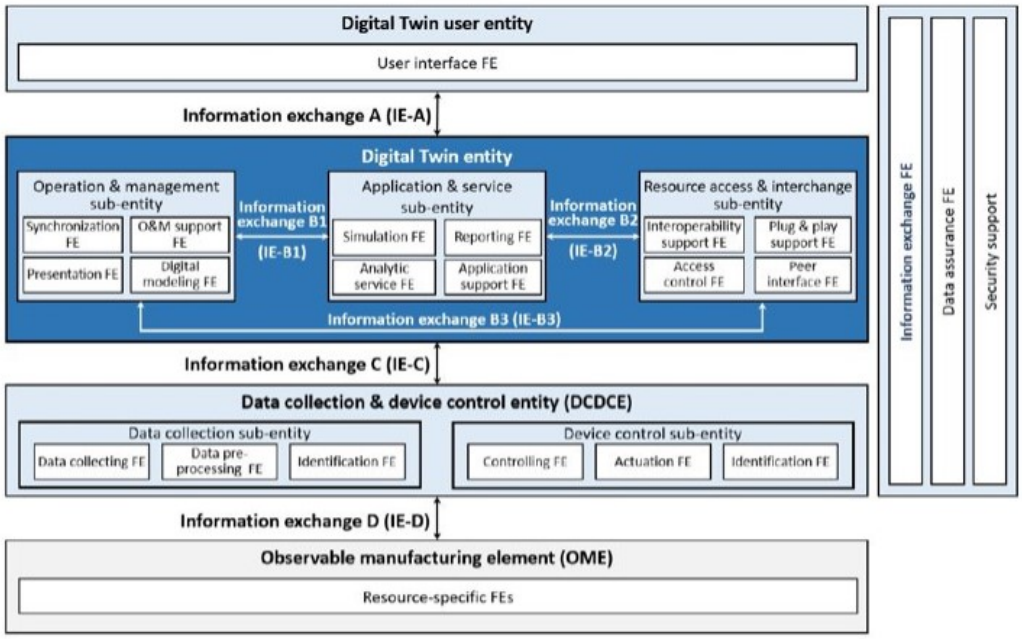

Figure 1.13: Information attributes for observable manufacturing elements ([1])

is no common language that they can understand. Now we have to design an interpreter which converts Bengali either into English or into Punjabi to facilitate communication. This means you need an interpreter which will convert Bengali into Punjabi and English and then communication will also require something called as an agent. Now, the job of interacting with the user is the job of the agent and it is onto the will of the agent and second person whether they want to exchange information in Punjabi or English as Interpreter has already converted Bengali into Punjabi and English. Now agent is requested information in Punjabi and English and it seeks information from Interpretor in Punjabi or English and transfers it to second person. We can observe that agent does not need to know Bengali as Interpretor has already converted Bengali. Now we can replace two pesons with twins as we discussed above and one Twin generated some standardized data in other format and second digital twin want data in XML format. It requires an interpretor to convert not known standard data to MT Connect and then an agent is required to transfer this XML data through MT Connect.
Let us make it more relevant and there are two machines suppose machine one is having Siemens controller and therefore Siemens has a certain format for data communication and there is another machine which has a FANUC

controller. The agent can be same but you need a separate interpreter which can facilitate communication and send data to Digital Twin User Entity. An agent will store some recent last one hour information from interpretor and any third party entity which can be a application or user interface through which user wants to interact with the machine can send a query to the agent asking what all information agent have and the agent will then respond. Getting the list of all tags, this particular application demands for some specific five tags from agent. And then the agent send data related to these five tags only which can be spindle temperature, spindle vibrations, etc. (Refer : 1.14)

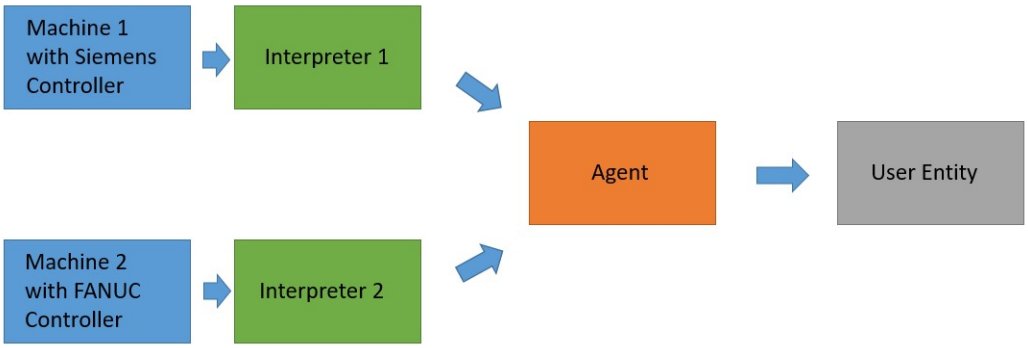

Figure 1.14: Information Exchange using Interpretor and Agent

Chapter 2

Literature Review

2.1 Azure Digital Twins

Azure offers a service that allows users to leverage the Digital Twins platform. This service empowers users to create twin-graphs based on digital representations of entire environments, encompassing diverse settings such as buildings, industries, farms, energy networks, railways, stadiums, and even entire cities. These digital models serve as valuable resources for obtaining insights that drive improvements in products, operational efficiency, cost reduction, and exceptional customer experiences.
Key Features:

- Azure Digital Twin serves as the cloud solution for IoT devices
- The Digital Twin platform utilizes the Open Modeling Language
- Querying on the platform can be performed using Structured Query Language (SQL)
- Azure provides a real-time execution environment and twin graph functionality for visualization
- Input from IoT devices and business systems can be integrated, connecting assets through Azure IoT Hub, Logic Apps, and REST APIs

To utilize this cloud platform, users are required to model a twin and establish its relationships by programming on the platform. The Digital Twin Definition Language (DTDL), which is an open modeling language, is employed for this purpose.

2.2 Digital Twin Definition Language

The ability to design one's own vocabulary and create a specific twin graph tailored to the user's business is a significant feature of Azure Digital Twins, facilitated by user-provided models. These models can be seen as the nouns in a description of a world. The Digital Twin Definition Language (DTDL), based on JSON-LD, is utilized to represent the models in Azure Digital Twins.
DTDL is language-agnostic, allowing it to be used by platforms other than Microsoft Azure. One example of its usage is in IoT services, such as IoT Plug and Play, where DTDL is employed for data representation.
Model Elements:

Interface: This top-level object encapsulates the entire model. All elements defined in DTDL, including Property, Telemetry, Relationship, and Component, reside within the interface.

Property: Properties represent the status of an entity and are akin to data fields in object-oriented programming languages. They have backing storage and can be accessed at any time.

Telemetry: Telemetry fields describe device sensor readings and indicate measurements or occurrences. Unlike properties, telemetry data is not stored on a digital twin; instead, it consists of a series of time-bound data events that require real-time handling.

Relationship: Relationships depict how a digital twin interacts with other digital twins. They can represent various semantic meanings, such as containment ("floor contains room"), cooling ("hvac cools room"), billing ("compressor is billed to user"), and more. Relationships are used to establish a network of interconnected entities. They can also have their own set of properties.

Component: Components allow users to construct their model interfaces by combining other interfaces if desired. For example, one can define a front Camera interface (and another component interface for the back Camera) as part of a model for a phone. Components can also be utilized for organizational purposes, grouping related properties within a model interface. In this context, each component can be viewed as a namespace or "folder" within the interface.

2.3 AWS IoT TwinMaker

As per the documentation, "AWS IoT TwinMaker is an AWS IoT service that enables the creation of operational digital twins for both physical and

digital systems. By utilizing measurements and analysis from a diverse range of real-world sensors, cameras, and enterprise applications, AWS IoT TwinMaker generates digital visualizations to facilitate monitoring and management of physical factories, buildings, or industrial plants. The real-world data collected can be leveraged to monitor operations, identify and rectify errors, and optimize overall efficiency."

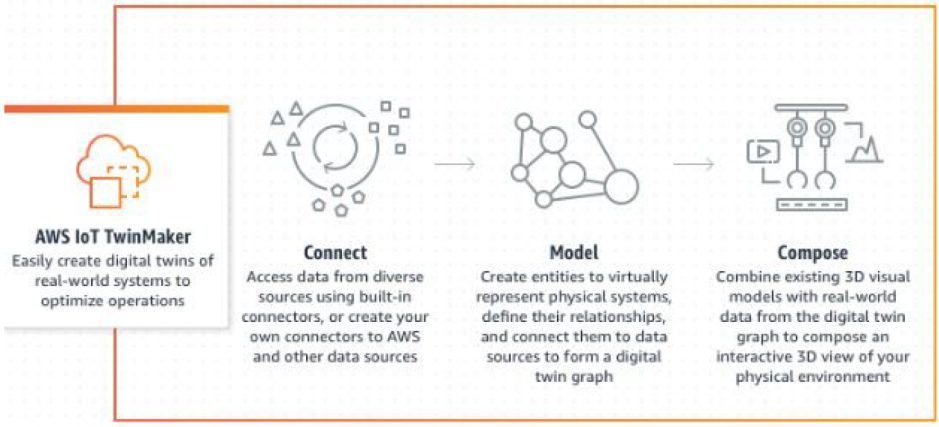

Figure 2.1: AWS IoT TwinMaker Architecture ([7])

The functioning of AWS IoT TwinMaker is similar to Azure Digital Twins, as illustrated in Figure 2.1. It involves the initial step of constructing models for physical assets, equipment, factories, or processes. Subsequently, these models are connected to the corresponding devices using Internet of Things (IoT) technology, enabling the collection of data for analysis, visualization, and the implementation of various functionalities.

2.4 Other Digital Twin Solutions

Digital Twin is an emerging field in technology, and numerous organizations are actively developing their solutions. Below, we discuss the digital twin solutions offered by three such organizations:

General Electric Digital Twin Software:
General Electric has implemented separate twins for their assets, networks, and processes. This approach has yielded significant cost savings. The benefits experienced include enhanced asset reliability, reduced risk, lower maintenance costs, improved production efficiency, and faster time-to-value.

IBM Digital Twin Exchange:
The IBM Digital Twin Exchange serves as a valuable resource for asset-intensive industries, catering to organizations that manage digital and physical assets on an enterprise scale. Users can download existing digital twins and developers have the ability to upload their own twins. However, since there is no standardized format, the interaction and structure of the twins may vary.

Siemens Digital Twin:
Siemens Digital Twin integrates technologies such as simulation, data analytics, and IoT to provide a holistic approach to digital twinning.The key components of Siemens Digital Twin include:

- **Simcenter:** Simcenter is Siemens' simulation and testing software suite that forms the foundation of their digital twin solutions. It enables engineers to create accurate virtual models of products, processes, and systems. Simcenter provides advanced simulation capabilities, such as finite element analysis (FEA), computational fluid dynamics (CFD), and multi-body dynamics (MBD), allowing engineers to simulate and optimize the performance, reliability, and efficiency of their designs.
- **Mindsphere:** Mindsphere is Siemens' cloud-based IoT platform that integrates with digital twins. It collects real-time data from physical assets and systems, which is then used to update and enhance the digital twin models. Mindsphere offers advanced analytics, machine learning, and AI capabilities to derive insights and optimize operations based on the data collected from the digital twin.
- **Tecnomatix:** It focuses on digital manufacturing, production planning, and optimization. Tecnomatix integrates with the digital twin environment, allowing manufacturers to simulate and optimize production processes, plan factory layouts, and improve overall efficiency and productivity.

2.5 Challenges

These are few challenges that the above defined Twins faces while implementing for MSMEs or in general:

1. Other than Azure digital twin and AWS IoT TwinMaker, all solutions comes as software pack and is for a single type of asset or use-case

2. Twin platform and twin services resides on single place and that is cloud, which makes even twin services unavailable when internet connectivity is lost

3. Every platform utilizes its own modelling techniques which are also open in scope, that makes these as non-standard solutions

4. These Industrial Solutions for Digital Twins are costly for the MSMEs

2.6 Refrence Literature Reviewed for Development of Machine Learning Model

During my search for work related to developing a machine learning (ML) model for tool failure predictions, I came across three valuable papers and a dataset that can enhance my skill set in model training.

In a paper by Dazhong Wu et al., they utilized force and vibration data along all three axes, as well as wear rate information from the PHM Society 2010 open challenge. Their study revealed that the Random Forest algorithm, although slower, achieved higher accuracy compared to Support Vector Machine (SVM) and Artificial Neural Network (ANN) approaches [2].

Rui Zhao et al. proposed the use of CBLSTM (Convolutional Bi-directional LSTM) to address tool wear issues using the same PHM 2010 dataset. In their approach, a Convolutional Neural Network (CNN) was employed to extract local features, followed by encoding temporal information through bi-directional LSTMs. Experimental results showcased the superior performance of CBLSTM [3].

Sungho Park et al. conducted an analysis on the effect of equipment operating conditions on tool wear using data obtained from a series of experiments with a CNC milling machine in the System-Level Manufacturing and Automation Research Testbed (SMART) at the University of Michigan. They categorized the measurement data into four groups based on the four CNC motors: x, y, and z axes, and spindle. Their findings indicated that among the three models, SVM, XGBoost, and Random Forest, employed for classification, SVM performed less effectively in terms of analysis time and accuracy compared to the other two models [5].

For our work, we will utilize the dataset from the SMART lab at the University of Michigan to train, validate, and test our model, aiming to deploy it on our Digital Twin. Our objective is to identify the best-performing model from the training process and deploy it for real-time tool condition monitoring.

