TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

JOE SAMUEL DAVID

# DEVELOPMENT OF A DIGITAL TWIN OF A FLEXIBLE MANUFACTURING SYSTEM FOR ASSISTED LEARNING

Master of Science Thesis

Examiners: Assoc .Prof. Andrei Lobov,
Prof. Jose. Luis Martinez Lastra

Examiner and topic approved by the Council meeting of the Faculty of Engineering Sciences on 28th of February 2018

# ABSTRACT

**JOE SAMUEL DAVID**
Tampere University of technology
Master of Science Thesis, 91 pages
November 2018
Master's Degree Programme in Automation Engineering
Major: Factory Automation and Industrial Informatics
Examiner: Prof. Jose Luis Martinez Lastra, Assoc. Prof. Andrei Lobov,

Keywords: Digital Twins, Flexible Manufacturing Systems, Ontological Reasoning, Learning Analytics, Learning Evaluation, Learning Outcomes, Pedagogy, Web Ontology Language (OWL)

Learning Factories provide a propitious learning environment for nurturing production related competencies. However, several problems continue to plague their widespread adoption. Further, assessment of attained competencies continue to remain a concern.

This study proposes the use of digital twins as an alternative learning platform for production engineering courses. It is proposed that in the context of manufacturing pedagogy, digital twins of manufacturing processes can play a significant role in delivering efficacious learning experiences. The high-fidelity replication of the physical system aids with reflective observation of the entailed processes in the greatest possible detail, fostering concrete learning experiences.

An iterative research methodology towards modelling a pedagogic digital twin is undertaken to build a learning environment that is characterized by ontologies that model learning objectives, learning outcomes and assessment of the said outcomes. This environment facilitates automated assessment of the learner via ontological reasoning mechanisms. The underlying schema takes into account the learner's profile and focuses on competency attainment through reasoning of behavioural assessment of aligned learning outcomes.

The thesis presents also a case study that demonstrates how the learner's competency level may be evaluated and compared with other learners thus warranting its use a learning tool that proves beneficial in an academic setting.

Tampere, 20.11.2018

Joe Samuel David

# PREFACE

The past nine months have been nothing short of an exciting journey while working on this master thesis and its completion would not have been possible without the guidance and assistance from a few people.

The work presented in the thesis was part of the Virtual FMS project that was carried out in the Department of Mechanical Engineering and Industrial Systems at Tampere University of Technology and I owe deep gratitude to Professor Minna Lanz for providing me with this opportunity.

I would like to thank Professors Jose Lastra and Andrei Lobov under whose supervision the thesis was carried out. My sincere gratitude goes out to Professor Andrei Lobov for his valuable guidance and scholarly inputs that helped me navigate through the various problems that I faced throughout the course of this work. His feedback, whether constructive or critical, helped me crystallize the content of the thesis and structure it well.

I would also like to thank Ville Toivonen, Veikko Valjus, Jouko Kiviö and Hasse Nylund who provided me with necessary support and materials throughout the thesis.

I would like to express my gratitude to my parents and family for their love and support throughout this journey.  I would also like to thank my friend Siva Sankar who got me through tough days.

And lastly, but by no means least, I would like to thank the Department of Mechanical Engineering and Industrial Systems for providing me with all necessary facilities to carry out the thesis.

## CONTENTS

# LIST OF FIGURES

## LIST OF TABLES

## LIST OF SYMBOLS AND ABBREVIATIONS

| | |
|---|---|
| BOL | Beginning of Life |
| MOL | Middle of Life |
| EOL | End of Life |
| PLM | Product Lifecycle Management |
| IoT | Internet of Things |
| NASA | National Aeronautics and Space Administration Limited |
| ERP | Enterprise Resource Planning |
| MES | Manufacturing Execution Systems |
| FMS | Flexible Manufacturing System |
| CNC | Computer Numerical Control |
| SMC | Single Machine Cell |
| FMC | Flexible Manufacturing Cell |
| REST | Representational State Transfer |
| HTTP | Hypertext Transfer Protocol |
| TCP | Transmission Control Protocol |
| URL | Uniform Resource Locator |
| URI | Uniform Resource Identifier |
| HTML | Hypertext Markup Language |
| MQTT | Message Queuing Telemetry Transport |
| IP | Internet Protocol |
| API | Application Programming Interface |
| SOAP | Simple Object Access Protocol |
| XML | Extensible Markup Language |
| OPC UA | OLE (Object linking and Embedding) for Process Control |
| MOM | Message Oriented Middleware |
| DL | Description Logics |
| FOL | First Order Logics |
| UNA | Unary Name Assumption |
| SPARQL | SPARQL Protocol and RDF Query Language |
| SERQL | Sesame RDF Query Language |
| OWL | Web Ontology Language |
| RDF | Resource Description Framework |
| RDFS | RDF Schema |
| SA | Situational Awareness |
| CAD | Computer-aided Design |
| SOLO | Structure of the Observed Learning Outcome |

# LIST OF PUBLICATIONS

[P1] J. David, A. Lobov, and M. Lanz, “Leveraging Digital Twins for Assisted Learning of Flexible Manufacturing Systems,” Proceedings of the 16th IEEE International Conference on Industrial Informatics, (INDIN), IEEE, 2018, pp. 529–535.

[P2] J. David, A. Lobov, and M. Lanz, “Learning Experiences Involving Digital Twins,” Proceedings of the 44th IEEE International Conference on Industrial Electronics, (IECON), 2018

[P3] J. David, A. Lobov, and M. Lanz, “Attaining Learning Objectives by Ontological Reasoning using Digital Twins” (Submitted)

# 1. INTRODUCTION

This chapter aims to contribute with an introduction that provides the reader with the context this research belongs to. Besides bringing the thesis to context, it also clarifies the objectives and scope of the work involved and a brief overview of the approach taken for the same.

This chapter is further structured as follows. Section 1.1 presents background information that brings the research problems that the author aims to solve into context. Section 1.2 defines the problem and formulates the research questions that presents the rationale for thesis. Once the research questions are formulated, the objective of the thesis is presented in Section 1.3. Section 1.4 briefly outlines the research process and techniques undertaken throughout the development of the thesis. Limitations of the work and challenges during the development of the same are covered in Section 1.5. Section 1.6 details the structure of the remainder of the thesis. Lastly, Section 1.7 presents the publications of the author as a part of the research done during the thesis.

## 1.1 Background

The manufacturing industry is vital to any economy as it has a "domino effect" on most other industry sectors, fostering employment and driving growth. This industry plays a pivotal role for every economy in pursuit of a sustainable economic growth.

This industry reaps benefits from the advantages of competition with most businesses constantly evolving for the development of sophisticated yet cheaper new mechanisms to manufacture goods. As one manufacturer finds a way of making products that is not only more efficient but also more cost-effective, one can be certain that another will follow. It is an industry that has been revolutionized time and time again, with new concepts and new beginnings that changes the way that things are done from the ground up.

These periods known as Industrial Revolutions started with the First Industrial Revolution wherein steam engines were popularized; the Second Industrial Revolution was the time of electrical energy and mass production; the third and most recent Industrial Revolution was the introduction of IT environments and other forms of digitalization.

With the arrival of the fourth industrial revolution what is better known as Industry 4.0, the manufacturing sector has seen a holistic shift from conventional automated systems to one that is driven by Internet of Things (IoT) and cloud computing involving cyber physical systems. However, manufacturing pedagogy and training have failed to keep pace with this rapid advancement. Industry 4.0 is largely concerned with digitalization and convergence of the real world with the virtual world and manufacturing pedagogy is facing a challenge now more than ever to produce workforce that can cope with this paradigm shift. Educators will need to adapt curriculum and teaching methodologies to help instill concrete understanding of the new trends and principles.

In recent times, attempts have been made to address this shift. Learning factories have been set up with the intention to help with action-oriented learning [1][2]. Teaching factories that aimed at solving some of the drawbacks of learning factories is yet another approach in this direction to integrate education, research and innovation based on the knowledge triangle in a single initiative [3]. A prime factor for the success of these factory concepts have been the participation of the industries that provide invaluable mentoring to students by exposing them to real world problems [4]. Computer simulations have also been utilized for manufacturing education and proved to be of valuable support to the learning objectives. Virtual Reality is another concept that is being exploited of late in manufacturing pedagogy [4]. Although these concepts have been around for decades and was not developed with the vision of Industry 4.0 in mind research shows that they have been vital in production based engineering education even in the current scenario.

## 1.2 Problem Statement and Research Questions

Of late, learning factories seem to be finding place in most universities and a popular method on giving hands-on experience to students. A learning factory is a facility that realizes a process or product in an academic setting for the purpose of training and educating students [5], normally in or in close proximity to the campus premises. They are set up with the intention to inspire action-oriented experiential learning [2][6][7][1].

However, Some of the limitations of the learning factories identified are as follows [5]:

- Limited mapping ability for challenges prevalent in academia and industry as learning factories generally focus on particular aspects of manufacturing.
- Space and cost related issues when it comes to mapping the different factory levels.
- Time required to complete production orders having a high cycle time.
- Fixed locations of learning factories mean limited mobility.
- Evaluation of production related competencies after the learning experience.

Assisted learning via digital twins aims to support student in education process grasping, understanding and applying new ideas. Digital twins are seen as an environment, where the learning process can be facilitated. In order to support student, the ‘digital assistant’ has to have some representation of learning objectives and feedback from and to the student to guide her or him towards set objectives. Such a system should be able to evaluate and compare performance of different students. These gives a rise to the following research questions:

- **RQ1:** How digital twins in an academic set up can augment the learning experience and how it can mitigate the limitations of learning factories?
- **RQ2:** What is a systematic approach to develop such a digital twin?
- **RQ3**: How to model learning outcomes in context of digital twins?
- **RQ4:** How to evaluate performance of the students?
- **RQ5:** How to guide a student towards desired level of skills with respect to her/his current status?

## 1.3 Objective

The purpose of this master thesis is to develop a digital twin of a (flexible) manufacturing system to help address some of the drawbacks of learning factories listed in the problem statement whilst developing a product where future work can be done in order to scale to address all the drawbacks. We investigate how digital twins can be used as an educating tool to expedite the learning process of manufacturing systems.

## 1.4 Research Methodology

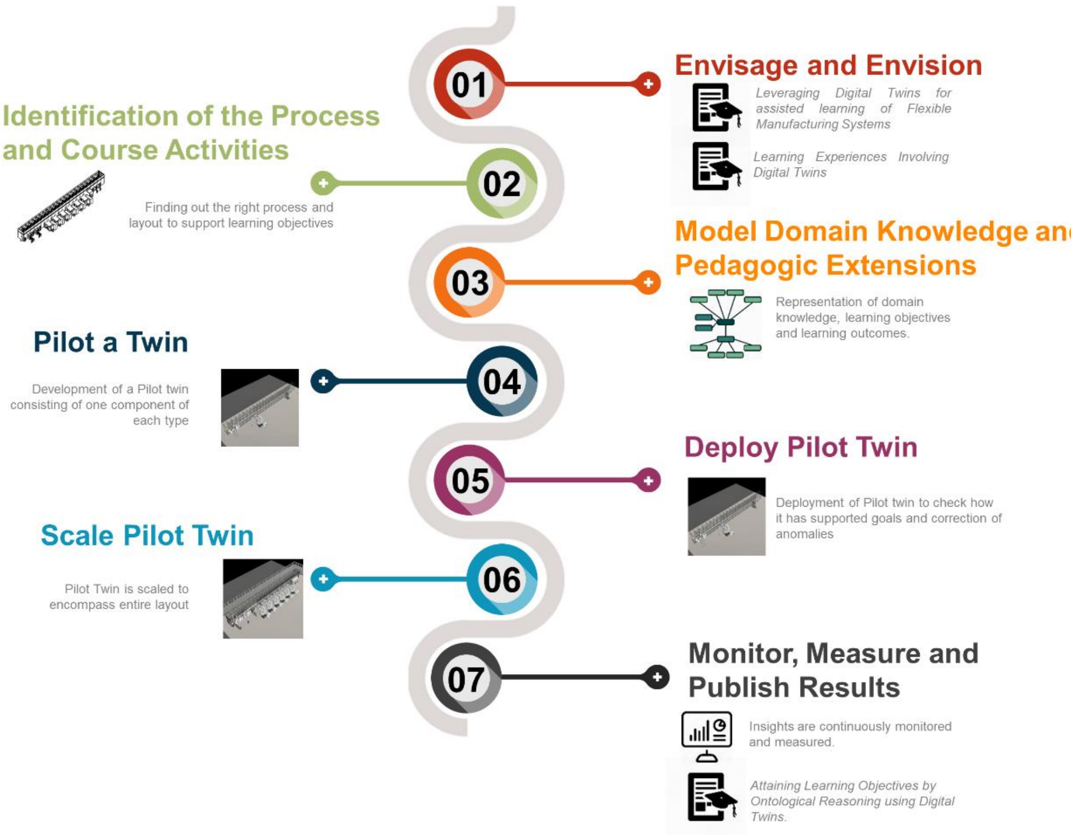


***Figure 1.*** *Research Methodology.*

Figure 1 shows the steps taken during the course of the study. One of the major factors to be taken into consideration while getting started with creating a digital twin is the optimal level of detail in creating one. A simple model will not yield the advantages that the concept assures and a detailed holistic approach would result in getting lost in the intricacies of all the encompassed sensors, signals equipments etc. The following steps were taken to realize the Digital Twin [8].

**Envisage/Envision:** This step involved brain storming to identify the possible use cases and how the digital twin can prove to be advantageous in an academic setting. Two publications emerged as part of this process. The first publication established a didactic methodology involving learning in a pedagogical infrastructure using Digital Twins and presented a use-case [P1]. The second publication justified such use of digital twins as a means of education by drawing a parallel to established learning theories [P2]. This stage also saw establishing of the learning outcome as "To understand the basics of production activity, key manufacturing techniques and operating models in Finnish industry." This stage, while laying the foundations of the research also gave the green signal to pursue solutions to the posed research questions.

**Identification of the Process and Course Activities:** Identification of the process involved determine the right configuration that are optimal in realizing the goals of the course, i.e. to understand the principle of flexible manufacturing systems. This was done over meetings with the course personnel and the company responsible for the manufacturing system until a configuration layout was mutually agreed upon. The activities and exercises of the course were also planned on this stage.

**Model Learning Objectives and Pedagogic Extension:** This stage models the learning objectives along with certain pedagogic extensions to make the digital twin realize its potential in the academic setting.

**Pilot a Twin:** Once the manufacturing layout system was agreed upon, a pilot digital twin of a subset of the complete layout is implemented that comprised of one component of each type. Throughout the implementation, an open approach is maintained that would allow flexibility to integrate new data, in the form of sensors, equipment etc., whilst keeping an emphasis on performance as digital twins can be resource-hungry upon scaling them to encompass big processes.

**Deploy Pilot Twin:** The pilot twin is next deployed and several performance parameters are tuned and optimized.

**Scale the twin:** After the successful deployment of the pilot twin, the digital twin is scaled to mirror the entire layout and equipment, to realize its full potential.

**Monitor, Measure and Publish Result:** The implemented solutions is continuously observed to quantify the usefulness of the Digital Twin in the academic setting in realizing the course goals. Once the value is realized from the digital twin, the drive for greater results in the form of optimization and enhancements is considered as future developments to the digital twin. The results are published in an academic conference at the end of this stage.

## 1.5 Limitations and Challenges

The manufacturing control system's controller was essentially a black-box with little or no information made available of what's going on 'under the hood'. This translated to implementation using the 'best available information'. As such, some compromises were made during the implementation stage.