Chapter 3

Earlier Versions of Digital Twin and Objectives

3.1 Digital Twin 1.0

The Digital Twin concept presented by Mr. Aniket Adsule and Mr. Saurabh Mandokar employs a client-server architecture, depicted in Figure 3.1. The communication between different clients and the central server is facilitated using the FIPA-Agent Communication Language (FIPA-ACL), which serves as the standard language for interaction.

3.1.1 Client-Server Architecture

The Digital Twin depicted in Figure 3.1 comprises various entities, including servers, machine agents, role agents, downloadable functions, and databases. These entities interact with each other using a standardized communication language.

Server:

Communication between agents is facilitated through a server. The server operates on a specific IP address and port number, known to all agents within the network. The connection between agents and the server is established using sockets and the TCP/IP network protocol.

In the server folder of the agent directories, there is a server.py file. To initiate the server, this file needs to be executed. When any agent file is triggered, the server will store its ID for communication purposes. Additionally, the server is responsible for handling incoming messages, determining if they are FIPA-compliant, and forwarding them to the designated recipient agent.

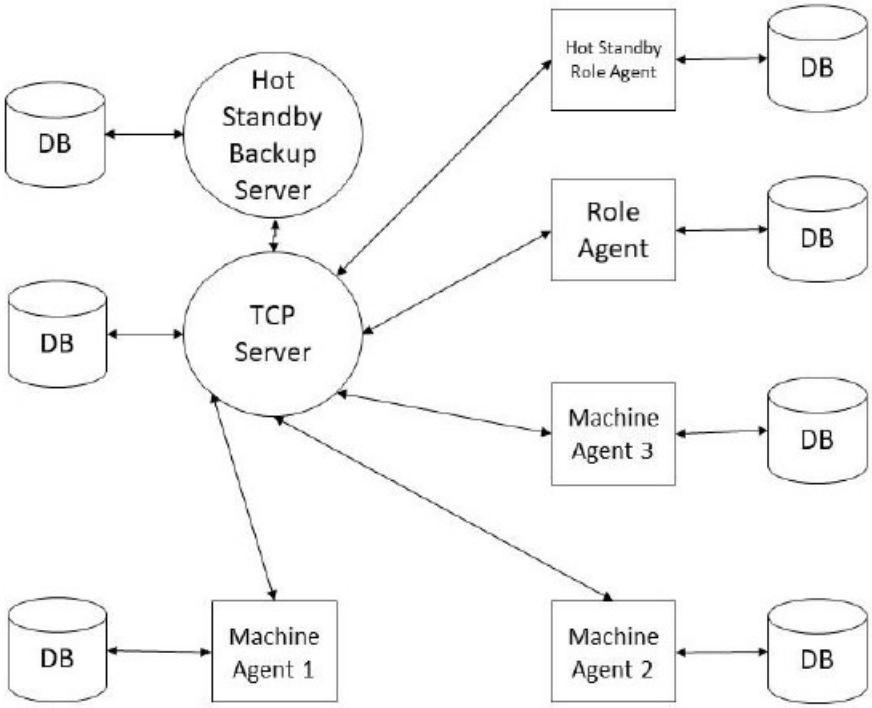

Figure 3.1: Client-Server Architecture

Machine Agent:

The Machine Agent is positioned atop the physical machine. It interacts with the physical world using sensors and actuators through the machine communication layer. This layer can be specific to the machine manufacturer. It incorporates a function that facilitates input/output operations to collect sensor data and issue commands to the physical machine. The downloadable function communicates with the physical machine utilizing these input/output operations. The architecture of the Machine Agent is illustrated in Figure 3.2. The **role agent** will also have similar architecture to the machine agent and since it is a complete algorithm to decide job sequencing and handling it will not have physical space.

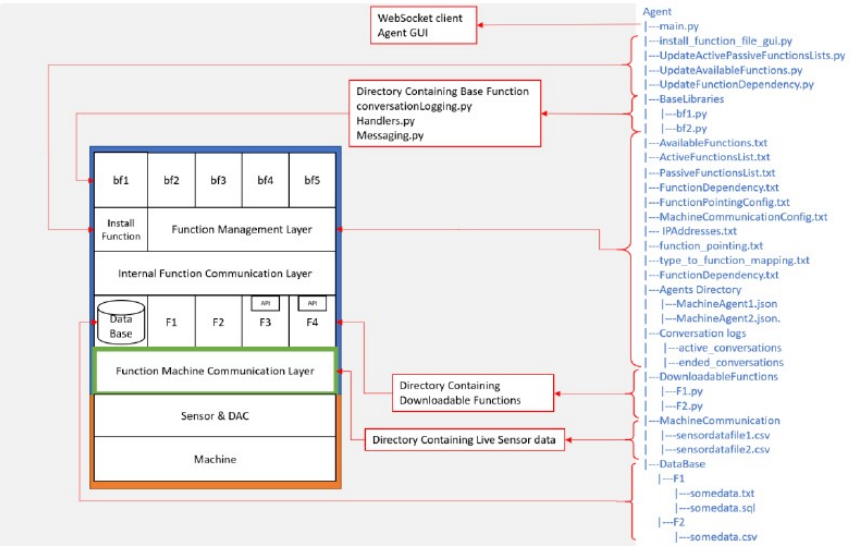

Figure 3.2: Machine Agent Architecture

3.1.2 Communication Language

The agent and server communicate with each other using FIPA-ACL as the communication language. FIPA, an IEEE Computer Society standards organization, focuses on promoting agent-based technology and ensuring interoperability with other technologies. FIPA is widely recognized for its standards that explicitly define interaction patterns and the messaging language structure, known as ACL, for autonomous decentralized agent-based architectures.

The messaging expressions within FIPA-ACL can be formulated in the content part. FIPA-ACL comprises 20 performatives, which represent communication acts. These performatives have defined semantics based on a formal semantic language called FIPA-SL.

3.1.3 Agent Functions

In a digital twin system, functions are responsible for performing specific tasks within an entity. These functions are not just algorithms but are structured and standardized to ensure interoperability in agent communication. They are installed in the corresponding agents. Since the agent-based software in this context is developed using Python scripting, an agent function refers to a Python module that contains various internal functions.

An internal function in Python is defined using the syntax "def FunctionName()". The internal Python file can be imported and called in another

file. Each function must have a unique name. Functions can be categorized based on their activation, either active or passive. They can also be categorized based on their origin, distinguishing between base functions and downloadable functions.

Base functions are essential for the agent's functionality and are always required. Downloadable functions, on the other hand, serve specific requirements of the agent. Downloadable functions can further be divided into two types: active functions and passive functions. Active functions start when the agent is initiated, while passive functions are called only when needed.

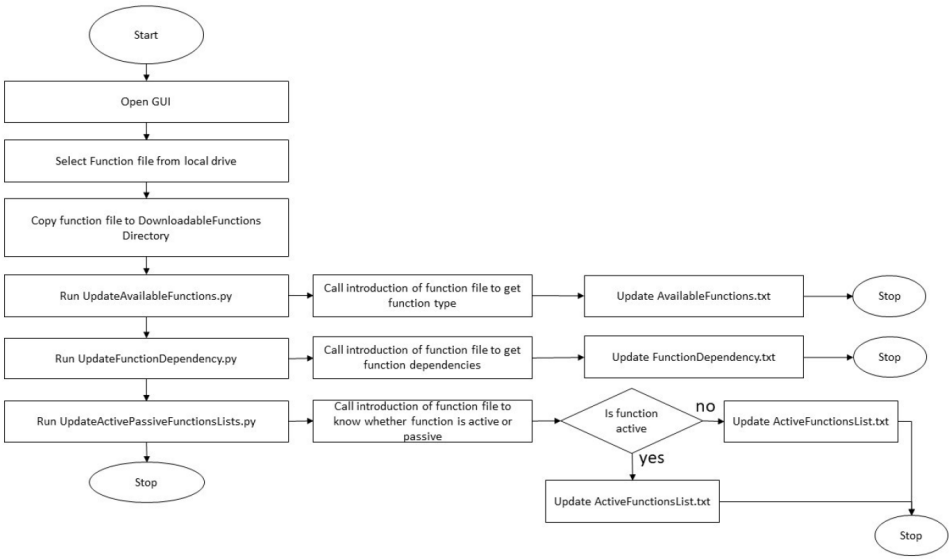

Figure 3.3: Procedure to install a function into agent

The installation process for any agent function is outlined in Figure 3.3. One of the internal functions within the agent is the introduction internal function. This function serves the purpose of providing essential information about the function, such as its type, whether it is an active or passive function, its dependencies, and its communication capabilities. The introduction internal function is primarily utilized during function installation.

In the Agent software, there are configuration files, typically in the form of .txt files, which contain information about function dependencies, function types, and the specific function that needs to be called for a particular task within the same function type. When converting an algorithm into functions, it is crucial to manage this information to ensure function interoperability.

3.1.4 Limitations

- This particular implementation of the digital twin focuses on job scheduling and sequencing tasks.
- Instead of having a centralized platform, each agent in this version of the twin incorporates a graphical user interface, which can be cumbersome to use.
- The programming of this twin restricts the data transmission between the agent and server to a maximum of 1024 bytes. As a result, it is not possible to transmit large files over the network.
- The twin has not been integrated and tested with the machine or asset aspect of the digital version.

3.2 Digital Twin 2.0

Mr. Mudit Sand developed a web-based Digital Twin Platform, addressing the limitations of Digital Twin 1.0. This platform allows customers, manufacturers, and service providers to register and access various services provided by the twin. Using an agent-based twin architecture based on the FIPA-ACL communication language, real-time data of the asset can be displayed by registering and connecting the twin to the web-based platform. Additionally, various platform-based services have been developed and integrated with the twin services.

3.2.1 Platform Overview

The platform offers scalable functionality for multiple purposes, allowing different levels of access to users such as customers, manufacturers, and service providers. The focus has primarily been on the manufacturer side, enabling registration, connection, and data retrieval from the agent-based digital twin, which utilizes the FIPA-ACL standard language for communication. The platform overview is depicted in Figure 3.4. Anyone who visits the platform must register on it to explore the functionality provided by the platform.

3.2.2 Registration of Digital Twin

The shop floor comprises numerous machines, with potentially a large number of machines of the same type. Integrating these machines with the user

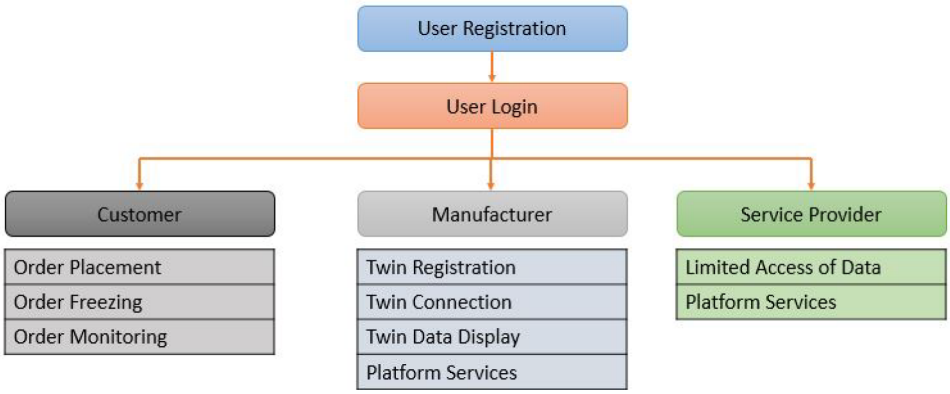

Figure 3.4: Platform Overview

entity, i.e., the twin platform, will facilitate the development of a standardized digital twin. To initiate the integration process, the twin needs to be registered on the platform, and this registration right is granted to manufacturers. The procedure for twin registration is illustrated in Figure 3.5, providing a clear understanding of the registration process. The Twin registration concept aims to establish a clear distinction between the platform and a digital twin, treating them as separate entities. This separation allows multiple physical assets to be connected to the platform through the same server. By registering these assets as Twins using a predetermined twin ID specified in the twin template, they can be seamlessly integrated with the platform.

3.2.3 Twin Connection

Once the twin registration is completed, the twin's address becomes available, allowing us to establish a connection or disconnect from it. Twin connection is particularly crucial for applications that rely on real-time data, as it enables live querying of data from the agent-based digital twin on the platform. The process flow for establishing the twin connection is illustrated in Figure 3.6.

3.2.4 Twin Data Display

To facilitate the visualization of twin data for the vertical CNC milling machine, a dashboard has been developed. The dashboard template, depicted in Figure 3.7, allows users to effectively view and analyze the relevant information associated with the machine.

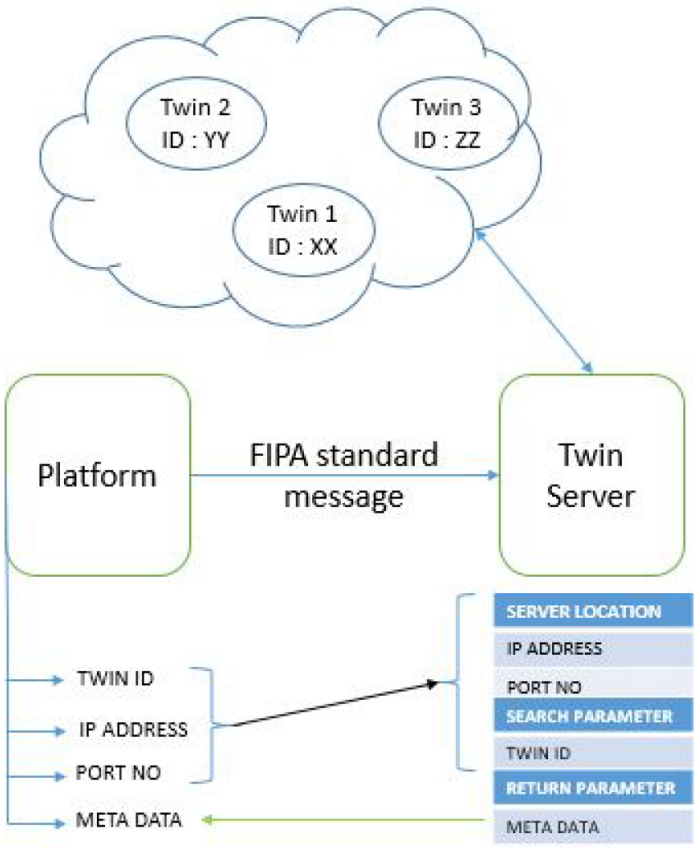

Figure 3.5: Twin Registration Process

3.2.5 Platform Services

Through the twin platform, users can utilize the services offered by the digital twin, along with the platform-specific services. To gain an understanding of the services that have been developed thus far, we can refer to Figure 3.8.

Tool Path Generation

When a customer submits an order on the platform, they are required to upload a CAD file containing the specifications of the part to be manufactured on the physical asset. This service enables the manufacturer to generate CNC code based on the downloaded CAD file. To determine the optimal tool path for machining operations, the **Google OR** solver was employed.

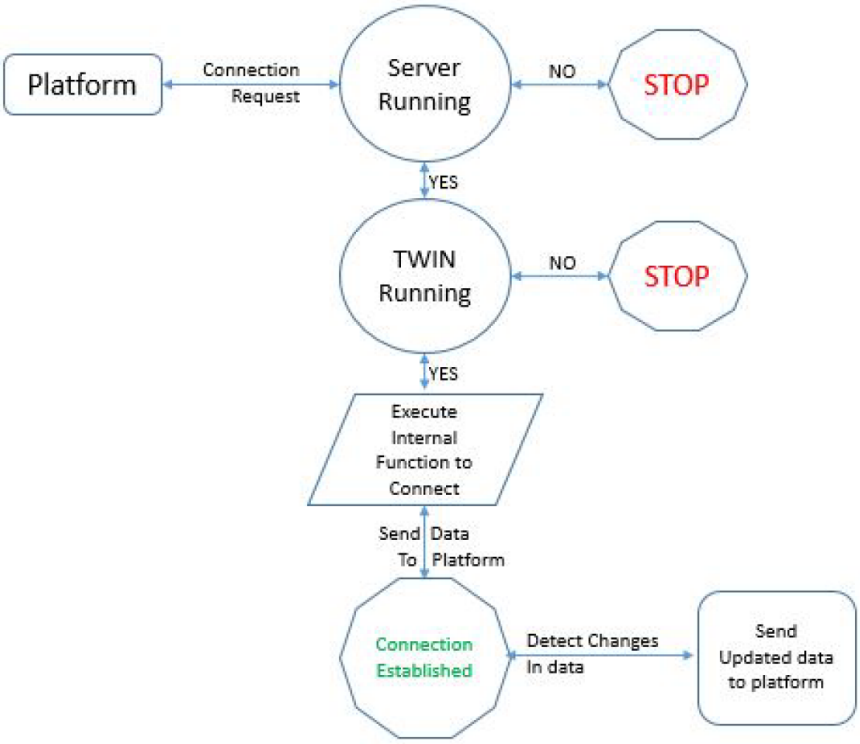

Figure 3.6: Process flow for Twin Connection

Cost Calculation

The manufacturer is responsible for assessing the manufacturing cost based on the CNC code generated. Several factors come into play when determining the cost, such as machine limitations, machine condition, and labor requirements for machine operation. Machine time is calculated by considering factors such as tool travel, the number of holes, and feed rate. The cost calculation is performed by multiplying the cost per hour by the machining time.

Querying Services

A digital twin system entity can offer multiple services, such as **inventory status, order status, and maintenance status**. To enable these services, various role agents within the agent-based digital twin need to operate in the background. The Querying service acts as a desk, where users can request information, and it will retrieve the response from the corresponding twin agent and present it to the user. The process flow for the platform to receive

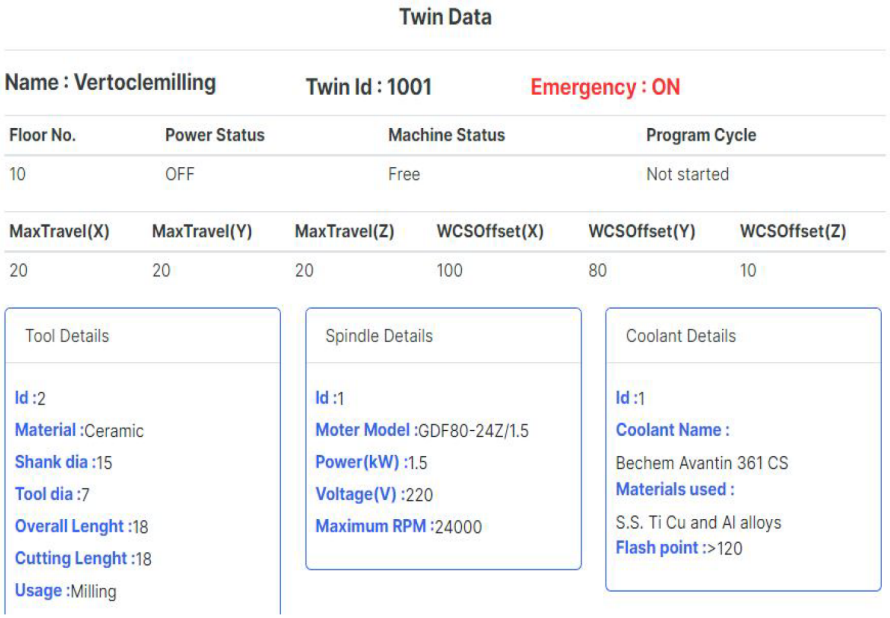

Figure 3.7: Display data dashboard

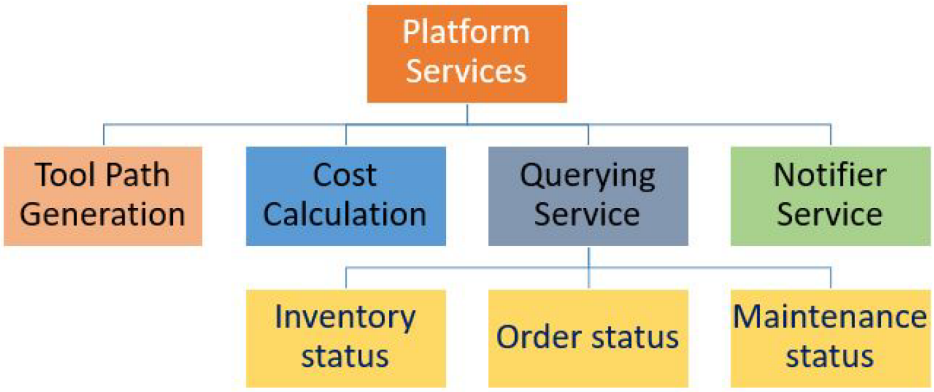

Figure 3.8: Platform Services

a response to a query is illustrated in figure 3.9. Since agent-based twins understand the FIPA agent communication language, the platform will package the message in the query interaction protocol performative. The platform acts as the sender of the message, while the recipient is the agent responsible for responding to the query. The content of the FIPA message includes the details of the query, allowing the appropriate function to be called based on the query requirements.

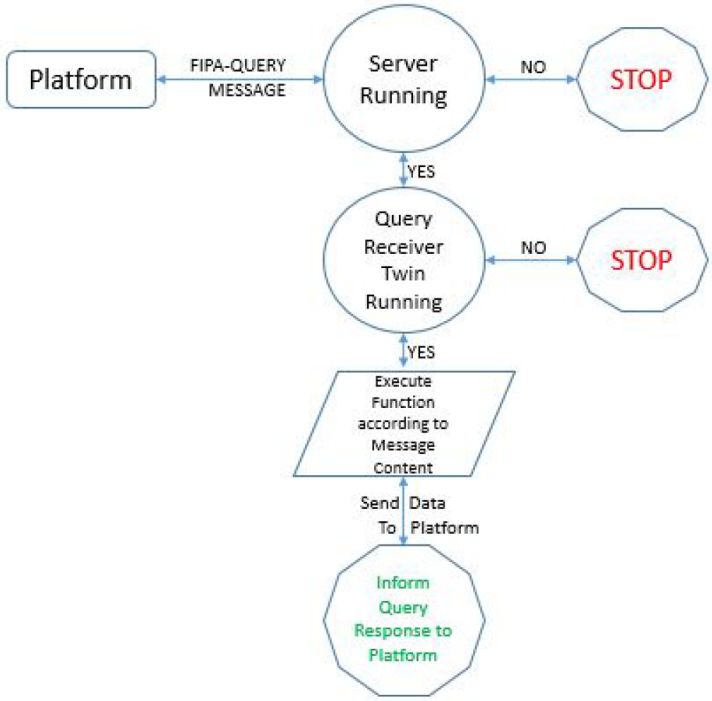

Figure 3.9: Flow of Querying Services

Once the twin server receives the FIPA message, it forwards it to the designated agent. The agent then processes the message and invokes the corresponding function to generate a response for the query.

Notifier Service

The concept behind this service is to allow not only the machine manufacturer but also third-party service providers to register on our platform. These service providers will have restricted access to the platform's services and data. The manufacturer will associate their twin with a specific service provider and grant them access to the relevant service data. This mapping process is depicted in figure 3.10.

3.2.6 Limitations

1. The deployment of the Digital Twin becomes complex due to the requirement of modifying numerous Python scripts with IP addresses, port numbers, function installation, etc., for initiating the Server, Twin Agent, various Querying Services, Notifier Service, and Platform.

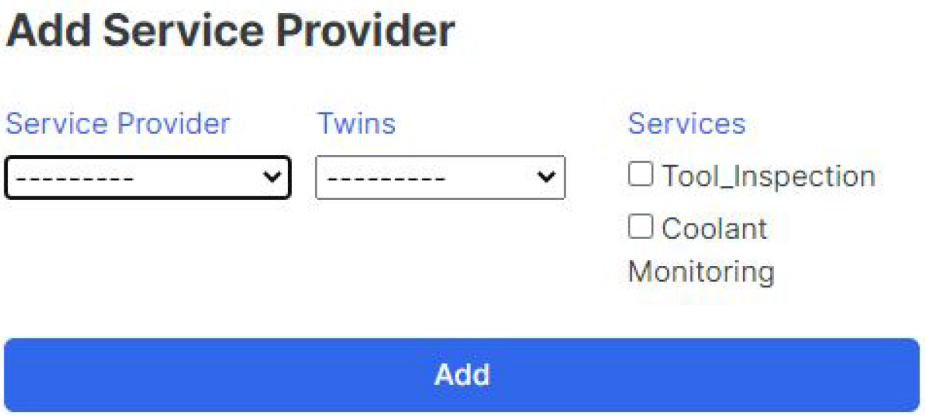

Figure 3.10: Add service provider User Interface

2. Digital Twin 2.0 was designed for manufacturing and therefore needed to be standardized according to ISO 23247 Digital Twin framework for manufacturing.

3. The architecture of the physical installation on the shop floor has not been established, including the determination of which entities will reside on which system, the number of computing devices required, and the hosting location of the Server.

4. Data acquisition from the Physical Asset is non-standard and specific to each machine, and it should conform to a recognized standard like OPC UA.

5. The acquired data is limited to the PLC of the physical assets and lacks the incorporation of external sensors, which is crucial for standardization on traditional machines.

6. The acquired data is sent for presentation on the Platform, but there is no provision for storing historical data required for Machine Learning model training.

7. The incorporation of applications and services for real-time decision-making supported by Machine Learning is necessary.

3.3 Objectives

- Establishing concept for ISO 23247 standardized Digital Twin based on earlier versions of Twin
- Establishing an Architecture for physical installation of standardized Digital Twin
- Developing a user interface for the initialization of the Server and other Agents to facilitate the deployment of the Twin
- Establishing framework for development, installation, initialization and deletion of downloadable custom function through User Interface
- Developing Application and Service Sub-system Entity with active and passive functions for Machine and Tool Condition monitoring, report generation and alarm generation
- Developing a Machine Learning model and deployment on Digital Twin by integrating it to Application and Service Sub-system Entity for tool failure prediction

Chapter 4

Digital Twin 3.0

4.1 Minimum Viable Digital Twin

4.1.1 Digital Twin Roles

In a minimum viable digital twin following are the **Digital Twin Roles** that need to be performed as per instructions from a User Interface:

1. **Respond:** A digital twin can gather real-time data from various sources such as sensors and IoT devices to create a detailed and up-to-date representation of the physical environment. These data can be shared on an Interface as a response to a user query.