**Limitations:**

1. It was understood that the Simulator supported the use of sockets (sockjs JavaScript Library) for communication. However, since its details were not made available, the implementation was made via polling RESTful (Representational State Transfer) API (Application Programming Interface) at a high polling frequency.
2. As a consequence of the limitation above, the digital twin was realized as a near real-time replication of the simulator. Typically, the delays (1/ (polling frequency in Hz) s) associated are the order of milliseconds and are not noticeable to humans (default Value: 100ms), such that the sense of the digital twin is maintained.
3. For avoiding performance-related issues, the movements of several parts are synced with the start and end states, and not throughout the movement. For example, if the Crane is to move from A to B and then performs task 1, the twin starts when the simulator starts from A, and if it reaches B before the simulator does, it waits until the simulator does so before performing task A. As opposed to this, if the crane were to replicate every position of the crane in the simulator, rendering its position in 3D 10 times every second, the implementation would require high computing resources and would not be feasible.
4. The Implementation was carried out using a software Visual Components. This decision was based on the fact that license was readily available on campus besides it being a leading developer of 3D simulation software and solutions for manufacturing.

**Challenges:**

1. Lack of ready-to-use components for the digital twin model meant that some components needed to be modelled from scratch (Crane.Storage) and most components needed to be upgraded in terms of its functionality to achieve higher fidelity of the twin's physical counterpart (the Simulator).
2. Lack of documentation of RESTful web services rendered discovering RESTful web services by unconventional methods, a time consuming task. Further testing discovered web services was another task of its own, that consumed considerable time.
3. Determining the right polling frequency, as a trade-off between performance and fidelity remains to be a challenge.
4. Use of Visual Components as the software of choice to implement the Digital Twin dictated the language of programming as a stackless version of Python 2.7.1.

This meant that 3rd party-libraries would not necessarily work with the implementation and python version and a suitable version needed to be searched for.

## 1.6 Thesis Outline

This chapter brings the thesis into context and formulates the research questions. The remainder of the thesis has been structured as follows. The second Chapter introduces the theoretical concepts underpinning the research done in the thesis. Chapter 3 introduces state of the art research done in developing a didactic framework where the digital twin may be used and justifies the proposition. In Chapter 4, research methods undertaken throughout the work done in the thesis is presented. Chapter 5 is about implementation of all of the work done within the scope of this study. In Chapter 6, how the thesis has addressed all the posed research questions is discussed. Finally, Chapter 8 reflects on the work done and different directions of possible future work are proposed.

The following traceability infographic (Figure 2) gives an overview of the thesis structure and maps each of the chapter to the research questions it addresses

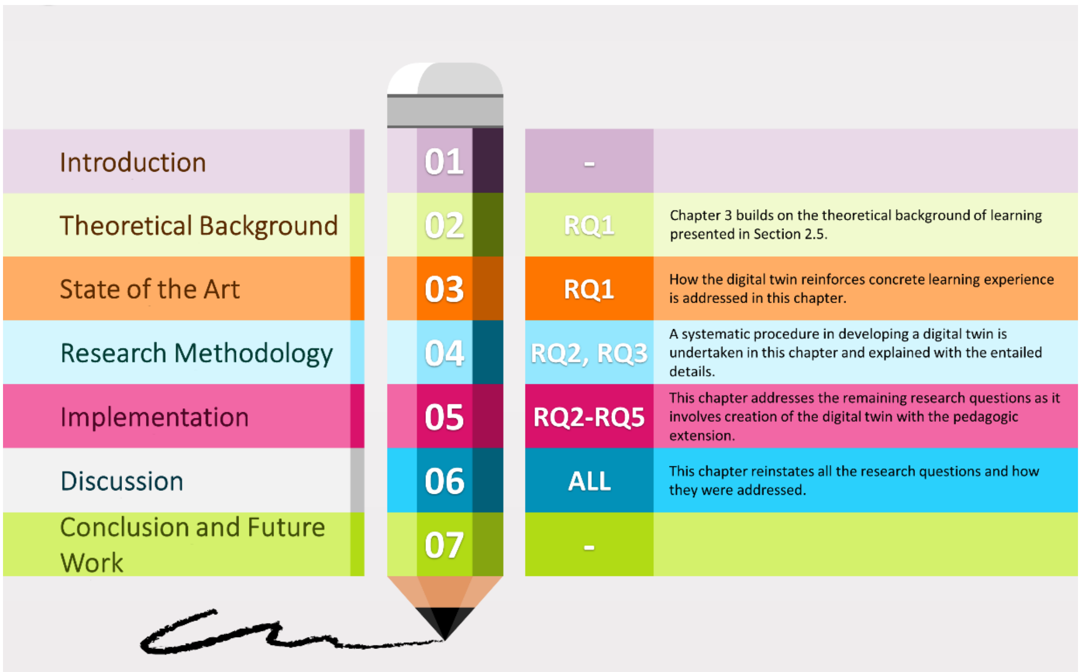


***Figure 2.*** *Research Traceability Infographic.*

## 1.7 Publications and Author's Contribution

The author has published three papers in academic conferences pertaining to research done during the initiation, implementation and completion of this Master Thesis as listed in the List of Publications Section.

The first publication titled "*Leveraging Digital Twins for Assisted Learning of Flexible Manufacturing Systems*" was done as a part of the envision stage of the thesis to establish a conceptual didactic framework using digital twins based on a sound pedagogical theory, i.e. Kolb's Experiential Learning. The paper presents also a case study where a Digital Twin of pedagogic significance is introduced and subsequently discussed.

The second publication titled "Learning Experiences Involving Digital Twins" was also a research that was done as a part of the research at the envision stage that aimed at justifying the pedagogical digital twin architecture in the first publication by instantiating behavioural, cognitive and humanist learning theories in it. Further, the paper investigates how situational awareness can be beneficial in such an environment and its contribution to the overall learning experience.

The third publication titled "Attaining Learning Objectives via Ontological Reasoning using Digital Twins" presents the results of this study along with the developed use case to evaluate student performance.

# 2. THEORETICAL BACKGROUND

In this chapter, the theoretical background relevant to the scope of the study in this thesis is presented. Section 2.1 introduces the concept of digital twins to the reader by presenting a comprehensive review of literature on its origin and evolution, before delving into a general architecture. Flexible Manufacturing Systems and its related concepts are introduced to the reader in Section 2.2. A review on some of the communication protocols used in the thesis, and those which may prove to be an alternative to the one used is presented in Section 2.3. Section 2.4 provides an insight to a popular debate on knowledge representation and introduces all the tools used for the same. Finally, Section 2.5 presents a literature review on behavioural, cognitive and humanistic learning theories and presents how situational awareness can contribute to the learning experience as an overture to state of the art work in the next chapter.

## 2.1 Digital Twins

The existence and development of manufacturing businesses is directly associated with the success of its products. Products have a certain life-cycle and its management in entirety have been in focus of late. The life cycle of a product is structured into 3 stages: Beginning Of Life (BOL), Middle Of Life (MOL) and End Of Life (EOL). Briefly, BOL comprises every- thing from conceptual design and production of the product; MOL comprises of usage and maintenance; EOL comprises of different scenarios such as complete product reutilization after updates or refurbishment, product component reuse and update, decommissioning, disassembly and recycling [9]. Product lifecycle management is defined as “the business activity of managing, in the most effective way, a company’s products all the way across their lifecycles; from the very first idea for a product all the way through until it is retired and disposed of” [10]. PLM evolved from Product Data Management (PDM) that evolved from the advent of Computer Aided Design Systems [11]. Of late, The Internet of Things (IoT) is augmenting the PLM landscape by allowing all the stakeholders of the product’s lifecycle to gain an insight into the performance of the product and how it is being used by its customers without them volunteering for feedback. Information from sensors embed in the product or systems can be analyzed to develop insights about its use including its potential failure [12]. Taking this a step further is the concept of Digital Twins.

### 2.1.1 Origin, Evolution and Definitions of Digital Twin

The concept of “twins” dates back to the early days of space exploration when NASA used in its Apollo program, a twin of the original space vehicle to mimic the conditions of the real vehicle during its mission. The twin played a vital role in the rescue mission when disaster struck the Apollo 13 mission. Although there was nothing digital about it then, they used the twin of the component there were trying to fix and had it tested on the ground before relaying the rescue strategies to those on-board the space vehicle. Almost five decades later, today NASA uses digital twins to manufacture next generation space vehicles [13].

Dr. Micheal Grieves first presented the Digital Twin concept as a “Conceptual idea for PLM” in 2002 at University of Michigan. It was termed as the “Mirrored Spaces Model” [14] and “Information Mirroring Model” [15] before being coined as the ``Digital Twin'' in 2011 [16].

NASA first defined Digital Twins as “an integrated multiphysics, multiscale, probabilistic simulation of an as-built vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin” [13].

On similar lines, Grieves defines the digital twin as “a set of virtual information constructs that fully describes a potential or actual physical manufactured product from the micro atomic level to the macro geometrical level” [17].

### 2.1.2 Viewpoints

Based on these definitions, it can be seen that the concept of Digital Twins has contextual nuances and hence different viewpoints. While in some contexts it is defined as an exact replica of an as-built product which replicates the mechanical wear and tear during the product life cycle [18], in others it is defined as a sensor-based digital model that produces real-time simulation [19].

From a simulation viewpoint [20], it is defined as a set of executable models with just sufficient levels of detail for the problem in hand, that evolves along the life-cycle of the product which helps solve problems relevant for the real system. Thus in the regard, it can be seen as augmentation of the MBSE concept from manufacturing to operation and service phases.

A digital twin differs from the traditional CAD or a sensor enabled Internet of Things device in the sense of its ability to provide comprehensive real-time association between both worlds, physical and digital, that exists for its entire life-cycle. Many useful insights can be obtained as a direct consequence of this. Answers to questions that could not be answered before in real-time can be provided by incorporating digital twins in business operations.

From the previous mentioned definitions and viewpoints the following characteristics of the Digital Twin can be generalized: (1) there exists a physical asset and its digital counterpart (the twin) and a real-time association between them with application specific granular fidelity. (2) This association lasts as long as the life-cycle of the associated physical asset.

### 2.1.3 Architecture of a Digital Twin

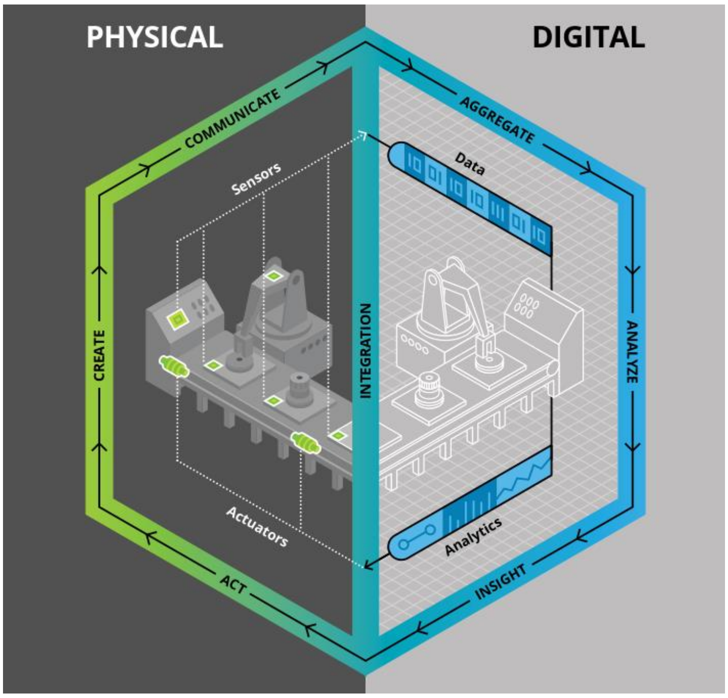


***Figure 3.*** *Architecture of Digital Twin* [8]

The figure above (Figure 3) depicts a general architecture of Digital Twins. It consists of the following:

**Physical Asset:** The physical asset in the physical world.

**Sensors:** Sensors are the lifeline of manufacturing and IoT has become the foundation with which smart they operate [21]. With such sensors, real-time information about physical parameters is made available which is an important enabler of Digital Twins. A myriad sensors distributed throughout the factory churn out data about the equipments (statuses), products (statuses, quality), processes (torque, displacement, etc.) and the environment (ambient temperature, humidity, etc.) to make it available for the twin

**Data:** Data is received from the sensors and is fused together with information from the Enterprise Resource Planning and Manufacturing Execution systems.

**Analytics:** State of the art technologies in Artificial Intelligence such as Pattern Recognition, Unstructured and Multi-modal data analysis are employed to analyze the data to obtain deeper insights with the available information.

**Digital Twin:** The digital counterpart of the physical asset that is modelled in software as its near real-time replica.

**Actuators:** Actuators are the means by which the Digital Twin acts upon the physical asset.

### 2.1.4 Digital Twin Process

**Create:** A myriad sensors distributed throughout the factory churn out data about the equipments (statuses), products (statuses, quality), processes (torque, displacement, etc) and the environment (ambient temperature, humidity, etc.) to make it available for the twin. Sensors are the lifeline of manufacturing and IoT has become the foundation with which they operate [21]. With such sensors, real-time information about physical parameters is made available which is an important enabler of Digital Twins.

**Communicate:** This step enables real-time, seamless, two-way communication between the digital and the physical worlds. This step could include some processing of data (edge processing) with the objective of translating data formats or to reduce network congestion. Key enabler of this is step is the deployment of fieldbuses (or similar functional alternatives) that enable real time, bi-directional data transmission between the digital world and the physical world [22]. This step also takes incorporates associated security challenges such as Encryption, centralized key management and so on [23].

**Integrate:** Sensor data is fused with data from the ERP and MES systems

**Analyze:** Integrated data is then analyzed using Advanced Information Analytics (Big Data, Predictive Analytics, Streaming analytics, Prescriptive Analytics, etc.). This is arguably the most important step in the sequence and where the core functionality of the Digital Twin resides. Without data analysis technologies, the perceived data from the previous step would continue to remain meaningless. It is only when the data converges with the digital twin, a high-fidelity, real-time system is realized.

**Insight:** In this step the insights, from the analytics are used to develop cognizance of existing processes, or product performance over time, and leverage this information to highlight undesirable differences in the performance between the physical asset and the digital counterpart.

**Act:** This step closes the loop between both worlds wherein questionable insights worthy of action from the previous step are acted upon with the help of the actuators in the physical asset.

## 2.2 Flexible Manufacturing Systems (FMS)

The inability to manage the highs and lows of consumerism can be detrimental to any manufacturing industry. Thus, it became imperative for these industries to come up with a solution that would allow them to be flexible with their handling of the needs of the market. Flexible manufacturing systems is a relatively new concept in manufacturing that allows for a great deal of flexibility in the manufacturing process. In such systems, several machines and stations are all under the control of a central computer that can handle the fluctuations brought on by consumer demand. True to their name, these systems are highly flexible and are adaptable based on the fluctuating needs of the market, with optimum competitiveness. Product lines can be dropped based on low demand and picked up later on during the season which can lead to an improved workflow.

### 2.2.1 What is an FMS?

FMS is an umbrella term for a form of manufacturing that comprises of a highly automated machine cell that allows for adaptable manufacturing. A typical flexible manufacturing system comprises of a computer system that handles the distribution of the parts as a whole. This system controls workstations (usually in the form of CNC tools), which are set to process pre-programmed sequences that allow for automated manufacturing. In an FMS, these workstations are then connected under a single machine that systematically stores and processes materials. This system's flexibility allows for the processing of unique sets of parts simultaneously in the workstations, which can be adjusted and controlled to respond to the fluctuations in demand.

### 2.2.2 Flexibility in FMS and its types

As we mentioned previously, flexibility is an umbrella term that can be applied to a wide array of concepts. It is not a term that can be easily defined or constrained under one singular form. Hence, its application towards flexible manufacturing systems are thusly as unique and continues to evolve and grow as it has been seen to do over the past twenty or so years.

Various definitions of flexibility exist in literature but, rather than attempting to define flexibility, it would be more apt to determine the "types" of flexibility and how these types are able to function in real-time situations. The following breakdown will include the definition of said flexibilities and the purpose in which one might find them being used within a manufacturing facility [24].

**Machine Flexibility:** Machine flexibility refers to the ability of the machine to perform a multitude of operations such as grinding, milling within a reasonable cost and time.

Such flexibilities are useful when they come together to perform an act that can replace the need for, say an assembly robot for an assembly operation. As an example, a machining center with a number of tools and NC programs is more flexible than one that can manufacture only a single part.

Purpose: In a sense, machine flexibility is the benchmark for an FMS system. It lays out a foundation that can replace the need for machines that need constant changing. These flexible machine systems are efficient when it comes to changing tools as they offer higher utilization and will result in less unnecessary parts.

**Material Handling Flexibility:** This type of flexibility provides for efficient transportation of parts between different operations, which allows for a smooth workflow that can handle variations in input and output through time.

Purpose: The purpose for this type of flexibility relies solely on its ability to boost the utilization of machines. The different automated storage and retrievals systems allow for efficient processing of information based on the input given, which as mentioned, will allow for easy transitions between one operation to the next.

**Operation flexibility:** Operational flexibility concerns the ability of a part to be manufactured in multiple ways in varying sequences. The system would then have to be able to accommodate changes in order, and thereby its operations by alternative orders using alternative operations.

Purpose: The ability to operate at such flexible rates allows for optimum efficiency concerning the scheduling of parts in real-time applications.

**Process Flexibility:** This refers to a system's ability to produce a variety of different parts (within a reasonable set type), without requiring major overhaul when it comes to setting it up.

Purpose: This type of flexibility allows certain machines to be shared between similar parts, which minimizes the need to require other machines for said parts. The application of this type of system results in fluctuating batch sizes and the reduction of storage costs. It is the perfect solution for managing and adjusted based on the needs of the market.

**Product Flexibility:** A system that allows for efficient substitution of parts without the need for extraneous dips into extra time and cost.

Purpose: The ability to switch out product mixes is of the utmost importance when it comes to staying competitive in the manufacturing market. It allows for certain parts of a similar type to be produced for a small time-window before being replaced by another. In all, it results in a facility that is highly flexible in regards to what it produces.

**Routing Flexibility:** A flexible system that allows for production to follow different routes through the same system. With this, certain parts can be deployed to connecting machines that function at a different sequences. This is different from operation flexibility in the sense that operation flexibility is a property of the part being able to be manufactured by different operations while routing flexibility refers to a system that allows for said parts to follow a predetermined route and is a property of the system.

Purpose: A system that is routine flexible make machine loads more balanced. Hence, these types of systems are perfect for reducing the rate in which parts are produced based on machine breakdowns, sudden influxes in demand, or even the expansion of storage capacity.

**Volume Flexibility:** A system that functions under different volumes and sets is a system that runs with volume flexibility. In this case, input and output of products can be reduced or increased based on demand.

Purpose: Volume flexibility is necessary for creating a system that will function at a rate that can compete with the state of the market, whether that be raising or lowering the speed in which products are reduced.

**Expansion Flexibility:** A system that functions in a way that allows for the scaling of manufacturing resources can be defined as one with expansion flexibility. In this case, the term manufacturing resources is used loosely to refer to manpower, machines, and other manufacturing resources that can ultimately result in the increased output rate of each unit. This kind of environment allows for the "expansion" of the system in both addition of new machine and the substitution of machines that are present.

Purpose: A manufacturer that operates with flexible expansions is different in that it allows for the overall growth of the facility. Which differs greatly from other types of flexibility that focuses more-so on the ability of the machines to adapt to the constant upward and downward spikes of the market.

**Program Flexibility:** This is defined by a system that is programmed to run unattended for as long as virtually possible.

Purpose: A system that runs with program flexibility can withstand long periods of time unattended. This precludes any need for setup, which increases the overall productivity and as steady-quality of a service as possible.

**Production Flexibilities:** A system that can produce a host of parts that precludes any further need for extraneous capital investment.

Purpose: This type of system can decrease the time required for implementing certain product operations. It increases the level of the company's competitiveness by diversifying the types of parts that can be produced using a singular machine.

### 2.2.3 Types of FMS

In a broader sense, manufacturing systems can be classified based on how they are operated, the machines used, and the overall flexibility. The following criteria are used when classifying FM Systems

**DEPENDING UPON OPERATION TECHNIQUES**

An FMS is often categorized based on the types of operations that are required or performed in the facility.

**Processing Operation:** This type of operation functions in a way that transforms the product from its initial state to the final continuously over time. Here transformation of state refers to the transformation in geometry and features.

**Assembly Operation:** An operation that connects two or more parts to make a whole. This is called an assembly/subassembly and uses tools like welding, soldering, adhesive bonding, press fitting, rivets, brazing, etc.

**BASED ON NUMBER OF MACHINES**

An FMS system might also take a form depending upon the number of machines involved.

**Single Machine Cell (SMC):** Depicts machines that perform a programmed sequence unattended. SMCs, by default, are programmed to dispense unique parts, react to manufacturing plans efficiently, and allow for efficient expansion.

**Flexible Manufacturing Cell (FMC):** A force of two or three workstations tasked with the dispensing of products connected to an external material handling system, which is then linked to a station that allows for the load and unload of said products that can function simultaneously.

**Flexible Manufacturing System (FMS):** This includes a set of four or more workstations (often CNC machine tools) connected to a handling system and powered by a distributive computer program. These systems also include support workstations that are not directly linked to the production of the product.

**BASED ON LEVEL OF FLEXIBILITY**

To add a better understanding on the flexibility of an FMS, two different classifications are described below:

**Dedicated FMS:** A dedicated FMS operates only on a singular set of part styles. The design of the product is completely fixed, with optional specialization processed to make the operation more efficient.

**Random Order FMS:** To compliment Dedicated FMS, there are Random Order FMS. These are able to handle a substantial number of similar part configurations. In order to accommodate for these different parts, a random order is made to be even more flexible than the dedicated system, and is powered by a sophisticated computer control system that is able to withstand the complexity of the processing parts.

### 2.2.4 Components of FMS

So far, there has been a lot of discussions over the type of flexible manufacturing systems. However, the matter of its components have not been addressed so far. An FMS is made out of several types of components: workstations, material handling and storage systems, and a computer system that controls the flow of work. One also has to consider the labor required to operate and manage the system, despite the fact that these systems are often highly automated.

**WORKSTATIONS**

The first main component of an FMS system is its workstations. These are processing or assembly equipment used to accomplish tasks set out for said system. For the most part, these often take the form of CNC machine tools. However, there are other machines that can be applied to the same practices. Types of workstations often found in flexible manufacturing systems include:

- **Load/Unload Stations:** These workstations make up a large part of the FMS and the rest of the facility. It is where input parts are sent through and output (finished products) stored. The process in which this happens can be both automated by storage and retrieval systems or manually operated by a worker. These should include a point of data entry that allows the operator to get important instructions from the system such as which part has to be loaded, what fixtures and so on.
- **Machining Stations:** CNC Machine tools make up a large part of FMSs. The most common is the CNC machining center, which includes automatic tool changing and tool storage features. CNC machining centers are also compatible with palletized work-parts, CNC tools, and have the capacity for a distributed numerical control.
- **Other Processing Stations:** Other processing stations are used to conceptualize FMSs. An example of processing stations include the metal fabrication process,

which includes press-working operations that can punch, shear, bean, and form certain parts automatically — erasing any need for labor-intensive work.

- **Assembly:** As manufacturing practices are propelled into the future, more labor-intensive jobs are being replaced by automated industrial robots. For example, assembly systems are developed to perform tasks that follow a certain sequence and motion pattern.
- **Other Stations and Equipment:** There are multitudes of other machines that are incorporated into certain FMSs. For example, there exists inspection systems that are able to confirm the quality of work and ascertain that everything had been done as programmed

Outside of what is mentioned, there are also stations focused on cleaning, pallet placements, delivery stems, and centralized chip removal systems that are installed and used in a flexible manufacturing system.

## MATERIAL HANDLING AND STORAGE SYSTEM

The second component that accompanies the workstations used in the FMS are material handling and storage systems. They are mainly concerned with random, independent movement of work parts between stations. In this case, it is necessary that parts are able to move from one machine to another, which supports routing flexibility and makes it easier to use alternative stations when one has broken down or is busy with another task.

**Material Handling Equipment:** The transportation of parts occur in several levels. The primary handling system is the system that establishes the general layout of the FMS and is in charge of moving parts between the workstations in the facility. The secondary handling system includes the devices that transport devices, pallet changers, and other mechanisms within the workstations.

**FMS Layout Configurations**: The material handling system is also in charge of establishing the layout of the FMS. These can be divided into five different categories: the inline layout (machines and handling system are laid out in a straight line); the loop layout (workstations are laid out in a loop along with the handling systems); the ladder layout (similar to the loop layout, except there are straight sections where machines can be moved or transported in a different area); the open field layout (which is a mixture of loops and ladder systems that can handle the processing of large parts families); and finally there is the robot-centered cell (the application of one or more robots that are equipped with tools that make them suitable for the handling of prismatic parts.)

## COMPUTER CONTROL SYSTEM

Finally, at the center of all flexible manufacturing systems, is a distributed computer system that is interfaced in the workstations, handling systems, and other hardware components. For the most part, these computer control systems include a central computer, microcomputers that control individual machines and other components. It is the central computer that coordinates the operations that the components are tasked with and smooth out the overall operation. Specific functions that the computer control system is involved in includes:

- **Workstation Control:** In an FMS that is fully automated, each workstations are in some form of computer control. For example, machining centers are controlled by CNCs.
- **Distribution of Control Instructions to Workstations:** In order to task stations with processes, part programs must be downloaded to them. This requires central intelligence to coordinate the events at individual workstations.
- **Production Control:** The central computer is also in charge of the part mix and the rate in which said parts are loaded into the systems. The production control provides the operator assistance by providing instructions and routing an applicable pallet 10 to the loading and unloading stations for the desired part to be manufactured.
- **Traffic Control:** The computer is also in charge of the traffic inside of primary handling systems.
- **Shuttle Control:** As for the secondary handling system, these are synchronized with the operation via a control function inside of the computer program.
- **Work-piece monitoring:** The status of the pallets in the primary and secondary handling systems are also kept inside of the computer program, which will allow one to check- in on the work progress thus far.
- **Tool Control:** Another feature includes the life monitoring and the location of the tools involved with the stations, allowing the operator to ascertain the life-expectancy of said tools and whether or not there are any complications.
- **K Performance and Monitoring and Reporting:** One of the programmed sequences inside of the computer control system is to collect the data on the operation and the general performance of the FMS.
- **Diagnostics:** In general, the operator should be able to efficiently ascertain any malfunctions or other problems within the workstations using the computer.

**HUMAN RESOURCES**

Outside of the machinery and computers, there are still tasks that are operated by human labor. These functions include the loading of raw parts into the system, the unloading of said parts, the settings and changes of tools, the programming and the operating of the computer system, and the overall management of the system.

## 2.3 Digital Twin Communication

In this section, we review the available protocols for real-time data transfer for the realization of Digital Twins. Although, for the context of the thesis the selection of the protocol is dictated by the priorly implemented protocol by the supplier of the manufacturing system, the author makes use of this opportunity to research the application layer protocols currently available at the time of writing for in establishing how the drawbacks of the current implementation can be mitigated as part of future work or other related work in the same field. Further, only a brief introduction relevant to the thesis is provided and the information is in no ways intended to be a comprehensive one.

### 2.3.1 Open Systems Interconnection Model

The OSI model is a conceptual model that standardizes communication that occurs in a system regardless of its principal structure and technology. It follows the 'divide and conquer' approach to fragment the communication system into abstraction layers with a goal of interoperability of standardized protocols.

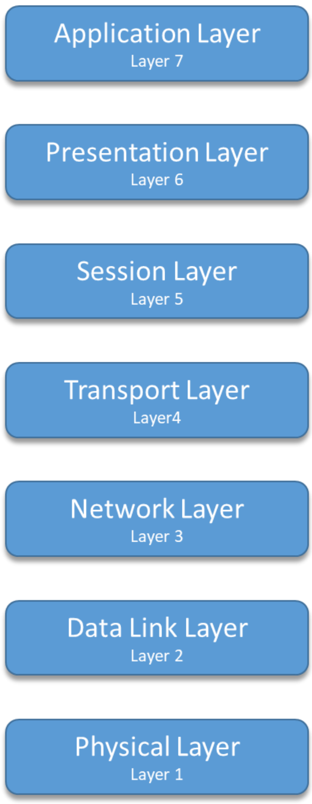


***Figure 4.*** *OSI Model*

The OSI model shown in Figure 4 is characterized by seven layers with each layer serving the layer above it and being served by the layer below it.

**The Physical layer** is responsible for the physical connection of the communicating devices. Data here are in the form of bits (0's and 1's). The main functions of this layer are bit synchronization, bit rate control, physical topologies and transmission mode. E.g. Ethernet cables, connectors, etc.

**The Data Link Layer** is responsible for packaging the bits into data "frames". It also provides physical addressing that adds the MAC addresses of the sender or the receiver in the header of the frame. Besides, it is also the performs flow, error control for intra-network communication

**The network layer** is responsible for routing data between different networks. If the both the sender and receiver are in the same network this layer is not as important. It plays an important function of selecting the best route to transmit the packet (shortest path).

**The transport layer** is the layer that is responsible for end-to-end communication between the communicating devices. On the senders side this includes breaking the data into segments before sending it to the network layer and reassembling on the receiver's side. The transport layer does flow and error control for inter-network communication.

**The session layer** is mainly responsible for establishment, authentication, maintenance and termination of sessions between communicating devices.

**The presentation layer** is where data is translated, encrypted (decrypted) and compressed (decompressed). Devices communicating may use different encoding techniques and hence it is the presentation layer that translates the data to a form that is understandable by the receiving device. It can be seen as the layer that "presents" the data to the application layer.

**The application layer** is implemented my network applications that interacts with the end-user and produces the data to be transmitted over the network. For example, a web browser uses the application layer HTTP protocol while rendering web pages over the internet.

### 2.3.2 Communication Design Styles: SO vs ED

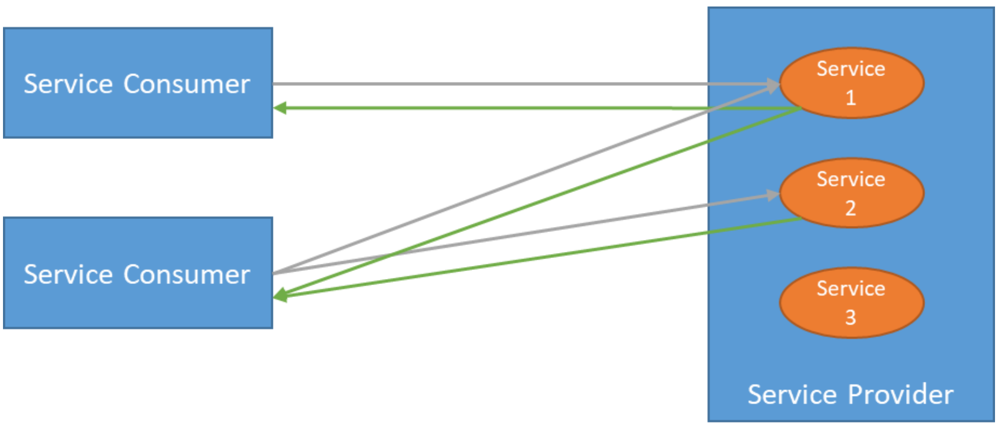


***Figure 5.*** *Service Oriented Communication*

An application designed using a service-oriented communication makes services available for use by other applications, through a communication protocol over the network (Figure 5). A service in this regard can be any basic functionality that provides is a discrete unit of information that can be consumed by service consumers independent of platform, vendor or technology. SOA is based on the traditional request/reply mechanism, wherein a service is consumer requests the service provider for the service, which then provides the service.

Principal characteristics of service-oriented communication are:

- **Loosely Coupled Interactions:** Loosely coupled services mean that the interactions between services are most often minimized to the point where they are only aware of their existences.
- **One-to-One Communication:** A service consumer requests a service provider for the service and the communication is bi-directional, one at a time.
- **Consumer-initiated:** The communication is initiated by the service consumer.
- **Synchronicity:** The service provider responds to the service consumer as and when the request is made.