2. **Display:** In the context of displaying information, a digital twin can be utilized to provide a comprehensive and interactive representation of data, enhancing the understanding and visualization of complex information. A digital twin can integrate real-time data from various sources and present it in a visual format. This allows users to monitor and interpret information more effectively. Digital twins can also provide immersive and interactive interfaces, such as 3D models or augmented reality (AR) applications. Digital twins can integrate data from various sources and systems, providing a unified and contextualized view of the information. For eg. in a manufacturing facility, a digital twin can display data from external sensors, equipment(machine PLC), and production databases(Enterprise Resource Planning softwares), offering a comprehensive overview of the entire operation.

3. **Alarms:** In the context of alarm generation, a digital twin can play a significant role in detecting and generating alarms based on predefined conditions or anomalies within a system. A digital twin continuously

monitors the data collected from sensors, devices, or systems in real-time. By comparing the real-time data with predefined thresholds or patterns, the digital twin can identify abnormal conditions or deviations from expected behaviour. When such conditions are detected, it can generate alarms to alert relevant stakeholders. A digital twin can also be programmed to generate alarms based on specific conditions or events.

4. **Recommendations:** In the context of generating recommendations, a digital twin can play a significant role in providing personalized and context-aware suggestions based on the data and knowledge it possesses. A digital twin leverages real-time and historical data to gain insights into user behavior, preferences, and patterns. By analyzing this data, the digital twin can identify trends, correlations, and relationships that can be used to generate relevant recommendations.

5. **Activity Support:** In the context of activity support, a digital twin can play a valuable role in assisting and enhancing various activities by providing real-time guidance, feedback, and contextual information. A digital twin can provide step-by-step instructions and guidance for performing complex tasks or activities.

6. **Decision Support:** A digital twin can serve as a decision support tool by providing recommendations in complex scenarios. By simulating different options and evaluating their potential outcomes, the digital twin can offer insights and suggestions to guide decision-making.

7. **Control Action:** A digital twin continuously monitors real-time data from sensors and systems, collecting information about the current state of the physical environment. By analyzing this data, the digital twin can identify deviations, anomalies, or patterns that require control action. By leveraging historical data and computational models, a digital twin can make predictions about the future behavior of the system. It can simulate different scenarios and forecast potential outcomes based on various control actions.

4.1.2 Digital Twin Modules

In order to perform the above role we need our minimum viable digital twin to have some standard and customized modules which facilitates its operation and working.

Standard Modules

1. **Description:** The Description module in a digital twin provides functionalities that enable the comprehensive representation and description of the physical counterpart of the twin. It captures various aspects of the physical system or object and forms the basis for understanding and interacting with the digital twin. Description module enables a digital twin to accurately represent, describe, and understand the physical counterpart.

2. **Monitoring:** The Monitoring module in a digital twin provides functionalities that enable the real-time monitoring and data acquisition from the physical counterpart. It captures and processes data from sensors and other sources, allowing for continuous observation and analysis of the physical system or object. Monitoring module enables a digital twin to continuously monitor and analyze the physical counterpart, providing real-time insights and facilitating timely decision-making. It forms a critical component in maintaining an accurate and up-to-date representation of the physical system or object within the digital twin environment.

3. **Communication:** The Communication module in a digital twin provides functionalities that facilitate communication and interaction between the digital twin and external entities. It enables seamless data exchange, collaboration, and integration with other systems or stakeholders. This functionality involves the integration and exchange of data between the digital twin and external systems, such as IoT platforms, enterprise software, or cloud services. It includes functionalities for data ingestion, transformation, and synchronization, ensuring that relevant data is shared within the digital twin.This functionality enables remote access and control of the digital twin. It includes functionalities such as secure remote login, authentication, and access control mechanisms.Communication module enables effective communication, collaboration, data exchange, and integration with external systems or stakeholders. It ensures that the digital twin operates in a connected and interoperable manner facilitating seamless interactions.

4. **Visualization:** The Visualization module in a digital twin provides functionalities that enable the visualization and representation of the digital twin and its associated data. It focuses on presenting information in a visual and intuitive manner, allowing users to understand and interact with the digital twin effectively. The Visualization module in-

cludes functionalities for creating interactive dashboards that present key performance indicators (KPIs), real-time data, and analytics related to the digital twin. Dashboards allow users to monitor the system's behavior, track metrics, and gain insights at a glance.Visualization module enables users to visualize, explore, and interact with the digital twin effectively.

5. **Digital Twin Management:** The Digital Twin Management module facilitates the installation and removal of user defined downloadable custom functionalities, enabling interoperability across multiple machine types. This functionality ensures that the Digital Twin is not confined to a specific machine and promotes standardization.

Customized Modules

1. **Descriptive Analytics:** The Descriptive Analytics module in a digital twin provides functionalities that focus on analyzing historical data and providing insights into the past performance and behavior of the physical counterpart. It enables users to gain a deeper understanding of the system, identify patterns, trends, and anomalies, and make informed decisions based on historical data. The Descriptive Analytics module includes functionalities for performing statistical analysis on the historical data.

2. **Predictive Analytics:** The Predictive Analytics module in a digital twin provides functionalities that focus on analyzing historical data and using it to make predictions or forecasts about future behavior or outcomes of the physical counterpart. It leverages advanced analytics techniques and algorithms to identify patterns, trends, and relationships in the data, enabling users to anticipate potential issues, optimize performance, and make proactive decisions. The Predictive Analytics module includes functionalities for developing and training predictive models using the historical data. It involves selecting appropriate algorithms, such as regression, time series analysis, or machine learning algorithms, and applying them to the data.

3. **Prescriptive Analytics:** The Prescriptive Analytics module in a digital twin provides functionalities that focus on optimizing system behavior, recommending actions, and providing decision support based on predictive and historical data. It goes beyond predicting future outcomes and aims to identify the best course of action to achieve desired objectives, considering constraints and optimization criteria. It assists

in identifying the best course of action, considering constraints and optimization criteria, and supports data-driven decision-making processes to achieve desired objectives efficiently and effectively.

4. **Asset Control:** The Asset Control module in a digital twin provides functionalities that focus on controlling and managing the physical assets or components of the system represented by the digital twin. It enables users to monitor, control, and optimize the behavior and performance of assets in real-time.

4.2 Architecture for ISO 23247 Digital Twin

In ISO 23247 Digital Twin for Manufacturing reference architecture there are mainly three layers : **Data Collection and Device Control Entity**, **Digital Twin Core Entity** and **Digital Twin User Entity**. All of these three layers along with another **Observable Manufacturing Elements** layer for understanding have been conceptualized for our Digital Twin. This can be visualized through the figure 4.1 where the architecture for a four-machine shop floor is presented. The earlier version of Digital Twin was

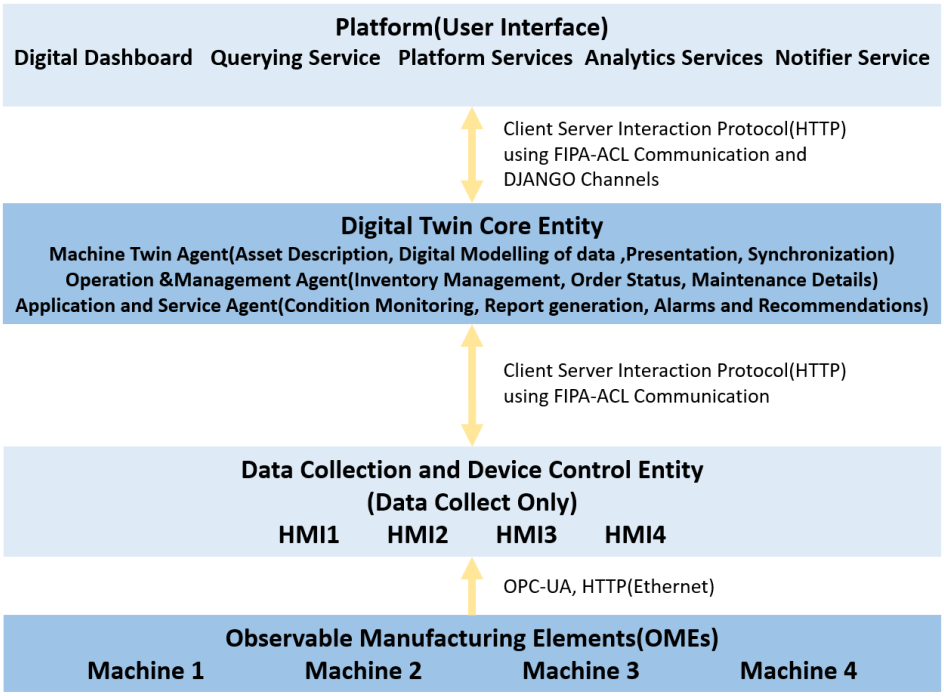

Figure 4.1: Concept Digital Twin 3.0

only collecting data and presenting it in real-time for a single machine and through this architecture, we can incorporate multiple machines along with their Twin Agents.
ISO 23247 Digital Twin has been developed based on the following four procedural activities:

4.2.1 Observe

This is the first activity we need to identify and establish while conceptualizing a Digital Twin. This activity is relevant to **Observable Manufacturing Elements(OMEs)** which is although not directly being the part of framework but still important for understanding. In this layer we **identify physical assets** on shop floor that **can be modelled**. It is in this layer we need to identify what parameters are critical to Twin and whether we are getting it directly from the PLC of the machine or we need to incorporate external sensors.

4.2.2 Collect

This activity is relevant to **Data Collection and Device Control Entity**. In this layer, we will install **Personal Computers(HMI)** on machines to enable machines to configure themselves in digital entities based on PLC and sensor data installed on them. This layer is only meant for Data Collection as of now and can be connected to physical assets via OPC-UA, HTTP(Ethernet), or IoT gateways. As of now in our Twin, we are getting data via the a python script called socketdll file which listens to socket communication happening between the executable application of the machine and PLC.

4.2.3 Model

This third activity is core to the Digital Twin framework, it is here where the data collected from DCDCE is modeled for representation onto the User Entity. It is about developing capabilities be it the modeling of data, presentation of data, analysis of data, or reporting of results. It is here where we are conceptualizing to develop functionalities that will increase the usefulness of our Twin way beyond just collecting and representing real-time data.
FIPA-ACL enabled **Machine Twin Agent** need to be installed on the HMI computer of each machine for connection to the Digital Twin Core Entity

where the central server is hosted. Also, this Machine Twin Agent will register itself on the Platform which is Digital Twin User Entity in our architecture.
We can also install custom functions for data pre-processing in Machine Twin before relaying the data for synchronization and presentation from the Machine Communication directory.
In the figure 4.2 , we can observe the Machine Twin directory which will collect data from the Socketdll file and store data temporarily in the Machine Communication folder as a JSON file. The functionalities which have been pre-installed in Machine Twin Agent are **asset description**, **digital modelling** of data, **presentation** of data on Platform(UI) and real time **synchronization**

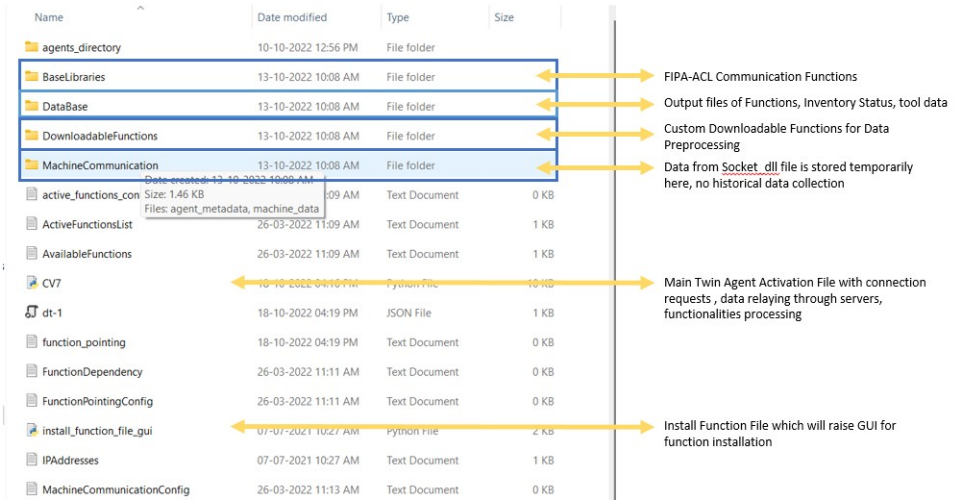

Figure 4.2: Machine Twin Directory

Apart from Machine Twin Agent there are other FIPA-ACL enabled role agents also in Digital Twin Core Entity which are installed on HMI computer and processes data from Machine Communication folder based on request from User Interface. The first such role agent is **Operation and Management** which provides functionalities for **inventory management, order status and maintenance details** of the machine. The other role agent is **Application and Service** which provides functionalities for **condition monitoring, report generation, alarms and recommendations**
If we compare our Twin with reference architecture Sub-System entities, in **Machine Twin Sub-system entity** following functional entities are present:

- **Asset Description FE :** In our current Twin, this functionality has been achieved as we are capable of extracting useful information digitally upon registration and connection of Twin on Platform which describes the type of machine modelled as Digital Twin, its machine limits and other useful informations

- **Digital Modelling FE :** The data collected is converted into JSON script which is transferred to Platform via TCP/IP protocol using FIPA-ACL framework and this data is then modelled for display onto dashboard

- **Presentation FE :** In our current Twin, we can present digital data on the dashboard of the Digital Twin User Entity(Platform) as charts and digital boards

- **Synchronization FE :** In the current version, we are not first collecting data and then relaying packets of data instead we are transferring data for presentation as and when it is received thus it is in real-time synchronization

In **Operation and Management Sub-system entity** we have developed a Role Agent named **Operation and Management Agent**. The functional entities in this agent are listed below:

- **Inventory Management FE :** In this functionality based on request from platform i.e. user entity we provide purchase requirements, available raw materials, work in progress inventory and finished goods based on data which can be received from Enterprise Resource Planning software.