Event-Driven communication are is a design style wherein communication is triggered by the occurrence of events (Figure 6). Events here can be seen as any significant variation in the state of information. The event is often notified to an event broker that usually handles the subscription for the event. The broker then notifies all of the subscribers of the event.

Principal characteristics of Event-driven communication are:

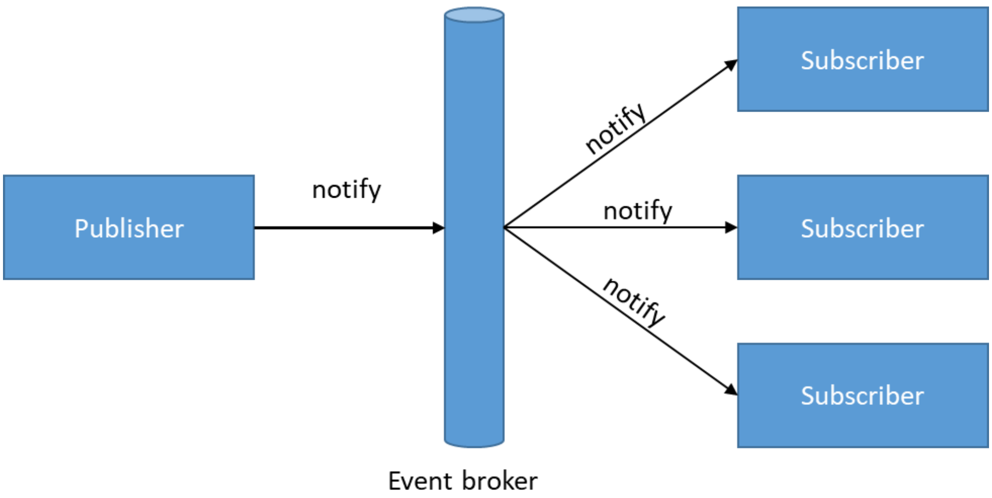


***Figure 6.*** *Event Driven Communication*

- **Decoupled Interactions:** Event publishers often do not acknowledge the existence, availability or state of event subscribers.
- **One/Many to One/Many communication:** One or more publishers may publish events that are subscribed by one or more subscribers.
- **Event-initiated:** Events trigger the start of communication.
- **Asynchronicity:** Due to the decoupling of subscribers from publishers, event driven communication is generally based on the asynchronous publish-and-subscribe pattern.

### 2.3.3 Representational State Transfer (REST)

REST, or Representational State Transfer, was initially coined by Roy Fielding as an architectural style for interactions between computer systems on the web [25]. A common misconception is that REST is a protocol, when it is only an architectural style. A protocol, such as the likes of HTTP, TCP, etc. represent a standard that defines the rules, syntax, semantics, choreography of message exchanges while and architectural style like REST specifies certain architectural framework through which a meaningful communication may take place. This framework, in the case of REST translates to the concept of resources being made available via URL, a vocabulary for its manipulation and so on, as we will delve into now.

Roy in his doctoral thesis [25], defines six principles of REST.

- **Client-server:** REST follows a client-server architecture that facilitates the separation of concerns of both clients and servers. While clients are not concerned with resource storage, thus improving its portability, servers are also not concerned with use of the data in any manner, thus allowing for its easy scalability.

- **Statelessness:** During communication, the states of the client is not known by the server and vice-versa. Any information necessary for successful communication is contained within the request, either in the query URL, parameters, body or header. Neither is a session state maintained by the server. If a resource access requires authentication, a client must authenticate itself on every request.

- **Cacheable:** If a response to a request is either implicitly or explicitly labelled as cacheable, the client reuses the response data for equivalent future requests.

- **Uniform Interface:** REST provides a uniform interface irrespective of the type clients to facilitate following the same style to speak to the server. Each message exchanged is self-descriptive in the sense that they include adequate information for it to be able to be processed.

- **Layered System:** In such a layered system, the visibility of the client is limited to the layer with which their interaction takes place. This enables for enforcing security policies and easy extensibility and scalability.

- **Code on Demand:** This is an optional constraint that allows for downloading code from the server for the purpose of extending its functionality.

REST mainly uses HTTP protocol or in other words, it can be said that HTTP protocol follows the REST architectural style. Thus the below discussion also applies to the HTTP application protocol.

**Resource and Resource Identifiers:**

Information that is made available via the RESTful system is termed as resource and it can be anything such as a document, image, a webpage, IoT sensor data, a collection of other resources and so on. Resources are identified by uniform resource identifiers (URIs)

**Client-Server Communication:**

In REST architecture, communication between clients and server takes place, by clients placing a request and the server subsequently responding. A client request generally comprises of the following:

- a HTTP verb, that defines the kind of operation to be performed
- a header, that describes the request
- a path that leads to a resource
- an optional message body that contains information

There are 4 HTTP verbs that are analogous to the CRUD terminologies that allows to create, read, update and delete resources. They are:

- POST – that allows for **creation** of a resource
- GET – that allows for **retrieval** of a resource
- PUT – that allows for **updation** of a specific resource
- DELETE – that allows for **deletion** of a specific resource

It must be noted that these associations for HTTP verbs were not mentioned in Roy's dissertation. In that sense, if an HTTP POST is decided to be used for resource updation instead of PUT, the application is not be any less RESTful as long as it does not violate any of the six principles.

**Headers and Accept Parameters:** An Accept field in the header of the request specifies the type of content that it is able to receive. For example, a website resource would be specified by the type "text/html".

A server responds with a content-type included in the header of the response. This content-type, typically one of the types specified in the accept field of the request, notifies the client the type of the data sent.

**Response Codes:**

Server responses contain status codes that informs the client on the success of the operation. Table 1 sums up the most important status codes.

***Table 1.*** *HTTP Status Codes*

| Status Code | Description | Meaning |
|---|---|---|
| 200 | OK | Standard response for a successful request |
| 201 | Created | Standard response for the creation of a resource (POST) |
| 202 | Accepted | The request has been accepted for processing. |
| 204 | No Content | Standard response for a successful request where nothing is returned as a response. |
| 400 | Bad Request | The request is not syntactically correct or another client error |
| 403 | Forbidden | The client has no permission to access the requested resource |
| 404 | Not Found | The resource could not be found, either deleted or does not exist yet |
| 500 | Internal Server Error | A generic answer for an unexpected error. |

### 2.3.4 Application Layer Protocols

**Hyper Text Transfer Protocol (HTTP)**

HTTP is the application layer protocol that is most commonly used by RESTful systems. It defines the set of rules for exchanging text, graphic, images, video and other multimedia content on the world wide web. The earlier discussion on RESTful architecture covers a brief overview on HTTP request and responses.

**WebSocket**

Websocket is a full-duplex, communication protocol that happens over a single TCP Connection. The Internet was not originally conceived to be dynamic, but rather to be a web of static HTML pages. As time progressed, there was a push towards richer and dynamic content.

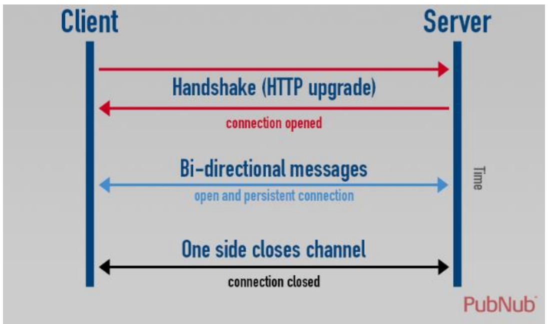


***Figure 7.*** *Websocket communication* [26]

As the web was largely built on the request/response mechanism of HTTP (Figure 7). This means to say that when a web application wanted to access any resource on the internet, an HTTP request is sent to the server that hosts the resource, which acknowledges it and sends back the response. There was clearly, a need for a better mechanism for frequently updating dynamic content to avoid round trips to the server and back without the overhead of HTTP for low latency applications. Several attempts were made in this regard, namely HTTP Polling[1], HTTP long-polling[2] and XHR streaming[3]**.**

However, all the comet[4] technologies inherently used HTTP and were susceptible to overhead present in HTTP and real-time applications required a better alternative as there wasn't a way to create a persistent low-latency connection that could be initiated by either applications exchanging information.

The HTML5 Specification presented websockets that provided a means to build scalable and real-time web applications. The client application establishes a connection known as an WebSocket handshake which is a regular HTTP request with and Upgrade Header that expresses its interest in establishing a websocket.

The server then agrees to the upgrade (if it supports it) with an upgrade header in its response. With the completion of the initial handshake, the HTTP connection is replaced

[1] Simple polling meant HTTP requests are sent at regular intervals which receives an immediate response
[2] In long-polling, the web application sends a request and keeps it open for a period of time and the server responds when it has new information
[3] XHR Streaming maintains an open connection between the two parties exchanging information but maybe buffered by intermediate firewalls as its HTTP based which futher increases the latency
[4] Umbrella term for long held HTTP requests that allow the server to push information without explicit requests from the client

by a websocket connection that uses the same underlying TCP/IP Connection and uses the same ports (HTTP: 80/ HTTPS: 443)

**Message Queuing Telemetry Transport (MQTT)**

MQTT is a light-weight messaging transport protocol that uses the publish/subscribe mechanism for Machine to Machine communication. The protocol was developed with special importance for operation in constrained environments where a minimal hardware and network footprint is required.

**The Publish / Subscribe Mechanism**

As mentioned earlier, the MQTT protocol uses the even driven mechanism, the specifics of which are generalized in secion 2.3.2 .

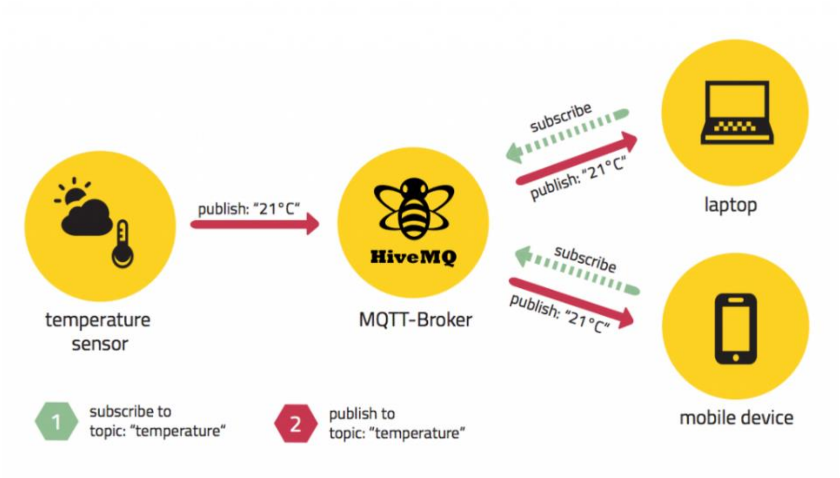


***Figure 8.*** *MQTT communication* [26]

Briefly the MQTT PubSub mechanism (Figure 8) can be summarized as follows:

- Spatial decoupling of the publisher and the subscriber, the clients only need to know the hostname /IP of the broker
- MQTT uses message "topics" to determine the subject of the message (similar to ROS[5]). Topics are "utf-8" strings that have a hierarchial strucutre using forwards slash "/". Clients subscribe to one or more topics of interest. The protocol enables topics to be created on the fly, as the broker accepts valid topics without any prior initialization.
- Although in most applications real-time deliver of data is preferred, if necessary the broker can store messages for clients that may be temporarily unavailble for whatsoever reason. Two prequesite for such a behaviour is the client should be

[5] Robot Operating System (ROS) is an open source collection of software frameworks for robot software development

connected with a persistent session and subscribed with a topic with a QoS greater than 0.

- MQTT has asynchronous operational flow although some libraries have synchronous APIs

**QoS**

There are 3 Quality of Service Level offered by MQTT

- Level 0: Utmost Once where the message will neither be acknowledged by the reciever nor be store and redelivered by the sender.
- Level 1: Altease Once where the message is guaranteed to be delivered atleaset once but could may be more.
- Level 2: Exactly Once which guarantees that the message is receive only once. This is the slowest of the 3

**MQTT over websockets**

The implementation of MQTT over websockets can render every web browser a MQTT capable client. This is can be useful as browsers are the defacto tool that people use to acesss information.

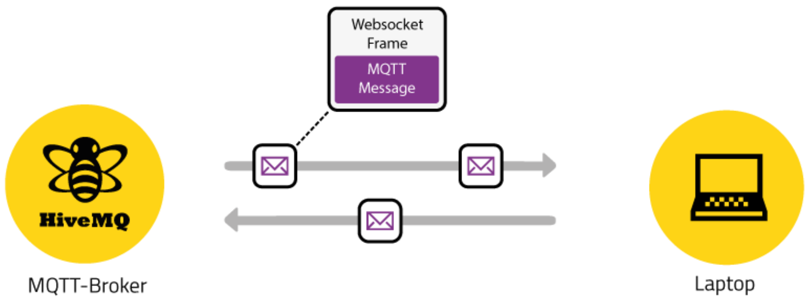


***Figure 9.*** *MQTT over websocket communication* [27]

In this scenario, MQTT messages (including headers and payloads) are sent via websockets as frames that have only a 2 byte overhead (Figure 9).

**OPC Unified Architecture (OPC-UA)**

The OPC UA in itself is a standard developed by the OPC foundation that evolved from its predecessor, the OPC (OLE for Process Control) so as to better suit the needs of industrial automation with emphasis on it being open, cross-platform, service-oriented and having a robust security model.

The Classic OPC required a Microsoft Windows OS to implement COM/DCOM[6] server functionality. However, this is not the case anymore as the OPC UA, in the transport layer (OSI) uses either a binary TCP protocol (URL: *opc.tcp://Server*) or web services (URL: *http://Server* ), e.g.: SOAP/XML over HTTP.

The binary protocol offers lesser overhead over its service oriented counterpart as it requires minimum resources without XML parsers for example and uses a single TCP port. In comparison, the web services use standard HTTP ports and has more overhead owing to the SOAP nature of messages transmitted.

The OPC UA models the architecture as a Client /Server with each server that waits for one or more clients to make a request. In the client-server model as shown in Figure 10, there are already mechanisms for subscriptions so that a client can go in to a server and ask for data streams or event streams from a server.

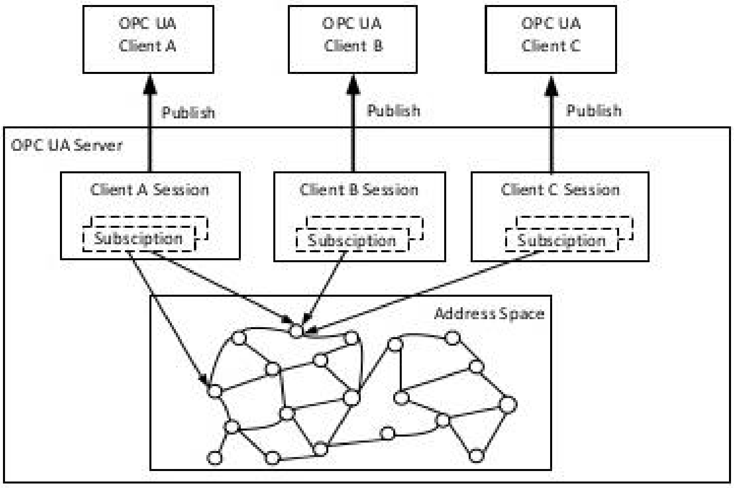


***Figure 10.*** *OPC UA client-server model* [28]

However, if there are a lot of clients they all get individual subscriptions which is still okay for smaller applications that have dynamically changing interests, for e.g. HMIs. However, for large scale applications this cannot be feasible.

As such, the OPC UA has extended its capabilities in the form of a Publish-Subscribe Model (Part 14 of the OPC UA Specification) recently.