- **Order Status FE :** In this functionality user can query regarding the due date of the job, route of the job that is planned to finish the work on the raw material to convert it into the finished product, current location of the job.

- **Maintenance Details FE :** In this functionality user can query regarding the next maintenance schedule of the twin, user can see the date and time of maintenance, type of maintenance and the estimated production delay due to this exercise.

In **Application and Service Sub-system entity** we have developed a Role Agent named **Application and Service Agent** It will be the prime focus for our version of Twin. The main reason for bringing it to prime focus can be understood from its functional entities:

- **Condition Monitoring FE :** We have developed custom functions for monitoring the condition of machine and tool based on the data from Machine Communication directory where PLC as well as external sensors data is being collected. These functions will store their analysis in Database directory from where it can be accessed by other functions for further analysis based on request from user interface(Platform).
- **Reporting FE :** Based on the analysis of condition monitoring data we can develop reports using custom functions and display them on the User Entity for action.
- **Alarms FE :** Based on reports generated we have developed functionalities to display alarms for machine and tool conditions on user entity(Platform).
- **Recommendations FE :** Reports generated for machine and tool condition has also been utilized to show recommendations for necessary actions on user entity

4.2.4 Learn and Act

This final activity is relevant to Digital Twin User Entity which in our case is the Platform. A PC with decent computing is required to act as a central server for client-server interaction between HMIs and hosting Platform. The platform is a separate entity and helps in the visualization of real-time data, registration of the user, registration of Twin Agent for interaction, and accessing various Platform Services. The platform in itself is a client agent of the main server and is hosted on the Main PC on which the server resides. The platform acts as a link between the User and Digital Twin Core Entity. The services offered by Platform have already been well-established in earlier versions of the Twin. In this version we have utilized the Platform as source to trigger user requests for various analytical functionalities of Application and Service Agent.

4.3 Digital Twin 3.0 Schematic

Digital Twin 3.0 is also based on Client-Server Architecture as shown in the figure 4.3. We can observe that in our twin there is a central server hosted on the Main PC which acts as a medium of Information Exchange among several Client Twin Agents depicted in the figure 4.3 as Twin Agent, Application and Service Agent, Operation and Management Agent and Platform.

The Server and all these twins are activated using interfaces which will be discussed later in detail. In this schematic we can observe that each twin

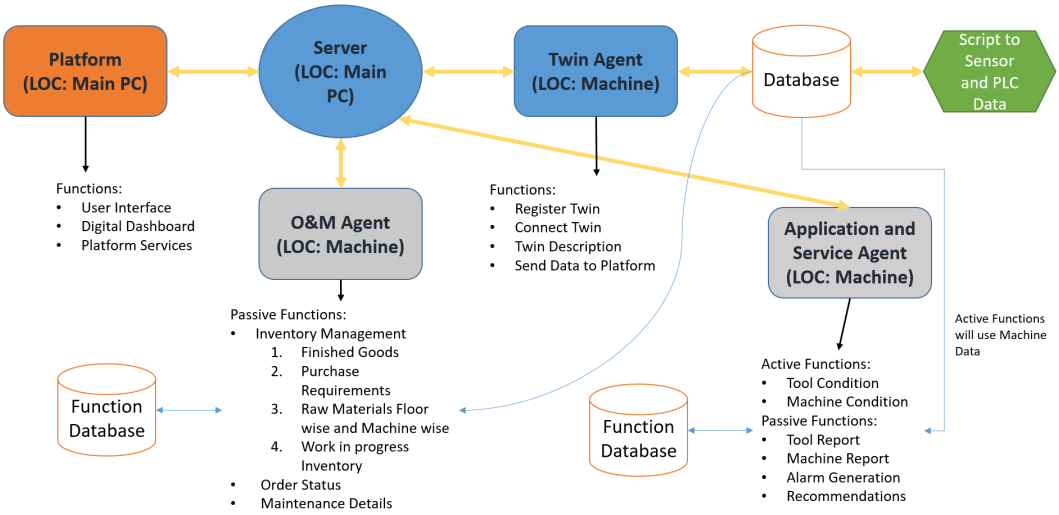

Figure 4.3: Client-Server Schematic of Digital Twin 3.0 along with their Locations

has certain functionalities which provides them capabilities to perform tasks based on request from User Interface i.e. Platform in our case. We will discuss about these functionalities in detail later sections. Also we can observe that in our Digital Twin 3.0 we have developed a python script which listens to the communication happening between PLC of the machine and its executable application via sockets. This scripts provide us with machine data which gets updated in Twin Agent database every second.We can also observe that the functions in role agents have database of their own where they store their analysis of machine data which can be accessed based on trigger from Platform.
In this schematic Twin Agent is a physical agent on top of the machine providing it digital identity for registration, connection, asset description and sending live data to Platform. Operation and Management is a role Agent which responds to the FIPA query from Platform seeking information regarding inventory management, order status and maintenance details. Application and Service is also a role agent which performs machine condition monitoring based on vibration data and tool condition monitoring predicting tool failure based on Machine Learning model. It also responds to various analytics related FIPA query being asked from Platform like tool report, machine report, alarm generation and recommendations.

4.4 User Defined Custom Functions

Every agent possesses its own set of agent functions, which encompasses all the functions utilized by the agent. This set includes base functions, as well as active functions, passive functions, and downloadable functions.An agent function refers to a ".py" file that contains various internal Python functions. These internal Python functions are defined using the syntax: "def FunctionName():". It is important that each function within the agent has a unique name. These functions can be categorized based on their activation, either as active or passive. Additionally, they can be classified based on their origin, such as whether they are base functions or downloadable functions.

4.4.1 Base Functions

Base functions refer to essential functions necessary for the agent to operate in its most fundamental state. These functions are inherent to the agent upon installation, serving a role analogous to the operating system (OS) of a PC or laptop, which is crucial for running other programs on the device. The base functions encompass communication functionality and the ability to invoke other functions based on communication requirements. However, they do not encompass functions that provide the agent with specific objectives. For instance, a base function could involve managing messages, creating FIPA messages, serializing and deserializing FIPA messages, defining send and receive functions for the agent, maintaining subscription topics, controlling conversations, and more.

4.4.2 Downloadable Functions

The agent's installation does not include these functions by default; instead, the user must download them from a repository and install them into the agent. These functions serve as objectives for the agent. As mentioned previously, the agent can accommodate multiple downloaded functions, each contributing to a distinct objective. For instance, a downloaded function could be machine condition monitoring, which conducts statistical analysis on machine axis vibration data and predicts maintenance requirements. These functions are user-defined, and we have established a framework for their development to ensure seamless integration into the agent.

4.4.3 Active Functions

Active functions refer to functions that operate continuously, performing specific dedicated tasks. When an agent is initiated, each active function obtains a separate thread on the processor and runs continuously on that thread. These active functions persistently execute as long as the agent remains active. In the event of an error causing one active function to stop, the cessation of that specific active function will not impact other active functions or any other functions, and they will continue to operate unaffected. Once initiated, active functions cannot be accessed directly. Their primary role is to identify triggers and initiate interactions, without handling the replies to these interactions. To address this situation, another category of functions, known as passive functions, exists.

Structure of an Active Function

The provided diagram 4.4 illustrates a suggested structure for an active downloadable function, consisting primarily of three internal functions: the introduction internal function, the active internal function, and the callable internal function. The introduction internal function serves the purpose of

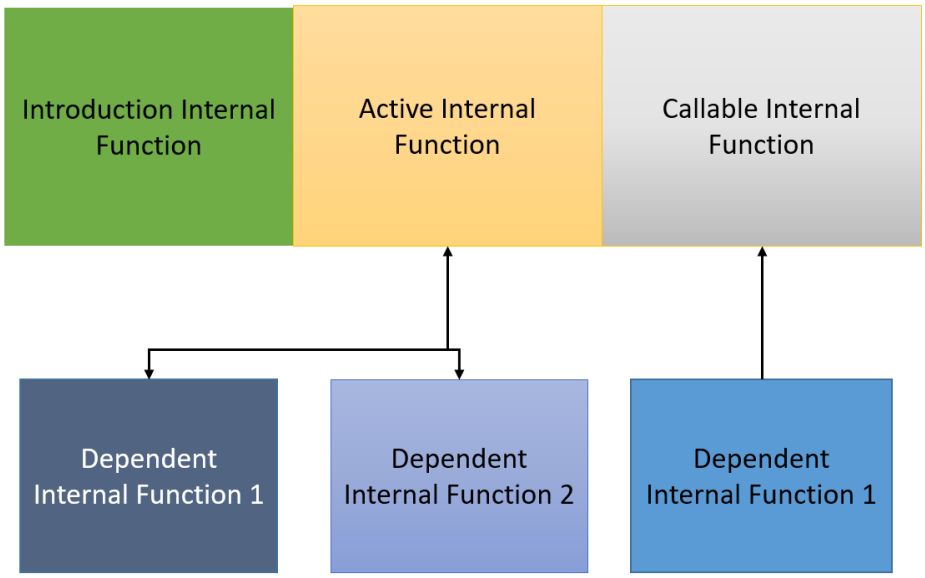

Figure 4.4: Structure of an Active function

offering essential information about the function when called upon. This includes details such as its type, active or passive, dependencies, and communication capabilities(performative referential expression). Its primary usage

occurs during the installation of the function.
For an active downloadable function, its designated task is to run continuously. During agent initialization, the main program file executes the active internal function of the active downloadable function on a separate thread. This active internal function persistently performs its intended job, such as machine condition monitoring. In the case of machine condition monitoring, the active internal function continuously reads real-time sensor data through the machine communication folder and generates output that it stores in its own database.
When other functions within the agent require information on the current health of the tool, they can invoke the callable internal function. The callable internal function inherits the characteristics of a passive function. It retrieves the current tool status from the database maintained by the active internal function and returns the corresponding value.
Once the installation of downloadable active function is successful, it updates a configuration file named ActiveFunctionsList.txt as shown in figure 4.5 with active function name and digit 0 in front of it. If we want the installed active function to run upon agent initialization we need to change the digit to 1 from the machine HMI interface which will be discussed later.

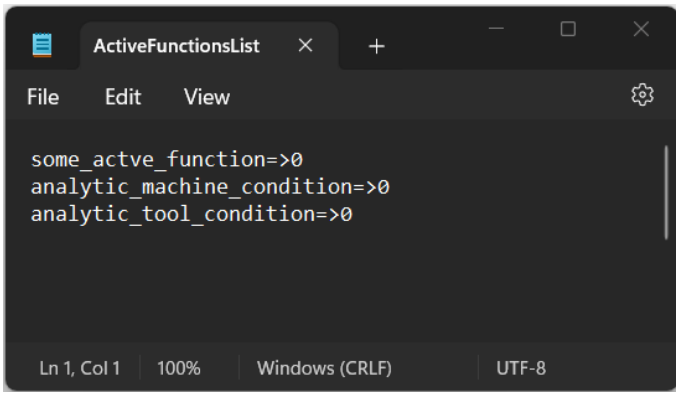

Figure 4.5: ActiveFunctionsList.txt Structure

Machine Condition

We have developed an active function for machine condition monitoring in application and service agent as shown in 4.6. This active function can be run on separate thread upon agent initialization and actively updates the analysis in function database. This function performs a two sample t-test on machine vibration data coming from sensors by splitting data into 80:20

ratio. It treats 80 percent data as historical and 20 percent as recent. It calculates the p-value of test statistic assuming a critical amplitude factor which is 2.5 in our case. The data which we have used in our analysis has been generated randomly on Excel for validation purpose.

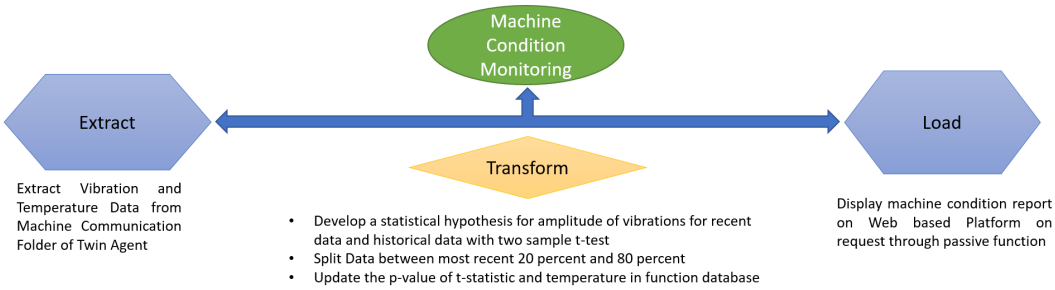

Figure 4.6: Schematic of Machine Condition Monitoring active function

Tool Condition

Another active function which have developed predicts whether the tool has been worn or unworn as shown in figure 4.7. We have used data from SMART lab of University of Michigan for development and deployment of Machine Learning Model which has been discussed in detail in upcoming sections. We have also used the same data for validation purpose.

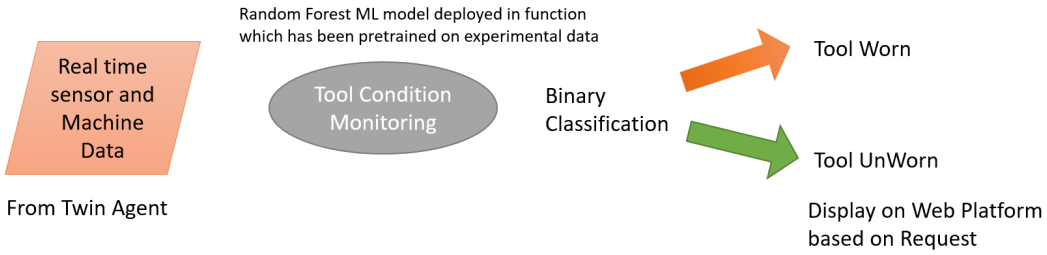

Figure 4.7: Schematic of Tool Condition Monitoring active function

4.4.4 Passive Functions

These functions belong to a category that exclusively responds when invoked. They are designed to accept input in a specific format and generate output in a predetermined format. These functions are responsible for analyzing both the twin agent database and the function database. They specifically

respond to FIPA queries triggered from the Platform that are relevant to their purpose.

Structure of Passive Function

The provided diagram, 4.8, presents a suggested structure for a passive downloadable function. This function consists of multiple internal functions. To begin with, the required libraries must be imported at the start of the function file. Among these internal functions, there is the introduction internal

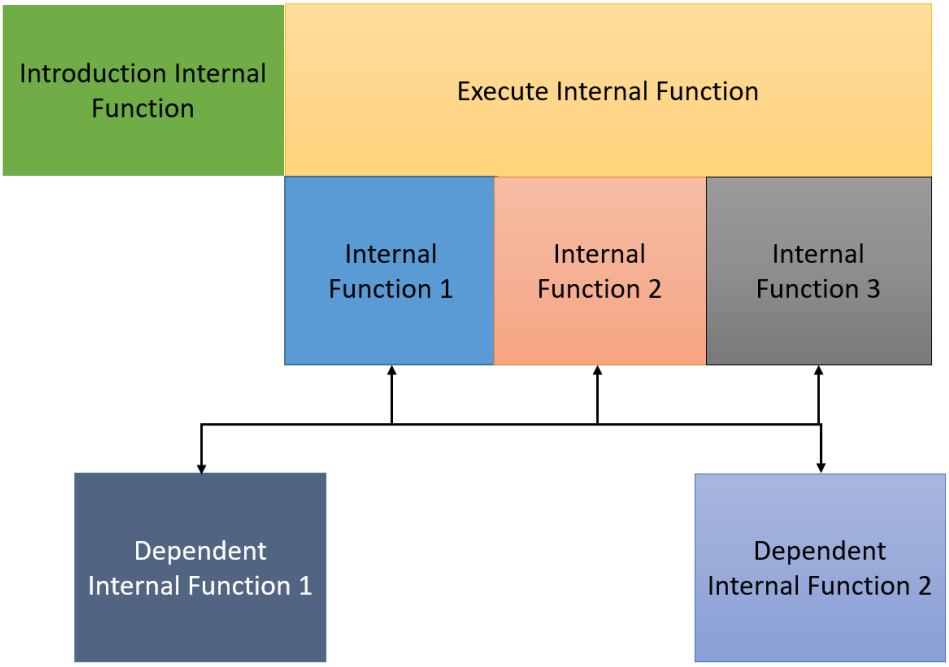

Figure 4.8: Structure of Passive Function

function. This function serves the purpose of providing essential information about the function when called upon. It includes details such as the function's type, active or passive, its dependencies, communication capabilities(performative referential expression). The introduction internal function primarily comes into play during the function installation process.
In passive downloadable function's execute internal function is called upon when there is inter agent communication. In current structure a FIPA query with query-ref performative and referential expression **"request_ machine_ report"** is triggered from platform seeking machine report from analytic and service agent. The analytic and service agent based on the referential expression will import the required downloadable passive function by comparing it

in another configuration file named PassiveFunctionsList.txt as shown in figure 4.9

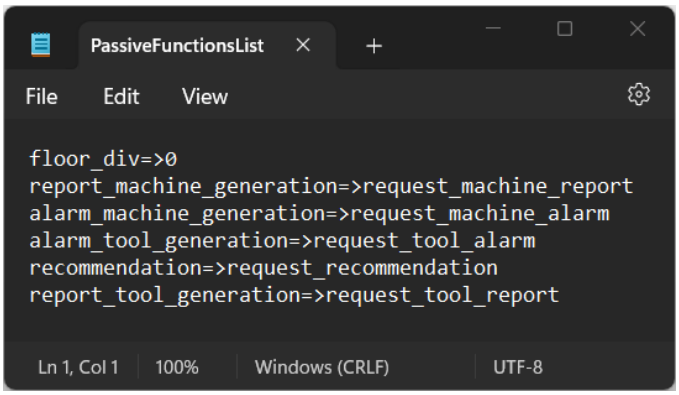

Figure 4.9: PassiveFunctionsList.txt Structure

Machine Report

The first passive function developed by us for installation in Application and Service agent is for machine report generation. This function is dependent on machine condition monitoring active function as it collects the data generated by it through callable internal function. A passive function is coded into the target agent in such a way that it responds to the specified referential expression of FIPA query generated from User Interface(Platform). The step by step functioning of passive function can be understood as below:

- First a FIPA request is generated from platform with agent id and query-ref performative with an expression
- This FIPA query is received by target agent through server and agent identifies which downloadable function to be imported for analysis based on query-ref expression
- Once the function is imported it performs its analysis and send a json reply in response to the query
- This json reply is being recieved by Platfrom(UI) and displayed on dashboard

In the machine report function the response to FIPA query is as shown in figure 4.10.

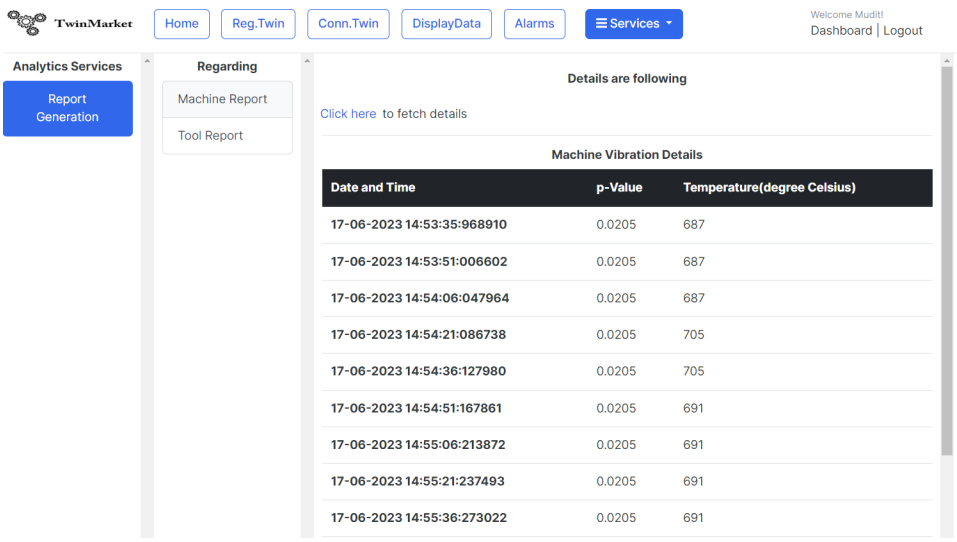

Figure 4.10: Machine Report

Tool Report

The other passive function which we developed and installed on our Application and Service agent is tool report. This function is dependent on tool condition monitoring active function as it collects the data generated by it through callable internal function. It also generates a tool report in response to the trigger generated from Platfrom as shown in figure 4.11. The step by step working is similar to machine report passive function.

Alarm Generation

We have developed two passive functions for alarm generation, one for machine based alarms and other is for tool based alarms. In machine based alarm generation we calculated average of recent 10 percent machining temperature data and if this average is above predefined critical temperature which is 650 degree celsius in our case, an alarm ON json reply is generated in response to the FIPA query. This response is then displayed on dashboard as shown in figure 4.12.
In tool based alarm generation we check whether the latest tool condition report is worn or unworn and generate corresponding json reply for alaram FIPA query. This repsonse is then shown on platform as in figure 4.13. The step by step working is similar to other passive functions.

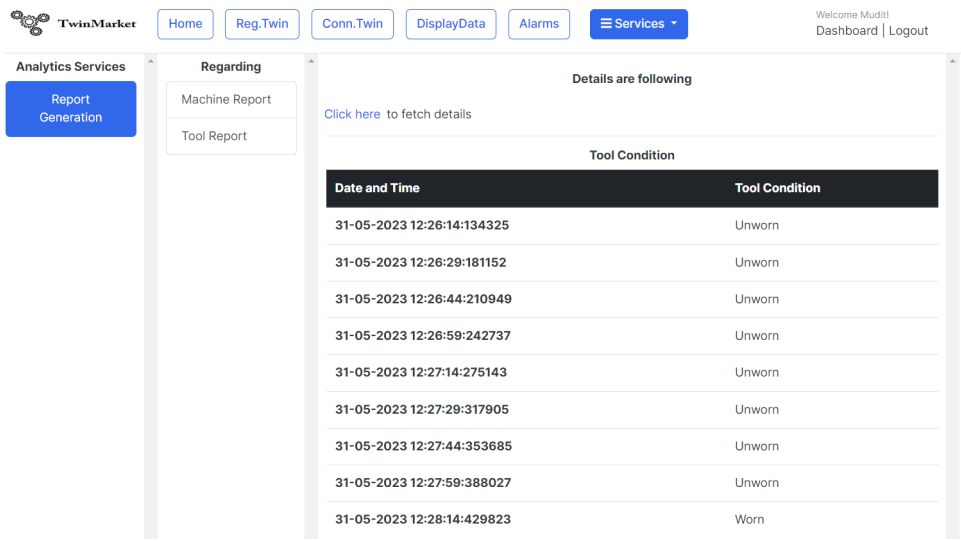

Figure 4.11: Tool Report

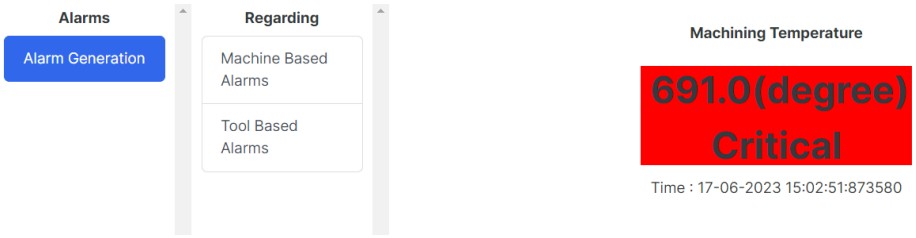

Figure 4.12: Machine Based Alarm

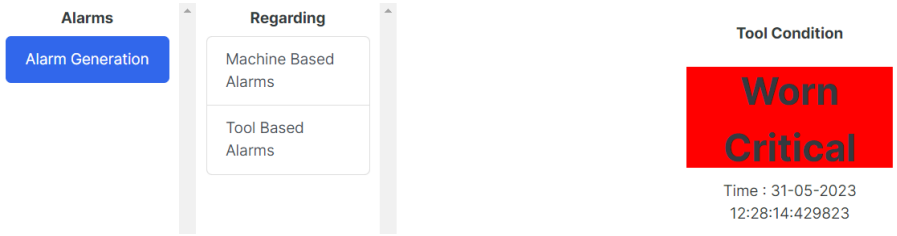

Figure 4.13: Tool Based Alarm

Recommendations

The recommendation function is very much similar to alarm generation. It recommends to turn on coolant if recent mean machining temperature is above the predefined cut-off which in our case is 450 degree celsius. Also if tool is worn it will recommend to replace the tool as shown in figure 4.14. The step by step working is similar to other passive functions.

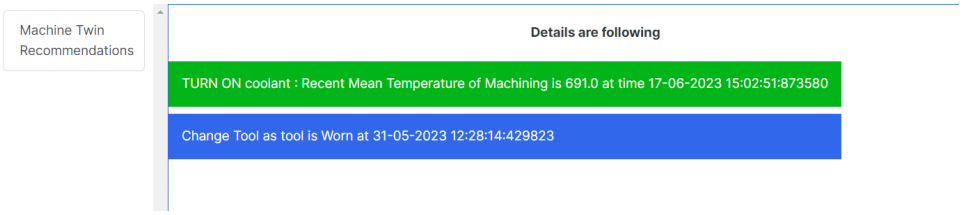

Figure 4.14: Recommendations

4.5 Chronological Operation of Twin and User Interfaces

In earlier versions of Digital Twin, we need to open python scripts and write Port No. and IP Addresses for server and other client agents. All these manual running of scripts are hindrances in the path of deployment of Twin as on the shop floor we cannot expect Operators to edit python scripts. To solve this problem we have developed two types of User Interface(UI), one is **Server UI** and the other is **Machine HMI**.

4.5.1 Server UI

The main purpose of Server UI is to enable the hosting of the server on the Main PC graphically through User Interaction. We can see from the figure 4.15 that in our UI there is a box to enter Port No. and IP Address. This Port No. and IP Address will be written in the server_ ip.txt file in the Main Directory of the Main PC. Upon clicking on Activate Server, a python script named "server.py" in Server folder which hosts a socket based TCP server on the Main PC using port no. and IP address data from the file. There is a button to launch the Platform and a logs window to depict success messages for our interaction with the interface.