In the Pub-Sub model (Figure 11), the publishers send messages to a Message Oriented Middleware (MOM) known as a Message Broker without knowledge of the existence of any subscribers. Similarly, subscribers make the MOM aware of its interests without

[6] While COM is a standard for inter-process communication between software components DCOM is an extension of COM that allows such communication between computers on a network. Both are proprietary Microsoft technologies for Microsoft Operating System

knowledge of the existence of Publishers. The specification defines the existence of two variants of the MOM;

- a brokerless form, that leverages the network infrastructure to route datagram-based messages, UDP multicast for example, and
- a broker-based form where the Subscribers and Publishers use standard protocols such as AMQP or MQTT to communicate with the Broker. The Broker in this case has the responsibility of exposing specific queues (topic, nodes) to which the subscriber listens. Additionally, the broker may also translate messages from one form to the other if the Publisher and the Subscriber uses different messaging protocols.

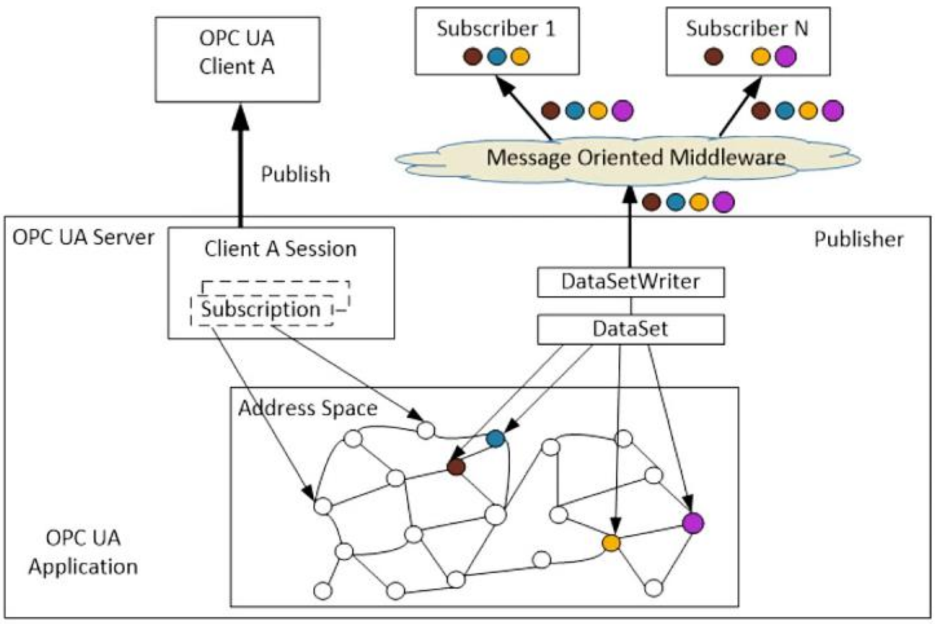


***Figure 11.*** *OPC UA pub-sub model* [28]

OPC UA uses X.509 certificates, Kerberos or user/password for secure transfer of information at the application level.

## 2.4 Knowledge Representation and Reasoning

Several methods have developed for representing domain information. Databases and Ontologies are the most popular of them and the latter has been of increasing popularity in the last decade.

### 2.4.1 Database or Ontologies?

A database system can be seen as mediators between physical systems that hold information and humans who want to use it. There is ample literature on databases since decades ago when they replaced the then traditional file-based systems. The relational model proposed by Codd in 1970 was arguably the most important development and several new technologies emerged as a consequence [29]. Several methods have tried to improve the shortcomings of relational database model in terms of its semantic representation using Description Logic (DL) and First Order Logic (FOL) [21][22]. However, the relational model remains to be the de facto standard for the most part and the predominant choice for most organizations for information representation.

Ontologies originated in research areas of metaphysics where philosophers used to describe the existence and nature of reality. Ontologies have been pursued by many in various fields and domains to simplify complex, unstructured and heterogeneous information to simple, useful data and knowledge. The case is no different in the manufacturing domain. A simple definition of ontology is that of Agarwal [32]who states that "an ontology is, therefore, the manifestation of a shared understanding of a domain that is agreed between a number of agents and such agreement facilitates accurate and effective communications of meaning, which in turn leads to other benefits such as inter-operability, reuse and sharing".

The past decades have seen keen development of ontologies due to its involvement in the semantic web [33]. It has also seen widespread adoption in the manufacturing domain [35][36][37]. In the next subsection, we compare the two on certain grounds relevant to the thesis to make an informed decision in going ahead with one.

### 2.4.2 Close World (CWA) vs. Open World Assumption (OWA)

Closed World Assumption, in simple terms, implies that what is not known is false while Open World Assumption implies that what is not known is simply known. In other words, OWA necessitates that in order for a statement to be true, it be explicitly written.

For example, consider a statement that says "humans eat food of plant origin" exist as true. If we were to ask "Do humans eat food of animal origin?" under a closed world assumption the answer would be a No. However, an open world assumption would be "I don't know".

Further, continuing with the example of "humans eat food of plant origin", let's say another statement exist that "humans can eat food of only one origin". If we were now to say that "humans eat food of animal origin", a closed world assumption reports an inconsistency while an open world assumes that "plants and animals are the same". This is known as Unique Named Assumption (UNA)

### 2.4.3 Database vs. Ontologies

Databases are normally focused on data storage and can be considered as effective data warehouses that emphasize on data storage and retrieval. Ontologies on the other hand specializes in adding semantics (meaning) with a focus on meaningful communication, interoperability, re-use and servers as a intuitive link between human that needs data and machines that contain it. In that sense it is most likely that databases are created from the ground up whereas an ontology creator can make use of existing ontologies or build on them.

Several works compare the pros and cons and is not of interest in this thesis [7][8]. Nevertheless, the factors that led the authors to choose ontologies is important and is discussed in Section 4.5.

That being said, ontologies comes with their disadvantages as well in terms of efficiency and performance among others. However, their relevance in the context of the paper is little.

### 2.4.4 Web Ontology Language (OWL)

Implementation of Ontologies requires modelling them in one of many available languages (Figure 12). Several works have compared the different ontology languages and it is beyond the scope of this text [30][31].

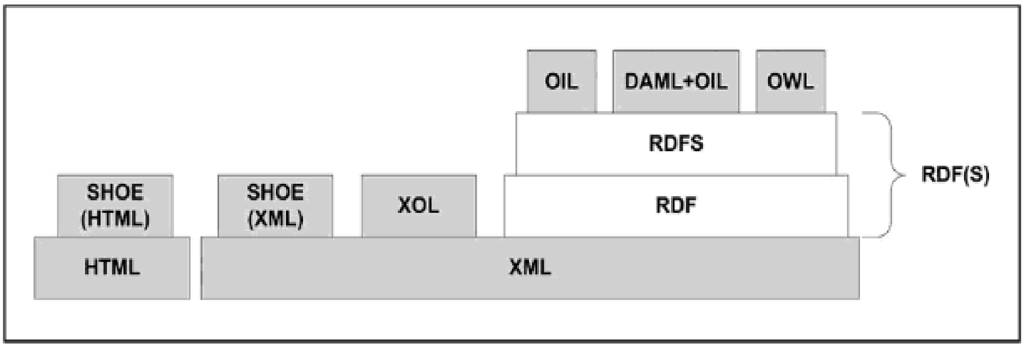


***Figure 12.*** *Ontology markup languages and their relationships* [38]

The most popular ontology language is the web ontology language (OWL) which can be seen as an extension to RDF (Resource Description Framework) and RDF Schema. The development of RDF was an attempt to extend the syntactic web to a semantic web, by bringing it the power of description logic, so as to make the web (not just HTML pages) more understandable to machines and to support information creation, use and exchange across different parties. The RDF Schema language can be seen as the first attempt to support ontology. However, it couldn't completely bring out the reasoning power of description logic. Some of the limitation of the RDFs have been identified as local scope of properties, disjointness of classes, Boolean combination of classes, cardinality restrictions and special characteristics of properties [39]. For this reason, OWL can be seen only as a syntactical extension of RDF and RDFS.

To solve these problems, OWL was defined as 3 different sublanguages, each of which are fulfill different requirements. They are:

**OWL Full** that has no restriction on the use of OWL vocabulary and is fully upward compatible RDF syntactically and semantically albeit has computational complexity

**OWL DL** in which only a subset of constructors from OWL and RDF can be used. Although this permits efficient reasoning compatibility with RDF is lost.

**OWL Lite** which further restricts OWL DL that is easier to implement and learn at the cost of decreased expressivity.

The detailed semantics and syntaxes of the language is beyond the scope of this literature but OWL builds on RDF and RDF schema and uses its XML syntax. While this is not most readable and not the only representation, this is quite the most popular.

The three main concepts in OWL are:

- **Class elements:** Classes define an object or a concept, for example class *associateProfessor*
- **Individuals:** Instances of classes, for example *JohnSmith*
- **Property elements:** Properties relate Individuals. There are two kinds of properties. Object properties which objects (instances of classes) to other objects, for example isLecturedby, supervises, etc. and Datatype properties which relate objects to datatypes, for example, title, age, etc.

Remaining concepts such as property restrictions, special properties, Boolean combinations, enumerations, etc. are not covered here and the interested reader is encouraged to read relevant material [39].

### 2.4.5 SPARQL Protocol and RDF Query Language (SPARQL)

SPARQL is a W3C recommendation for querying RDF data. It is a recursive acronym that stands for SPARQL Protocol and RDF Query Language. SPARQL queries may consist of triple patters, conjunctions, disjunctions and optional patterns. There are generally four different type of SPARQL queries:

- **SELECT** query that returns the variables that matches with the query pattern
- **CONSTRUCT** that lets you create RDF triples or graphs
- **ASK** that returns a Boolean values in response to asking for matches
- **DESCRIBE** that describes the resources matched by the query pattern

### 2.4.6 Protégé and Apache Jena Fuseki

Protégé is a free open source Java-based software, developed by the University of Stanford for development of ontological knowledge bases to represent domain information. It has an active community of over 300,000 users [40] and is the leading ontology development tool that supports the latest OWL 2 standards and specifications from the W3C. This software provides for a user-friendly GUI for creating the ontology rather than having to develop them manually using OWL.

Apache Jena Fuseki is a SPARQL server that serves RDF data over HTTP [41]. It provides a REST style interaction with the RDF data that is necessary for presenting and updating RDF triples during runtime using SPARQL and HTTP.

## 2.5 Learning

Learning, from an educational standpoint, reflects the link between the contents of a course and the pedagogic methodologies on one hand and a well-equipped labour force on the other. Therefore, having an effective learning method and environment is of vital importance especially in professional courses at the university level. Harris and Schwann [42] mentions three perspectives of viewing learning: as a product, which accentuates the outcome of the learning experience; as a process which focuses on the developments during the learning experience and as a function which underlines important aspects of learning such as motivation and memory that makes behavioural changes possible in individuals.

The important of the three, as far as the context of the thesis goes, as a process, can take place in an active or a passive mode. Passive learning is majorly by means of lectures with no or very little involvement of the students through exercises. Students who go through a passive-learning process were said to have been at least been 1.5 times more likely to fail than students who had access to direct or even contrived experiences [43]. These studies further proved that a more modernized “practice-based” curriculum, that allowed the student dive into experiences were much more effective, especially when applied to students who studied engineering. Studies have shown that cognitive outcomes of students are also higher when an active approach is followed as opposed to a passive one [44].

### 2.5.1 Learning Theories

Learning theories are concepts that describe the learning process in terms of how individuals assimilate, act and contain knowledge [45]. In literature, various learning theories are discussed and this section presents a literature review of only what is relevant in the context of the paper, namely behaviourism, cognitivism and behaviourism.

Behaviorism equates learning in individuals with observable changes in behaviour produced in response to an environmental stimulus promoted through repetitive stimulation and reinforcement [28][29] .An important aspect of the behaviorist approach to learning is the stimulus and the response and how their association is created and sustained. Behaviourists claim that responses to stimuli that are reinforced are likely to part of the learnt behaviour [48]. They call this learning as conditioning and points out two different types, classical and operant conditioning. In classical conditioning, an association between an involuntary response and a stimulus is created as in the case of Pavlov's dog experiment while operant conditioning involves creation of an association between a voluntary behaviour and a consequence. Further, in operant conditioning the learner is lured to learn by use of rewards and punishments but no such persuasion exists in the classical counterpart. However, the learner assumes a passive role and is characterized as a reactive agent responding to environmental stimuli as opposed to an active exploratory interaction with it. It is important to say here that while some behaviourist view exhibiting behaviour as an activity and thus an active role from the part of the learner [49], for most researchers and in the context of this paper, the behaviourist learner is perceived to play a passive role as the learner is characterized as a reactive agent responding to environmental stimuli as opposed to an active exploratory interaction with it. In sharp contrast, the teacher is expected to provide stimuli (cues) that can trigger responses so that apt reinforcements may be subsequently made. Further, behaviorists hold the view that all learning, regardless of its complexity, takes place through stimulus-response associations and any cognition is studied as an external observable behaviour as opposed to an innate value. Notable contributors the behaviourist theory are Pavlov (classical conditioning) [50], Thorndlike (connectionism)[51], Watson (behaviourism) [52] and Skinner (operant conditioning) [53].

Cognitivism learning theory came into being as the dominant paradigm in the mid-20th century as a response to behaviourism and advocated that people are rational creatures and learning involves active participation and actions that are a consequence of cognition. It places emphasis on the mental thought process that occurs behind the exhibited behavior rather than the behaviour itself [46]. It takes into account how information is obtained, structured, retained and recovered from the mind. As such, the learner assumes a very active role[48]. Similarly, the teacher is expected to provide with efficacious instruction that taps on existing mental structures of the learner and assimilates and/or accommodates new information seamlessly. However, cognitivism pays little or no attention to physical or environmental factors that leads to learning and although behavioural transformations are observed, they are only perceived as an indication of cognitive activity. Piaget [54] is a notable contributor to the cognitivism theory.

Humanism learning theory also deplored the behaviourist stimuli-response approach and espoused a learner subjective approach. The need to learn is perceived as an inherent desire in the learner and learning is most effective when it aligns with the subjective desires of the learner . While Rogers [55] described this inherent desire as an instinctive inner core, Maslow [56] called it "self-actualization". As such, the learning revolves around the learner and as such assumes an active role as opposed to the teacher that takes upon the role of a facilitator. As facilitators, they provide an atmosphere where students have the option of pursuing topics of choice to them and learn by more natural means. Thus the goal of learning is to create learner centered educational experiences that align with the natural desires. Although the role of these experiences was acknowledged it wasn't before Kolb [57] that a sound experiential theory was formulated despite having its roots from the earlier works of Dewey, Lewin and Piaget. However, Kolb's Experiential theory was not a third alternative to the behaviourist and cognitive theories, but one that integrated experience, perception, cognition and behavior from a learning perspective [56].

### Situational Awareness: A Learning Perspective


***Figure 13.*** *Endsley's three level model of Situational Awareness* [58]

Situation(al) Awareness is term that originated from the aviation domain and for the most part is associated with the operator's perception and understanding of his environment [58]. In literature, various definition and models of SA exists. Bedny and Meister presents the Activity Theory Model and defines SA as "the conscious dynamic reflection on the situation by an individual". He goes on to state that "it provides dynamic orientation to the situation, the opportunity to reflect not only on the past, present and future, but the potential features of the situation. The dynamic reflection contains logical-conceptual, imaginative, conscious and unconscious components which enables individuals to develop mental models of external events" [59]. Smith and Hancock [60] presents a model based on Niesser's perceptual cycle model summarized by Stanton [61] as "the invariant in the agent-environment system that generates the momentary knowledge and behaviour required to attain the goals specified by an arbiter of performance in the environment". However, the most widely used definition of SA is that provided by Endsley who defines it as "the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future" and presents a 3 level model for the same shown in Figure 13 [58].

The cognitive mechanism involved in SA as presented by Endsley is shown in Figure 14 [62]. It constitutes of a long-term memory and working memory that is viewed as its current activated subset. The learner's attention primarily dictates the perception and hence Level 1 SA. For example, if the learner is staring at the ceiling, the digital twin does not have the learner's attention and is most likely to miss visual cues from it. Further, this information from the environment is stored as activated mental models. Often schemas are attached to mental models that assist with immediate understanding of cues from the environment and activates scripts about associated actions facilitating rapid decision making. While schemas provide a mechanism for organization and coherent storage and retrieval of information, a script provides with a string of actions to be carried out in response to events. This ability to activate mental models and associated schemas and/or scripts based on cues from the environment is referred to as pattern matching and is learner dependent, i.e. one learner may be able to pattern-match more efficiently than another.

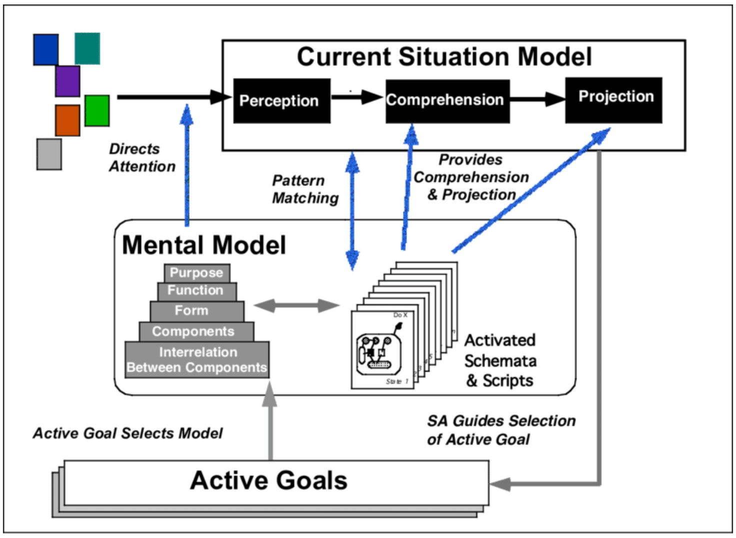


***Figure 14.*** *Cognitive Mechanisms in Endsley's Model* [62]

Two other factors as mentioned by Endsley play a role in situational awareness, expectations and goals. Expectation can be translated to anticipation of occurrence. The occurrence may be that of a process, product or even the environment and are a result of existing mental models, prior training, experience or instruction among others. Goals on the other hand is what the learner is in pursuit of. As processes unfold, circumstances change and newly perceived environmental cues may necessitate the activation of new goals in pursuit of the original goal. Switching between intermediate goals is seen a vital to the realization of the bigger goal, the larger picture.

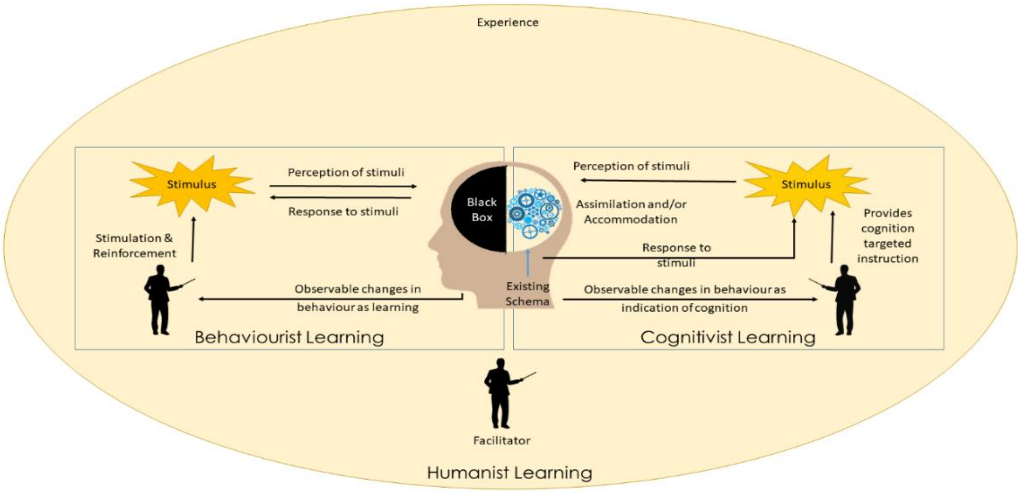


***Figure 15.*** *Learning Theories: A visual perspective*

Table 2 summarizes the discussion on the learning theories and provides a comparison between them taking into account six factors. Further, Figure 15 provides a visual perspective to the discussion. This section was by no means intended to be a comprehensive review on the learning models and theories but only a subtle introduction to relevant topics for further discussion in subsequent sections. The interested reader is encouraged to read the referenced citations.

***Table 2.*** *Summary of Learning Theories*

| | **Behaviourism** | **Cognitivism** | **Humanism** |
|---|---|---|---|
| **Learning Outlook** | Observable behaviour changes to as responses to stimuli. Learner is considered a “black box”. | Change in learner’s schemata/ mental models | Learning via reflection on experiences that are aligned with the subjective natural desire. |
| **Learning Influential factors** | Environmental conditions, stimuli. | Learning methodology/tools, Existing Schema. | Personal traits of the learner and teacher. |
| **Role of Teacher** | Active Role, Provide stimuli (cues) that can trigger desired responses and provide reinforcements. | Active Role. Provide Instruction by means of which learners are able toconnect with existing schema. | Assumes the role of a facilitator / supervisor. |
| **Role of Learner** | Assumes a passive role, “black box” | Assumes an active role. | Learner centred learning. |

| | | | |
|---|---|---|---|
| **Emphasis** | Environmental stimuli that provoke behavioural response. | Mental Processes that leads to behavioural response. Mental processinclude reception, storage, organization and retrieval of information | Individuals subjective experience. |
| **Relation with other theories** | Cognition is studied as an external observable behaviour as opposed to an innate characteristic. | Although behavioural transformations are observed, they are only perceived as an indication of cognitive activity. | Integration of experience, perception, cognition and behaviour from a learning perspective. |

### 2.5.2 Learning Goals, Objectives, Outcomes and Course Alignment

The education agendas of today's higher universities are moving towards an outcome-based approach towards implementing courses. This is part of the vision to ensure that the curriculum design shifts from a teacher centered to a more student-centered one. There are majorly three terminologies associated with the curriculum design of such courses: Learning Goals, Learning Objectives and Learning Outcomes. The idea here is not research about course design principles but to define and understand necessary concepts that would be modelled in subsequent chapters.

**Learning Goals:** Learning goals broadly defines the student's competency after taking the course. Goals can also be interpreted as a broader learning outcome aimed at providing the overview of a course, for example, “”to equip students with basic programming skills to solve day-to-day problems” Course goals are realistic and achievable but not usually measureable. For example, in the example above it is not immediately clear how basic or advanced the programming skills would be but the goal sounds achievable and realistic in a classroom or laboratory setting.

**Learning Objectives:**

The course goals help formulate the learning/course objectives. Continuing with the example above, if the learning objective is not directed towards enhancing a student's programming skill, then it shouldn't be included. A learning objective example would be Given a task of basic complexity the student will be able to develop a program in C++ to automate the task using standard coding practices and techniques with ease”. Learning objectives are course-level statements describing what the course participant is intended to be able to do upon completion of the course. They are generally less broad than the learning goals of the course.

The ABCD model [63] is one of many guides that ensures a learning objective is on point. It consists of four components:

- Audience: who the learning is intended for ?
- Behaviour: what behavior is expected from the learner in response to the learning that has occurred usually described using action verbs such as “identify”, “demonstrate”, etc.
- Condition: the circumstances under which the learner will be able to exhibit the behaviour.
- Degree: the level of mastery or expected performance by which the outcomes may be judged.

Example (Condition, Audience, Behaviour, Degree):

Given a task of basic complexity, the student will be able to develop a program in C++ to automate the task using standard coding practices and techniques with ease.

**Learning Outcomes**

The specificity increases with learning outcomes and it is an explicit statement that describes what a course participant will have achieved and demonstrate at the end of the course. A learning outcome example: “the student will be able to develop object oriented solutions in C++ to automate routine tasks”. They are specific, demonstrable (measureable) and student-centered. In the above example, it can be tested as to whether the developed solution adheres to standard coding techniques and conforms to object oriented principles.

**Assessment:**

Assessments are strategies or techniques to determine the extent to which the student demonstrates learning outcomes aligned towards the course objectives. It is a continuous process that logs the learning improvements of the student.

**Course Alignment:**

Alignment of courses is the process of mapping of the learning activities, learning outcomes and assessment (Figure 16). Assessment validates the learning outcomes against the learning objectives and presents the instructors with evidence of how well the student has learnt what the course intend them to learn. Hence, it is necessary to align the assessments with the objectives. Alignment of courses is necessary because if not, [64]the student ends up spending time on activities not geared towards the intended objectives and thereby the course goals. This means that good grades may not necessarily translate to good learning from the course’s perspective.

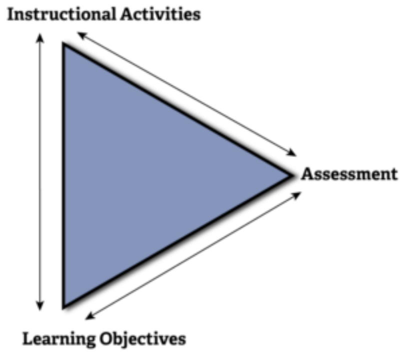


***Figure 16.*** *Bigg's Triangle of Effective Learning* [65]

### 2.5.3 Learning Taxonomies

Learning classification or taxonomies are often used to assist with the classification of learning objectives and by doing so, help with identifying the extent to which its associated skills or knowledge has been attained. Several such taxonomies have been presented in literature but the most popular among them are the taxonomy of Bloom [66], the SOLO taxonomy by Bigg and Collins [67], Flink's Taxonomy and Six facets of Understanding by Wiggs and McTighe.

Although the Bloom's Taxonomy is the most widely used, this was not originally developed with higher education in mind. It was more about selection of testing materials rather than the student's response to it which makes it limited to the events prior to measuring the learning. Although, the author is aware of a revised Bloom's taxonomy by Anderson and Krathwol [68], the author was in favour of the SOLO taxonomy as it was more suited to the context and due to its simplicity that would aid modelling the same.

**SOLO Taxononmy**

The SOLO taxonomy is an acronym for the **S**tructure of the **O**bserved **L**earning **O**utcome. True to its name, the SOLO model defines the structure of the observed learning outcome as five distinct levels of that spans across three levels of knowledge; Surface, Deep and Constructed (Figure 17).

**SOLO 1: Pre-structural Level**

At the pre-structural level, the learner has not grasped the concept and in need of help. The learner is not able to apply logic in response to cues and any responses are characterized by high closure and low consistency.

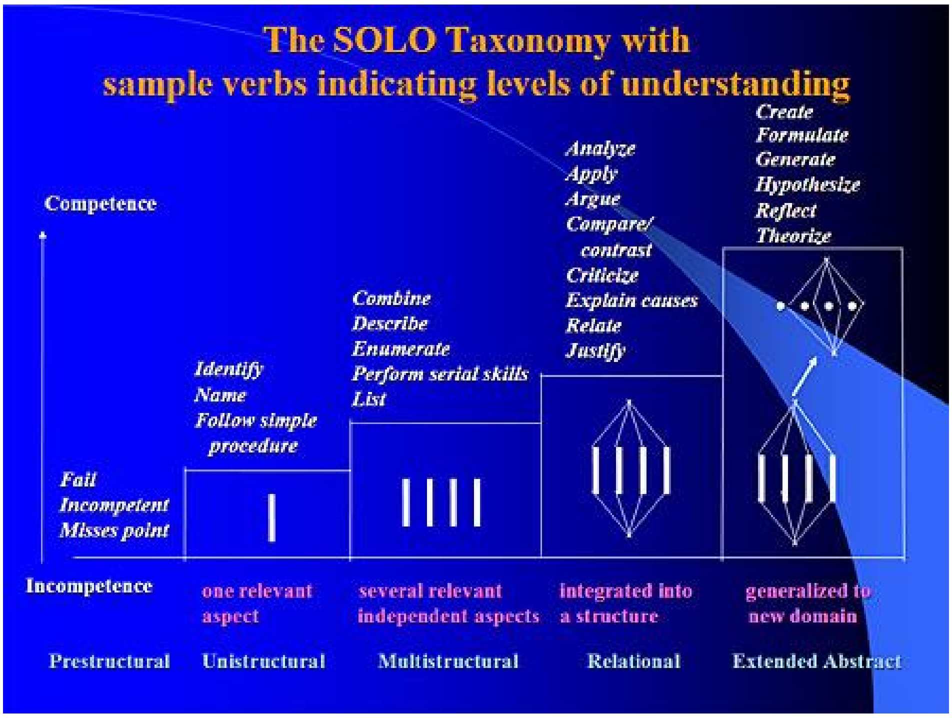


***Figure 17.*** *SOLO Taxonomy* [69]

**SOLO 2: Uni-structural Level (Surface Knowledge)**

At the uni-structural level one aspect of the concept is picked up and the student starts to make obvious connections.

**SOLO 3: Multi-structural Level (Surface Knowledge)**

A **quantitative** jump from SOLO level 2 lands the student at the multi-structural stage. Here, the student deals with several aspects but these remain disconnected. This means to say that students understand certain aspects but are not able to draw the relations between them.

**SOLO 4: Relational Level (Deep Knowledge)**

A **qualitative** leap from the multi-structural level lands the student in the relational level. In this stage, the student understand the relations between several aspects based on the his/her understanding of the aspects and how these fit together to form the bigger picture.

**SOLO 5: Extended abstract Level (Constructed Knowledge)**

In this level, the student has extended the related aspects by generalizing beyond what was given, In this thesis, SOLO Taxonomy is used to define the Learning Objectives, and to formulate an assessment mechanism to evaluate to what extent the Learning Objectives were achieved.

# 3. STATE OF THE ART

This chapter presents the state of the art research done in warranting digital twins as an educational tool in two sections. This chapter along with section 2.5 completely addresses the first research question identified in chapter 1. The first section in this chapter develops a didactic framework that the digital twin can be used and while the subsequent section attests to the framework's ability to provide concrete learning experiences by instantiating widely researched learning theories in it.

## 3.1 Digital Twin as a Learning Tool in a Didactic Framework

The digital twin allows for a homogenous mixture of theory and the application of said theory inside of the classroom. Opening the doors to similarly contrived experiences has shown significant benefits that cultivate the learning process [70]. Being able manoeuvre around machinery is a complicated task, one that requires a level of experience at handling said machinery. Lectures deliver the theory behind the machinery offering an insight as to what they do and what they can achieve. In order to excel in manufacturing, one needs to be able to understand their surroundings. The digital twin allows the student to follow the sequence of operations and fosters situational awareness. The process of monitoring the data as it passes by real time helps the student develop a better understanding as to what goes on inside of a real factory. These processes are often complicated, explained in numbers and theoretical situations inside of a classroom. The advantage of the digital twin is that it expresses real-time data, and real-life situations. The replicated manufacturing system is a high fidelity one, that is as true to life as it can possibly get. The machinations of the entire process is displayed in a user-friendly interface that inspires experiential learning. Moreover, the Digital Twin could allow automated capturing of the behavioural competence acquired by the student operator through the interface that it provides. The Digital Twin could draw inferences based on certain triggers and events that is performed by the user on the Digital Twin Interface. For example, a trigger could mean the pressing of certain buttons on the screen such as switching between various tabs or navigating between different menu items, while events could mean contextually more significant actions such as placing an order, etc. although this may be done through triggers such as pressing a button. Additionally, there could be observations that are made from time to time that provides information about the current widget in use, the current function in use, the displayed graphics, tracking mouse movements and so on. Ontological models containing a timeline of the key triggers, events and observations can be further analysed to provide insights on the operators understanding of the process. For example, the time taken to produce an order, system idle time, time wastage from searching for the right function, etc. can be scrutinized. Taking this a step further would be the ontology-based human activity recognition [46][47]. The time taken by the operator at the operator station can be monitored. For example, if there is a necessity of loading the system with raw materials the time taken by the operator to realize this (situational awareness) and feed the raw materials can be investigated to understand the state of awareness of the operator whilst also monitoring the operator's movement around the system to study any difficulties with the said task. The digital twin thus allows for the monitoring of real-time data as it passes through the system, allowing the user to obtain insights as to the process and everything that it entails. It is intuitive learning at its finest.