4.5.2 Machine HMI

The main purpose of this UI is to enable Machine Twin Agent to connect to the TCP server hosted on main PC. Along with this initialization of Application and Service, Operation and Management, and Notifier Agent is also done using this user interface. This interface also has establish connection button which activates the **socketdll** file for transfer of data to the Machine Communication folder of Twin Agent and then to Platform and Application and Service Agent upon request. Like Server UI it too has boxes to enter Port No. and IP addresses and buttons to enable agents to establish connection

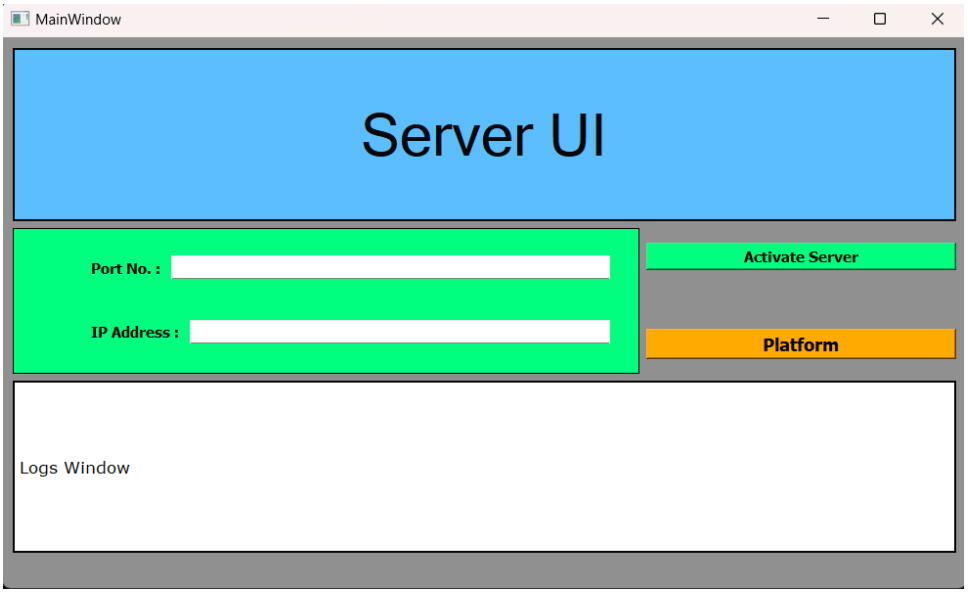

Figure 4.15: Server UI

to the server as shown in the figure 4.16. The logs window depicts where our interaction with interface is successful. The menu bar functionalities of this

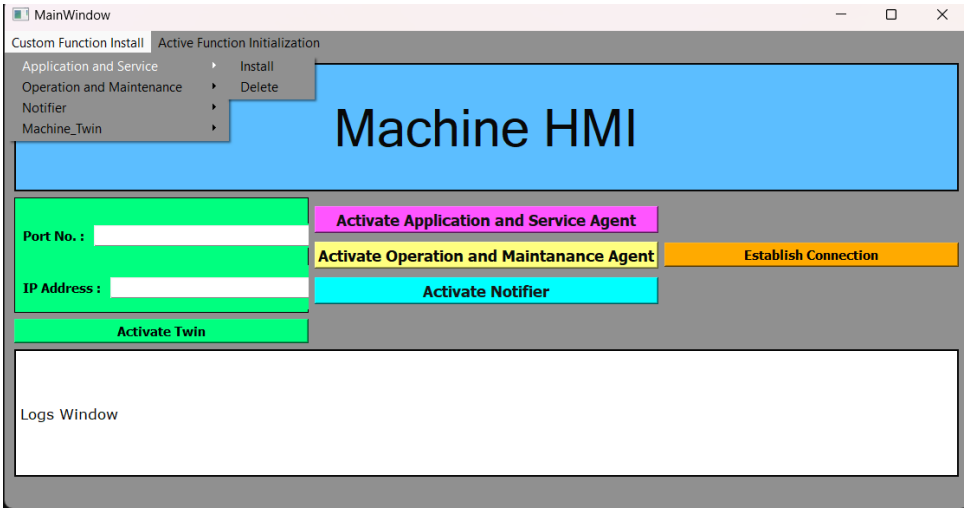

Figure 4.16: Machine HMI

interface are following:

Custom Function Install

As discussed earlier in order to standardize our twin we need to enable our Twin to incorporate custom downloadable functions. These functions can be active or passive depending on its usability. The structure to develop these functions has already been discussed. The Machine HMI interface enables the installation and deletion of these functions seamlessly.

1. **Install:** In install we need to first select the target agent where we need to install the functionality. Then we will click on install button which pops up a GUI with install button as shown in figure 4.17. Clicking on that we can browse the location of our function and install it. The function gets copied to the Downloadable Functions directory of the agent and a directory with function name is also created in Database directory. Configuration files like availablefunctions.txt, ActiveFunctionsList.txt and PassiveFunctionsList.txt also gets updated upon function installation.

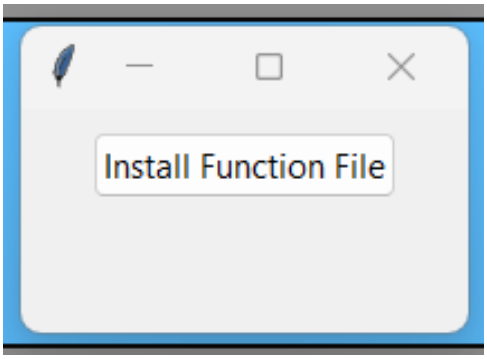

Figure 4.17: Install Function Button

2. **Delete:** Clicking on delete button of target agent pops up a delete button which leads to file manager window with selection of downloadable functions as shown in figure 4.18. Once the function to be deleted is selected it is removed from DownloadableFunctions directory of agent and its directory in Database folder is also deleted.

Active Function Initialization

As discussed earlier we need to configure 1 in ActiveFunctionsList.txt configuration file to start active function on separate thread upon agent initialization. This is achieved by selecting Active Function Initialization button and respective agent on HMI. This pops a tkinter based editor where we can

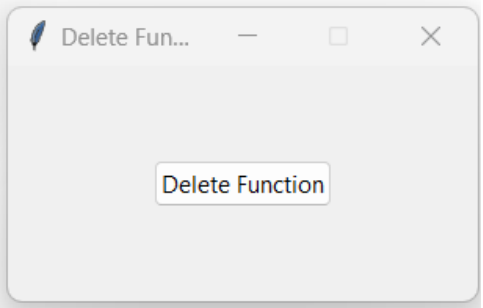

Figure 4.18: Delete Function Button

open ActiveFunctionsList.txt, edit it and then save it as shown in figure 4.19.

Figure 4.19: Active Function Initialization

Establish Connection

One of the most critical step in development of a standardized Digital Twin is to establishing and standardizing a connection between Observable Manufacturing Elements(OMEs) and Data Collection and Device Control layer as shown in figure 4.20. In our version of digital twin we have tapped into socket based communication happening between machine PLC and machine software. We have extracted machine parameters of our choice like spindle status, spindle speed, emergency status, coolant status, axes position,etc. and added then as node to OPCUA server. Also axes vibration data coming from external sensor can be added as node to OPCUA server. This data can also be accessed using using standard UA Expert client software on main central PC.

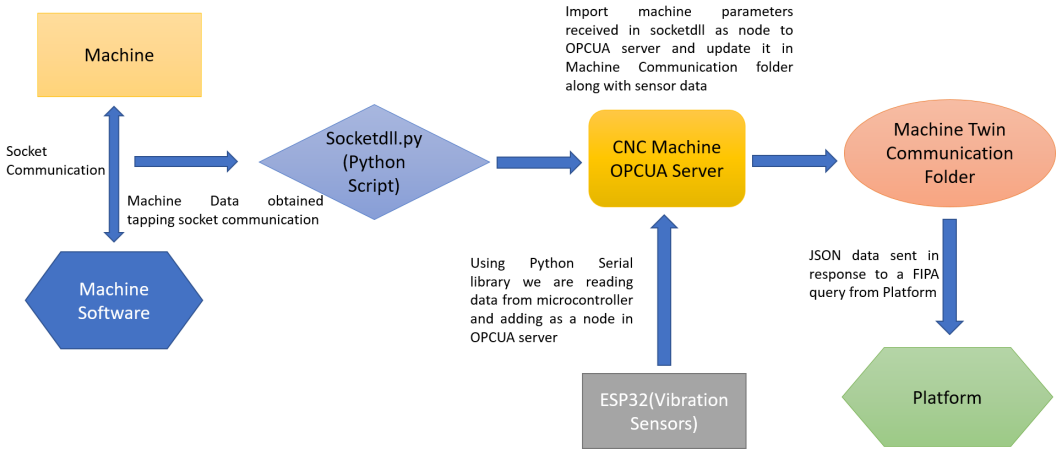

Figure 4.20: Schematic for Establishing Connection between OMEs and DCDCE

4.5.3 Process Flow of Digital Twin 3.0

Step-by-step sequential processes that need to be carried out to operate Digital Twin 3.0 has been represented in the figure 4.21.

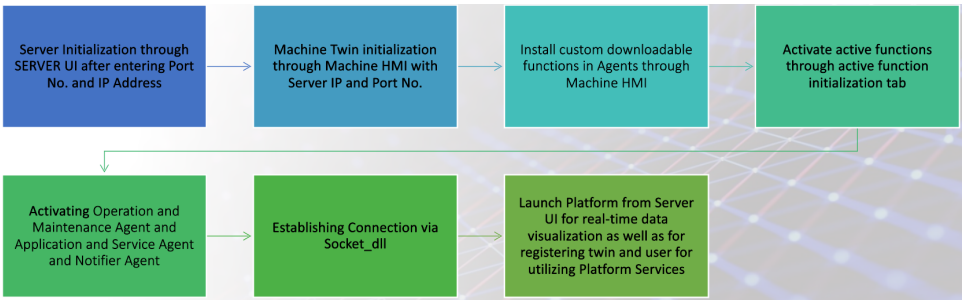

Figure 4.21: Process Flow for Digital Twin 3.0 activation

4.6 Development and Deployment of a Machine Learning Model for Tool Failure Prediction

To facilitate the functionality of **Analytic Service** we need to develop a Machine Learning Model for the analysis of historical data stored in the Machine Communication folder of the Application and Service agent. The

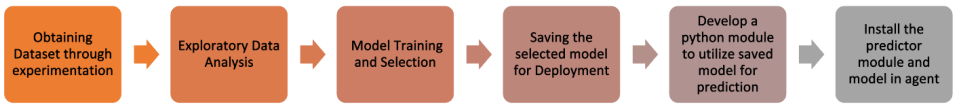

Figure 4.22: Concept Development for Deployment of ML Model

following six steps as shown in figure 4.22 are involved in the development and deployment of the Machine Learning model for tool failure prediction:

4.6.1 Obtaining Dataset

In our Digital Twin we need to do experiments with various tools and machining conditions and then download the historical data from the Application and Service agent but here for concept development, we have taken publicly available data of system-level manufacturing and automation research testbed(SMART) at the University of Michigan. A series of 18 experiments were conducted with worn and unworn tools for various machining conditions as shown in figure 4.23. Equipment operating conditions were recorded for 4 CNC motors of x, y, and z axes and spindle. The data is available to us as 18 Experiment files with CNC motors operating condition data and one file with machining conditions for all 18 experiments.

No	material	feedrate	clamp_pressure	tool_condition
1	wax	6	4.0	unworn
2	wax	20	4.0	unworn
3	wax	6	3.0	unworn
4	wax	6	2.5	unworn
5	wax	20	3.0	unworn

Figure 4.23: First Five Experiments Conditions

All 18 experiments data files were concatenated and the experiment condition file was concatenated as a column to each experimental data point for which the experiment number is valid. Thus finally a data frame with all 4 CNC motors operating condition data, exp number, material, feed rate, clamp pressure, and tool condition as columns was created.

4.6.2 Exploratory Data Analysis

Firstly **leaky columns** which will not have any impact on our model training like exp no., M1 sequence no., and M1 current program no. were removed. Secondly, columns with low cardinality were removed like material column removed as the same material wax was used in all 18 experiments, hence irrelevant in our model training. Also, certain features like Z1_CurrentFeedback, Z1_DCBusVoltage, and S1_SystemInertia have only one unique value thus removed.
To determine collinearity a correlation heat map was plotted as shown in figure 4.24 We can easily observe that there is a significant amount of multicollinearity among features hence in our binary classification problem of whether the tool is worn or unworn, linear regression models would not be significant. Hence we should consider Decision Tree based classification models.
Before proceeding to the selection of the model and finding the best model we need to plot the Class Balance plot of True and False data to observe whether our data is balanced for training or not. If the data is unbalanced we need to perform **re sampling** of data. In our case, as we can see from figure 4.25 the data for classification is balanced but since we are doing this

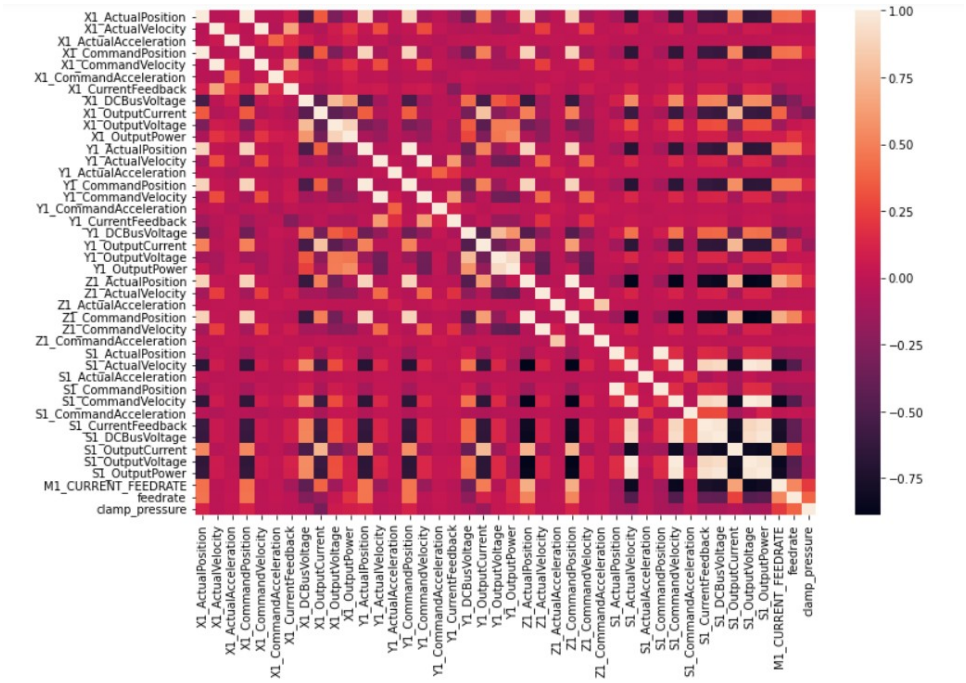

Figure 4.24: Correlation Heatmap

whole exercise for concept building **over sampling** was performed on data before training.