### 3.1.1 Didactic Framework

This sub-section aims to contribute with a methodology that promotes using Digital Twin as an educating tool. A logical architecture that shows how the Digital Twin effects the stage of the learning cycle is presented in Figure 18. The architecture places an emphasis on learning and how the Digital Twin plays its role in the process.

Attainment of learning outcomes is a standard measure of the success of an educational activity. We define the learning objectives as the attainment of a manufacturing related competencies and its associated knowledge and skill.

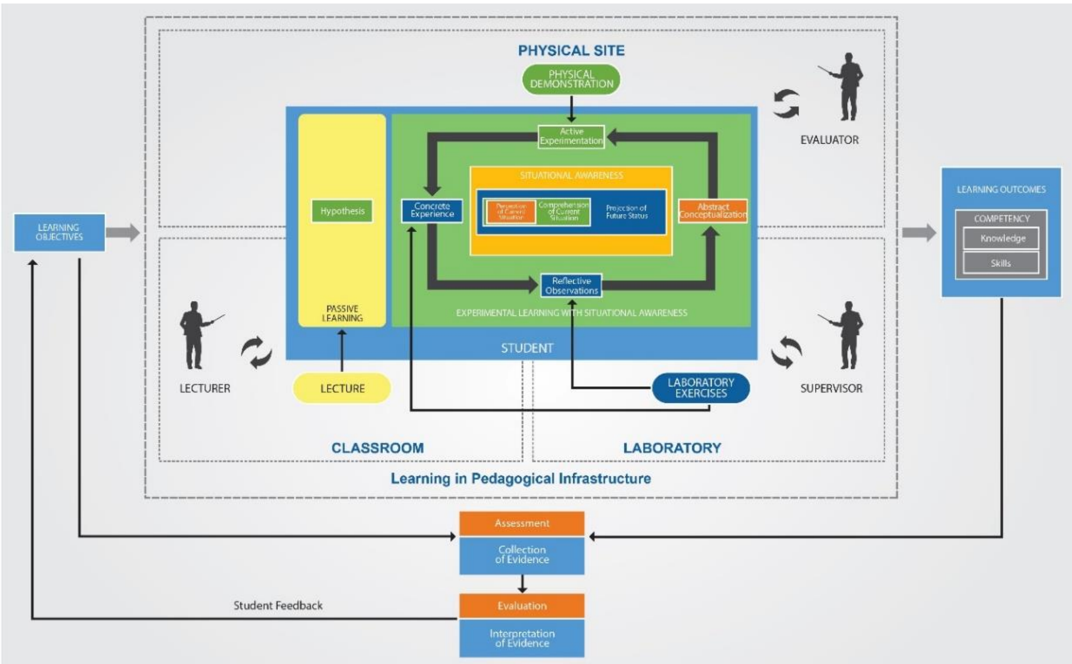


***Figure 18.*** *Didactic framework involving digital twins* [73]

**Lectures:** The student first attends an introductory lecture or a series of lectures that presents the motivation of the course and the principle of FM systems which are explained in a passive environment through presentations with the lecturer being the protagonist and students taking the back-seat. During the course of the lecture, the interfaces and the sequence of operations that must be carried out are explained as step- by-step instructions.

The main purpose of the lectures is to develop a general hypothesis which can help expedite further stages of the learning process. Here the role of the teacher is that of a lecturer

**Laboratory Exercises:** The lectures are to be followed with laboratory exercises that are carried out with the Digital Twin in a virtual configuration under the supervision of laboratory teacher. The virtual configuration decouples the Digital Twin from its physical system. Although the sense of Digital Twin is lost here by definition due to absence of its physical counterpart, the virtual configuration models and simulates the physical system in its exact detail as it would in real-life. The students are made to work on the exact interfaces as would be on the physical site via a web browser. Here the students are the protagonist while the teacher is present only in case students need help (supervisory role).

The students are divided into groups if necessary and are presented with a production requirement consisting of plurality of product variants and consequently varying process requirements. The intention here is that the student understands the principle of the FM systems via observation of creation of different product variants as it unfolds in front of them. As the entire class works on a single virtual FM system, bottlenecks are created. Evaluation of these bottlenecks by the students are important aspect of learning FM systems. Students get to see real-time how the production orders schedule the resources and how to availability of resources are optimized to facilitate flexibility in production.

In the context of experiential learning, the laboratory exercises enforce concrete experience and reflective observation enhancing the already previously developed hypothesis during the lectures. Students are doing as instructed in the lectures and in turn obtain a first-hand practical experience at per- forming a new task reinforcing the concepts learned (concrete experience). As the production orders come into fruition, students observe many insights the virtual configuration provides (reflective observation). Students use their self-judgement to develop cognizance of the unfolding processes in real-time from observation rather than doing. Impartial assessment and careful evaluation of cause effect relationships happen in this stage and they look for the meaning of things. For example, the bottlenecks may be caused due to the unavailability of either raw materials or machine tools or simply due to the excessive workload on the system. Such an analysis may be performed by only reflectively observing the material storage, tool magazine or the schedule of the machining centers respectively provided by the virtual configuration. Observations such a these helps further the understanding of the underlying principles of FM systems. Reflective observations could also be inter or intra group discussions where students reflect on their individual experiences.

Forming abstract ideas from experiential learning of a new concept may not be possible for all students. Hence, teacher intervention at this stage could be useful in connecting the learning process with the contextual theory via peer-to-peer discussions that would help reinforce the learning process taken place so far and help get rid of existing dissonances. The student draws connection between knowledge gained from the concrete experience (stage 1) and his or her own reflections (stage 2). The student understands abstract concepts like why the layout of the FM system is the way that it is among other things.

**Physical Demonstration:** After successful completion of the laboratory exercises the students are taken to the site of the physical environment that now would seem more than familiar. By now, the students are equipped with the technical know how to work on the manufacturing line. The students are given tasks similar to that of the laboratory exercises albeit without step by step instructions. Practical application of the concept is accentuated at this stage. There is also an opportunity to broaden the scope of the learning objectives by trying the acquired knowledge or skill in difference environments. For example, the configuration layout of the FM system maybe different or the interfaces could differ from what was previously encountered or the available resources could be different. In all such scenarios, the student must be able to cope up with the improved environment with ease. The teacher at this stage assumes the role of an evaluator.

**Assessment and Evaluation:** The success of learning is defined when the learning outcomes are achieved and more importantly noticeable among the students. The learning out- comes can be divided between knowledge that involves the underpinning principles of flexible manufacturing and manufacturing related skills such as the speed in operating the Digital Twin Interface for production orders, performing new tasks etc. Often the problem that arises is this context is the sound measurement of these learning outcomes. Assessment of the learning outcomes can be divided into behavioral, generic and holistic. Behavioral competencies are not directly observable but may be observed through performance [74][75]. Behavioral assessment can be used to assess the skill of the student using the Digital Twins by use of ontological models as mentioned the previous section. Alternatively, teaching assistants may monitor the individual activities of the students or student groups to see if they have difficulties navigating through the interfaces for executing production orders. In addition to behavioral competencies, knowledge facets are crucial in the development of the overall competence. This can be probed by a generic approach by means of quizzing the learner by theoretical and comprehensive questions via questionnaires, reports, exams, etc. An example of a theoretical question could be what are the steps taken to create a production order while a comprehensive question why the production sequence followed a specific order. The students could be encouraged to give feedback at the end of the course via forms or online surveys. This feedback can be used to improve the learning goals and thereby the con- sequent learning outcomes for subsequent implementations. Success at the evaluation stage would mean the attainment of learning outcomes and thereby the related competencies

## 3.2 Instantiating Learning Theories within the Pedagogical Digital Twin Framework.

As mentioned earlier, we believe that digital twins have the potential in delivering quality learning experiences. We investigate how digital twin of pedagogic nature can assist as a rich educating tool and justify the same by aligning the experiences with widely researched pedagogically sound learning theories mentioned earlier.

Figure 18 represents a didactic methodology involving Digital Twin of a flexible manufacturing system for production engineering courses at the university level. We use this as a reference to justify our proposition of using digital twins as an educating tool. In the proposed methodology, the student experiences learning in 3 different environments in the pedagogical infrastructure namely classroom, laboratory and physical Site. We study each of the environments in terms of the environment characteristics and the learner experience and draw a parallel to the learning theories. Further, we also see how being aware of the environment can further augment the learning experience.

**Passive learning in classroom:** The proposed classroom learning takes place via lectures and presentations as a means to introduce students to the concepts related to flexible manufacturing system. This is can be seen as a part of direct instruction (DI) model as presented by Siegfried Englemann and Carl Bereiter in the late 1960s [76]. Here it is important not to equate the lecture alone as Direct Instruction as that not what Direct Instruction is [77]. Instead, Direct Instruction is an instructional approach and the lecture is perceived as a medium of "directing" the instructional process.

The lecturer progresses through the slides, as the sequence of operations to be performed on the digital twin is presented systematically. Alternatively, instead of using presentations the digital twin of the manufacturing system maybe directly introduced in the classroom and operated by the instructor. Either way, the classroom (environment) instruction can be seen as the presenting of visual aids such as slides or digital twin (stimuli) and the reaction from students when asked questions (response). These responses can be analysed to substantiate behavioural changes equated to learning (behaviourism).This generates a hypothesis about the functioning of the FM system which from a SA point of view can be seen as an expectation, which we term as "hypothesis as expectation". It is also to be noted that the students(learner) here play a passive role as opposed to the lecturer, the protagonist, which is a typical trait of the behaviourism.

**Experiential Learning in Laboratory and Physical Site:** The laboratory exercises takes place with the help of the digital twin on which the students are expected to release production orders. The students play an active role here as opposed to the classroom and the exercise is expected to foster a concrete experience and the unfolding production processes provides with reflective observation reinforcing the priorly developed hypothesis. This correlates with the humanist theory of learning which is learner-centered and states that people have a natural tendency to learn. The experience (here provided by the digital twin) acts a catalyst that promotes the learning process. Situational Awareness(SA); A learning Outlook: The digital twin is also proposed to instil a state of situational awareness by allowing the learner to follow the entailed process in its finest detail, real-time. Endsleys three level model [58] is the model used to depict SA in Figure 18. Endsley's model of SA places emphasis on cognitive aspects and accentuates the role of mental models and goals in the creation and sustenance of SA (Figure 14).

Level 1 SA arises from environmental facets that help understand the spatial and temporal state of the manufacturing system as viewed from the digital twin. The digital twin in itself deliver the potential to offer deeper insights on the underlying processes. For example, by just inspecting the digital twin it will be possible to perceive a range of scenarios such as the current location of the pallets, status of the machining centers, the current capacity of the storage and so on. Temporal (time-related) facets are perceived via dashboards [73] that display key performance indicators of the manufacturing system among others. This may include the cycle time for various product parts and machine throughput and lead times. The spatial and temporal aspects are crucial in achieving SA and is what is intended in Endsley's definition mentioned earlier as "volume of space and time". Temporal aspects play an important aspect in all levels of SA. Time required to perceive information would be as important as time required to comprehend its meaning. Further, time required to project future states of the manufacturing system would important as well depending on the context. For example, if a machine tool is nearing end of life, or a tool has been in fact damaged, it would be vital to observe this information via a warning message from the digital twin (level 1), comprehend it and realize aspects such as which machining center, what tool, etc. (level 2), and then project future states as to which orders would not be ready on time, increased workload on remaining machining centers and how that would in turn affect other work orders and so forth (level 3). Often is the case with digital twins that the level of automation is high that future states are predicted and are produced for direction perception but we do not consider that here for the sake of this example.

Comprehension of these facets to develop an understanding of the current state constitutes Level 2 SA. This is debatably the most important level and where most of the learning as a consequence of SA takes place. In literature, the term mental model is used to connote organization of a person's knowledge and has been used synonymously with other terms such as knowledge structures and cognitive maps. Integration of environmental cues (obtained via perception) for comprehension may also involve updating priorly existing mental models (accommodation) or creation of new mental models (assimilation) if pattern matching fails and unfolding events cannot be matched to existing mental models. Under the circumstance, having sufficient situational awareness can help facilitate this activity by being able to relate the event with a probable cause. Causality is an important factor involved in learning from situational awareness and helps attach schemas of prototypical system states to mental models. Learning takes place when people develop advanced mental model in a specific domain as a result of the learner's experience in the domain [78]. Thus, learning that stems from SA can be attributed to the cognitivism theory of learning where emphasis is placed on how information is obtained, structured, stored and recovered from the mind. Projecting future states of the system is what level 3 SA is comprised of and unlike from a SA standpoint, it is of lesser importance in the learning perspective. However, by virtue of "learning" from a combination of level 1 and 2 SA, the learner is able to foresee the future states of the system. Figure 19 provides a conceptual architecture to encompass the discussed learning theories supporting the pedagogic digital twin framework.

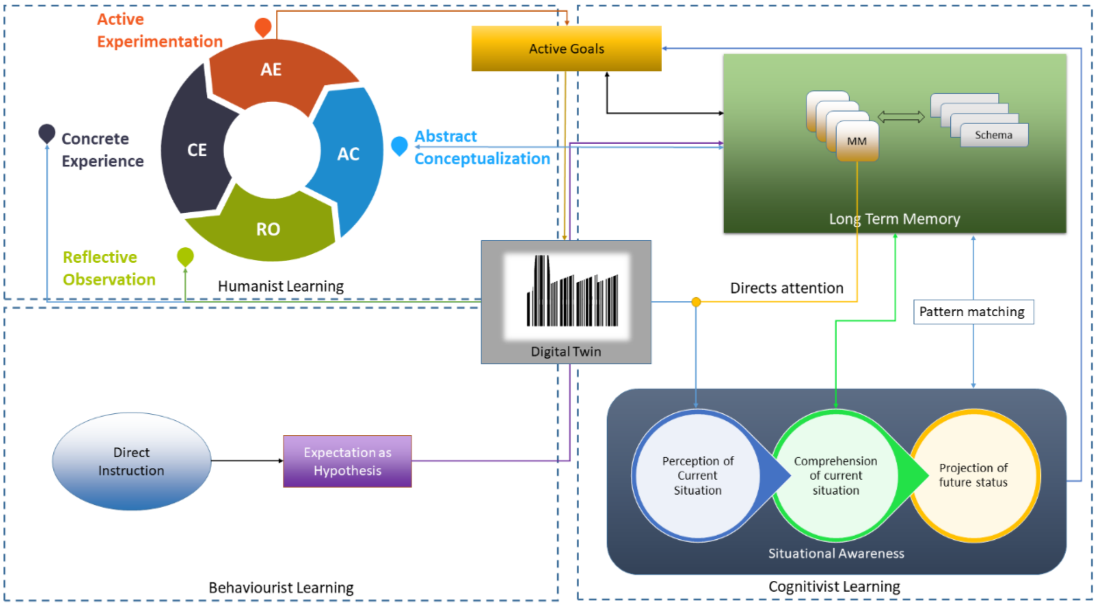


***Figure 19.*** *Instantiation of Learning Theories in the didactic framework* [79]

The main motive behind the architecture is to highlight how learning theories can be instantiated within the framework and how they co-exist with analogous relations. The direct instruction provides the initial hypothesis to the learner. The generated hypothesis provides the basis on which the learner is expected to have a concrete experience and reflectively observe unfolding processes. The hypothesis from the cognitive learning standpoint is seen as an expectation that is responsible for the formation of early mental models and directs attention thereby influencing perception (Level 1 SA). This is followed by comprehension (Level 2 SA) that uses prior mental models to form and existing environmental cues to form an understanding of the situation. This is analogous to the abstract conceptualization of the humanist learning where either assimilation or accommodation takes place. Projection of future states of the system along with operator goals help the learner to take goal-oriented decisions (omitted from the figure) on the digital twin. Pursuit of goals also requires activation of appropriate mental models and associated schema and/or scripts. This is analogous to the active experimentation of the ELT where the learner applies learnt concepts into goal-oriented actions to solve problems make decisions and influence events. Thus although the learning theories could seem mutually exclusive from an individual theory perspective, we see how multiple theories can be instantiated within the pedagogic digital twin framework supported via analogies. Thus the pedagogic framework is not in support of a particular theory, and while one theory may be better suited to explain the didactic learning transformations we do not rule out the others completely. Rather, the argument here is that the digital twin being able to support existence of multiple theories only justifies its proposition as an educating tool.