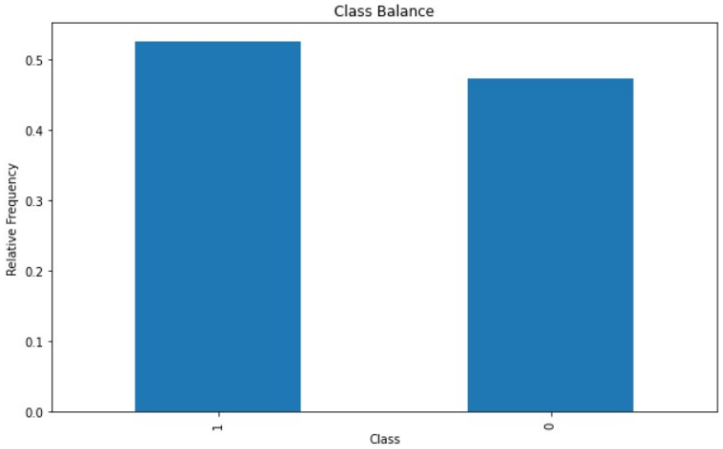

Figure 4.25: Class Balance

4.6.3 Model Training and Selection

The comparison of the performance of all the models on which we fitted our data can be seen in figure 4.26. As we can observe from figure 4.26, to achieve similar model accuracy Gradient Boosting takes a comparatively longer time which is understandable as boosting type ensemble method acts sequentially trying to fit on all the data points and improving on accuracy. Thus the best

Model	Hyper parameter Tuning	Model Accuracy	Time Elapsed
Logistic Regression	max_iterations = 1400	0.53	0.75
Decision Tree Classifier	depth_parameter = range(1,28,2)	0.98	2.42
Random Forest Classifier	Grid Search with 60 fits	0.9934	28.42
Gradient Boosting Classifier	Grid Search with 280 fits	0.9909	337.08

Figure 4.26: Comparison of Training Model

model for our use case is Random Forest Classifier. We can implement this as long as the binary classification is required. Once we have a very large data set then we can also train our model on Neural Networks and compare its performance with Random Forest and identify the best fit. The first 70 test data points plotted with actual against predicted for all the models for comparison in figure 4.27.

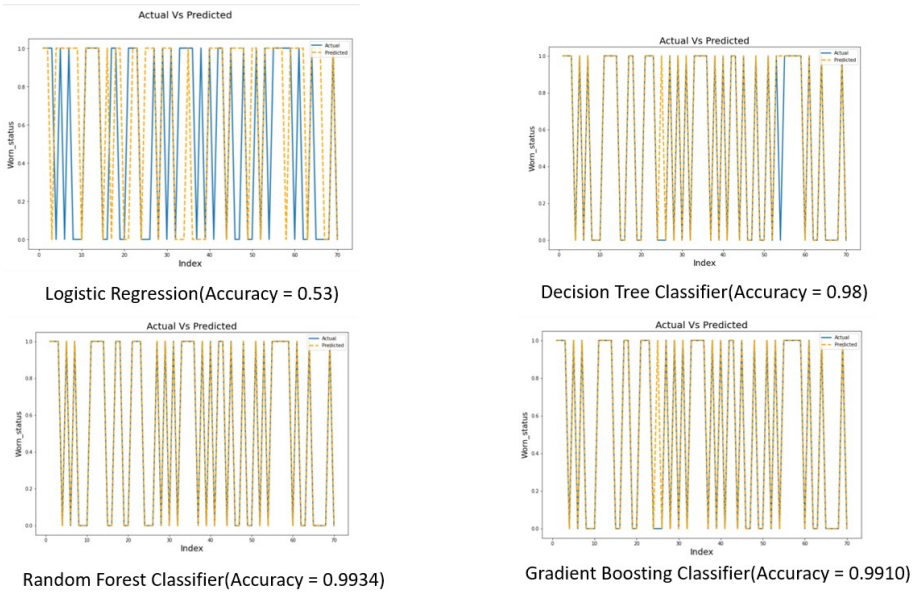

Figure 4.27: Actual Vs Predicted

4.6.4 Predictor Python Module

We have developed a python script and in this script we have defined a make predictions function which will take saved model and standard features data and predict whether the tool is worn or unworn. In order to invoke this function we need to install this module in Application and Service Agent and import make predictions function from the module in Agent script to give real-time predictions for report generation.
A plot of 50 random data points from X_test against one predicted by predictor module is shown in the figure 4.28

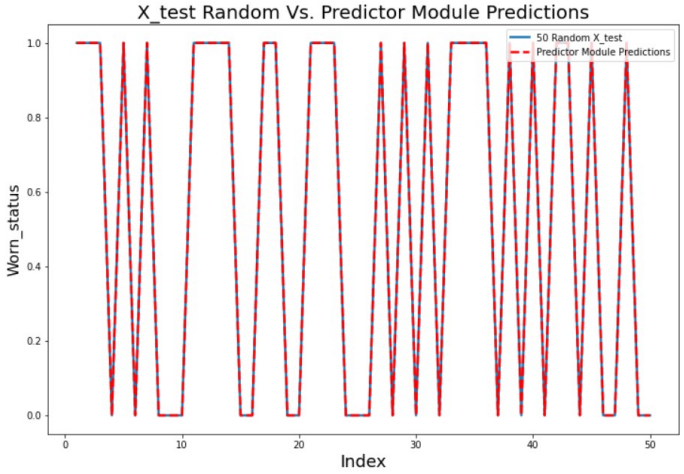

Figure 4.28: X test Random Vs. Predictor Module Predictions

Chapter 5

Conclusion and Future Work

5.1 Conclusion

We have utilized the Platform developed earlier as User Interface and Agent based Client-Server Architecture for the design and development of our Digital Twin. All the three layers of ISO 23247 reference architecture has been incorporated. Server UI and Machine HMI were developed to enhance deploy ability of the twin. In the interface we have also provided for the structure and installation of custom downloadable functions. Data from machine has also been standardized by incorporating OPCUA server.
In Chapter 4 we have talked about Minimum Viable Digital Twin, its roles and modules that need to be incorporated. In our Digital Twin 3.0 the digital twin roles which we have addressed are following:

- Digital twin developed respond to the query generated from user interface based on the data collected from sensors and machine
- Our digital twin displays the data collected on a digital dashboard on platform graphically enhancing the visualization and understanding of the user
- We have integrated alarms also in our twin which raises machine alarm based on axes vibration data and tool alarm predicting the tool failure
- Our twin also shows recommendations for turning on coolant when temperature is above defined temperature and changing the tool if worn tool is predicted

Apart from these roles, the all the standard modules of Minimum Viable Digital Twin has been integrated and also some customized modules were also incorporated. These are addressed in following:

- We have incorporated Asset Description in our digital twin as after registering and connecting twin on platform, we display twin data which describes twin id, name, floor no., machine limits, tool details, spindle details, coolant details, etc.
- Our digital twin also supports real time monitoring as data collected is displayed on platform simultaneously which can help in swift decision making
- We have utilized FIPA-ACL based communication protocols and performatives in our twin which enables interaction and transfer of data and information among various entities of twin
- Data collected in our digital twin is presented in a graphical and dynamic boxes which allows users to understand the data effectively and interact with the twin appropriately
- We have also incorporated Digital Twin management by providing install and delete functionalities to install new downloadable functions and delete the obsolete ones
- In our twin we have developed a downloadable functionality which performs hypothesis testing on 80 percent historical and 20 percent recent mean vibration data which in a way is descriptive analytics
- In our twin we have also developed and deployed a Random Forest based binary classification model for tool failure prediction which in a way is predictive analytics

5.2 Future Work

- Develop some analytical functionalities involving multiple machines and query from Platform
- Develop a chat bot which can query from Twin Agent and Sub-system Entity agents based on Natural Language Processing which can be step-by-step guidance to some activity that need to performed sequentially
- Develop some functionality to predict failure of one machine hence helping in decision of taking material on other machine for processing in a twin with multiple machines

- Incorporate simulation models in our Application and Service Agent to determine remaining useful life of the tool based on predictions from simulated data
- One can also incorporate control action based on historical data and computational modules to make predictions about future behaviour and taking control measures
- Conduct experiments with various machining conditions using worn and unworn tools and verify the functioning of binary classification model on our machine operating conditions
- We can facilitate the graphical animation and 3D Visualization for the observable manufacturing asset
- System robustness can be provided by using backup servers and agents in parallel
- User authentication process can be converted into more secure options like two factor authentication
- Since the Platform is Django based it can be hosted on server and can be accessed remotely on mobile phone through a website

References

[1] ISO 23247, Digital Twin Framework for Manufacturing

[2] Dazhong Wu et al., A Comparative Study on Machine Learning Algorithms for Smart Manufacturing: Tool Wear Prediction Using Random Forests, Journal of Manufacturing Science and Engineering, JULY 2017, Vol. 139 / 071018-1

[3] Rui Zhao et al., Learning to Monitor Machine Health with Convolutional Bi-Directional LSTM Networks, MDPI, Sensors 2017, 17, 273; doi:10.3390/s17020273

[4] BAOTONG CHEN1 et al., Smart Factory of Industry 4.0: Key Technologies,Application Case, and Challenges, SPECIAL SECTION ON KEY TECHNOLOGIES FOR SMART FACTORY OF INDUSTRY 4.0 IEEE Access

[5] Sungho Park et al., Prediction of the CNC Tool Wear Using the Machine Learning Technique, International Conference on Computational Science and Computational Intelligence (CSCI), IEEE

[6] Rockwell Automation, Digital Twins providing real value in manufacturing

[7] Aws iot twinmaker features. https://aws.amazon.com/iot-twinmaker/. Accessed:2022-01-18

[8] Evgeniy Krasnokutsky, Machine Learning Application in the Manufacturing Industry, MobiDev - Custom Software Development Company

[9] Azure digital twins documentation. https://docs.microsoft.com/en-us/azure/digital-twins/. Accessed: 2022-01-12

[10] What is aws iot twinmaker? https://docs.aws.amazon.com/iot-twinmaker/latest/guide/what-is-twinmaker.html. Accessed: 2022-01-18

[11] Greg Cline, Industry 4.0 and Industrial IoT in Manufacturing: A Sneak Peek, Aberdeen Group

[12] ME714 Computer Integrated Manufacturing - Lecture Notes by Dr. Soham Majumdar (IIT Bombay)

[13] Ibm digital twin exchange. https://www.ibm.com/in-en/products/digital-twin-exchange. Accessed: 2022-01-20

[14] Henning Kagermann, Johannes Helbig, Ariane Hellinger, and Wolfgang Wahlster. Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Securing the future of German manufacturing industry; final report of the Industrie 4.0 Working Group. Forschungsunion, 2013

[15] Shanshan Yang, Aravind Raghavendra MR, Jacek Kaminski, and Helene Pepin. Opportunities for industry 4.0 to support remanufacturing. Applied Sciences, 8(7):1177,2018

[16] Felipe Orellana and Romina Torres,From legacy based factories to smart factories level 2 according to the industry 4.0, International Journal of Computer Integrated Manufacturing, 2019, VOL. 32, NOS. 4–5, 441–451

[17] Monika Gadre and Dr. Aruna Deoskar,Industry 4.0 – Digital Transformation, Challenges and Benefits, International Journal of Future Generation Communication and Networking, Vol. 13, No. 2, (2020), pp. 139-149

[18] Luis Ribeiro. The design, deployment, and assessment of industrial agent systems. In Industrial Agents, pages 45–63. Elsevier, 2015

[19] Jeffrey Voas, Peter Mell, and Vartan Piroumian. Considerations for digital twin technology and emerging standards. Technical report, National Institute of Standards and Technology, 2021

[20] Digital twins are mission critical. https://www.ge.com/digital/applications/digital-twin. Accessed:2022-01-20.

[21] EDA content director Mike Santarini. Digital transformation: How siemens eda helps you engineer a smarter future faster. Technical report, Siemens Digital Industries Software.