# 4. RESEARCH METHODOLOGY

This chapter aims to contribute with the methodology used in the development of the digital twin in the thesis (Figure 1).This chapter completely addresses second research question identified in chapter 1 of the thesis. The subsections guide the reader into systematic iterative steps adopted by the author in studying the research problem along with creation of the digital twin.

## 4.1 Envisage and Envision

This stage involved researching the aptness of Digital Twin in the educational set up by examining the value it brings to the students. As part of the research done, two publications emerged.

In the first Publication titled "*Leveraging Digital Twins for Assisted Learning of Flexible Manufacturing Systems*" an approach in using Digital Twin as a medium for educating students on (flexible) manufacturing systems is presented based on a pedagogically sound framework, i.e. Kolb's experiential learning cycle. The transformation of a general hypothesis to competency development during different stages of the learning methodology is adjacently compared with Kolb's experiential learning to justify the proposition.

The second Publication Titled "Learning Experiences Involving Digital Twins" presented widely researched learning (behavioural, cognitive and humanistic) theories were presented before using them to draw a parallel to a learning framework developed in the earlier publication. In doing so, an attempt was made for warranting its use as a tool to educate students.

At this stage the learning outcome of the existing course were identified as "Ymmärtää tuotannollisen toiminnan perusteet, tärkeimmät valmistustekniikat ja toimintamallit suomalaisessa teollisuudessa" or as "Understand the basics of production activity, key manufacturing techniques and operating models in Finnish industry" in English.

This stage, while laying the foundations of the research also gave the green light to pursue solutions to the posed research questions.

## 4.2 Identification of the Process and Course Activities

This stage involved finding out the right process and layout that would help in accomplishing the course objectives. The manufacturing process chosen was that of a flexible manufacturing system and understanding its principles were the objectives. The configuration layout was ascertained after multiple discussions and negotiations with course personnel and the supplier of the manufacturing system until a configuration layout was mutually agreed upon.

Some of the factors taken into account (besides the ones that are typical of FM Systems) during identification of the configuration were as follows:

- The number of loading stations should be enough to accommodate at least five groups at a time taking into account the number of course participants.
- Distributing the loading and material stations to prevent cramping as it would be unsafe to as people and forklifts would be in the same place. Although a virtual system, safety standards were not to be violated.
- The processes and thereby the machines that are incorporated should be broad enough for students to comprehend those important practices prevalent in the industry.

The Layout drafted in CAD at this stage is shown in Figure 20

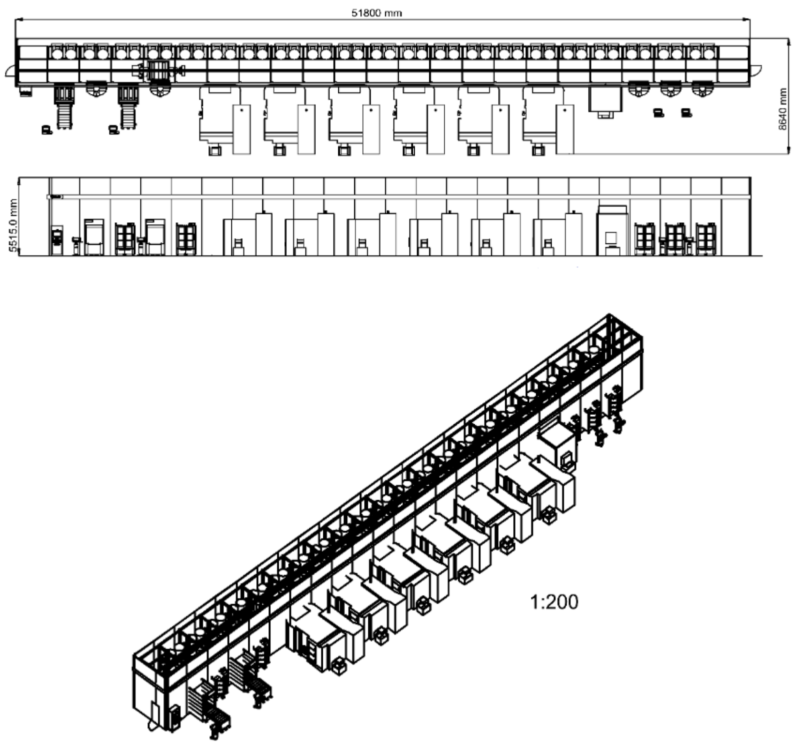


***Figure 20.*** *Identification of the Process: CAD Layout*

The layout consisted of an Automated Storage and Retrieval System that included a crane and a storage that could house 126 pallets (84 machine and 42 material pallets), five Loading Stations, six Machining Centers and two Material Stations.

The course was set to follow the didactic framework developed in the Chapter 3, Section 3.1. The lectures are planned to introduce the course to the student while explaining theoretical concepts underpinning flexible manufacturing systems. The digital twin is introduced here. This is to be followed by an online quiz where the students would have to answer questions such as “Where can I find information about the remaining life on this tool?” The goal of this exercise is to familiarize the students with the user interface and the ideology behind it.

The next part of the didactic framework takes place in the laboratory where in the laboratory where the students were to focus mostly on the non-physical side of the system and create a fixture, a part, install the fixtures on a pallet and then make an order. The required NC-programs and materials will already be present and the students are allowed to use them. This was done so as to save time considering the number of students taking the course.

This was to be followed by an exercise at the Training Center. In the second exercise, the students will now manufacture and complete the order they made in the first exercise. The main focus here is the physical aspect of the system such as installing the fixtures, adding material to the fixture and seeing the material getting machined and moved in the system and actually using the loading stations.

## 4.3 Pilot a Twin

A pilot twin is next developed consisting of a component of each type (Figure 21). During the development of the twin, several design principles (covered in Section 5.1) were followed. Of this, modularity is of prime significance and is worth a mention here. The pilot twin was realized with each component having logic necessary only for its own functioning. In other words, there is no central middle-ware that takes care of the functioning of other components. This was part of a vision of the digital twin to be able to adapt to various layout configurations as part of future work to the work done in the thesis.

Known performance related issues with prior implementation necessitated to be vigilant in this regard. Several steps were taken in this regard.

1. The number of simultaneously running python scripts were kept to a minimum in each component. The same is the case with other behaviors and properties
2. Features (covered in Section 5.1.2) of components were “collapsed” and “merged” so as to have lesser geometries to be tracked in real-time while the simulation is running,
3. Raycast sensors used to detect the presence of pallets on conveyors were activated only if there was an incoming pallet to the station rather than to have it run throughout.

4. Services were selected carefully only to serve their required purpose. For example, services that enabled to obtain the location of pallets in particular positions were invoked by components checking if there were pallets in only their respective containers rather that invoking services that provided location details of all pallets.

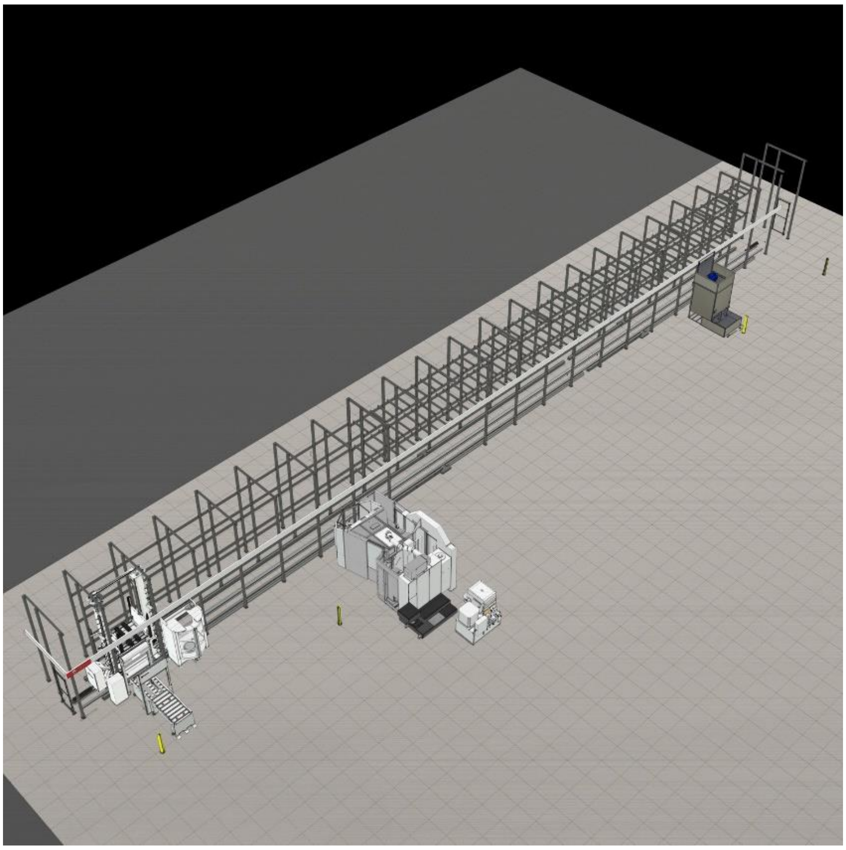

***Figure 21.*** *Pilot a Twin: Pilot Twin*

## 4.4 Deploy Pilot Twin

The Pilot twin is deployed and what followed the deployment is a number optimizations in performance parameters tuning to get it to the desired state. This deployment serves mainly two purposes:

1. Checking how the deployment has supported the goals identified as a part of the envision and modelling stages is analyzed.
2. The deployment of the pilot twin also serves as a midway point in implementing the functional logic in the creation of the digital twin.

Since the next step would be to scale twin to encompass the complete layout any anomalies present would be duplicated and it would be appropriate to rectify any present at this stage. Commenting source codes using software design styles and conventions mentioned in Section 5.1 was also done in this stage so as to avoid re-commenting similar codes once scaling the twin to the complete layout.

## 4.5 Model Domain Knowledge and Pedagogic Extensions

As mentioned in the earlier section, any digital twin of pedagogic value needs to have didactic transformations that justifies its pedagogical context of usage. Such transformations, as mentioned in the research questions in the previous section, needs to have a description of the learning objectives. Further, any pedagogic tool would be incomplete without assessing and evaluation of the learning outcomes.

A discussion between ontologies and databases was presented earlier in this paper. However, the author has chosen to go ahead with ontologies for the following reasons:

- Taxonomic reasoning was fundamental in deciding with ontologies due to their ability for semantic modelling of concepts. With the ability to use classes, properties, instances, aggregation and generalization relations, ontologies were deemed more suitable as opposed to databases that focus on data storage and prove challenging to represent manufacturing domain knowledge.
- The Open World Assumption (OWA) by ontologies would prove useful in times when modelling the knowledge acquired by the students
- Ontology Query Languages such as SPARQL or SERQL uses rich vocabulary of the ontology via predicates.
- Future developments on works presented in this paper can be made with the ontology developed in this paper. Using databases, it would be relatively difficult to build on existing databases and a new one would have to be built from the ground up.

## 4.6 Scale the Twin

The digital twin is scaled to encompass the entire layout (Figure 22). The modular approach to developing the pilot twin meant that this stage involved only duplicating existing components as the pilot twin constituted of a component of each type in the complete layout.

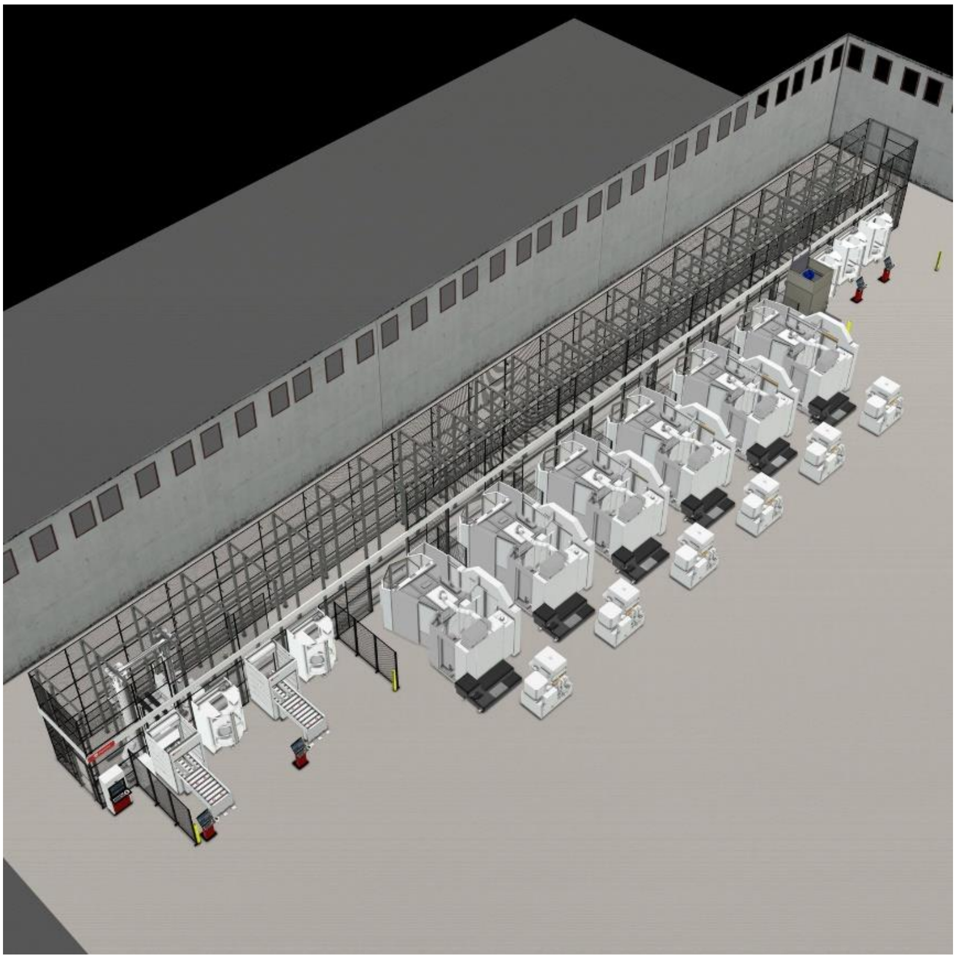

***Figure 22.*** *Scale the Layout: Completed Layout*

## 4.7 Monitor, Measure and Publish results

The completed model is deployed and the solution is examined to measure the learning that occurs through the digital twin. More on this is covered in Section 6.

A third publication titled "Attaining Learning Objectives by Ontological Reasoning using Digital Twins" that examines the outcomes of this research is published. This publication mainly focusses on the automated validation of the learning outcomes of the operator that uses the digital twin.

# 5. IMPLEMENTATION

This chapter presents the concrete work done in regard with development of the digital twin in the thesis. Section 5.1 presents the software design considerations and conventions while section 5.2 presents the component specific modelling of the physical system in the digital twin environment.

## 5.1 System Architecture

**The manufacturing system**

The Manufacturing Control System consists of hardware and software components. The hardware consists of a controller that functions as the brain of the system an interface between the Digital Twin and the physical system. The control cabinet is often integrated with a control station. The Manufacturing Control System software components include:

- A control system simulator representing the physical system that exchanges supervisory statuses and control commands with the controller.
- A web based interface known as the Data Manager that is used to manage master data relating to part routing and manufacturing, make and visualize orders and schedules and view production logs.
- A web based Dashboard that shows real-time 3D view of the system with live status updates about devices and operational conditions.

There exists two configurations in which this system exists; virtual and physical. For the purpose of classroom or distance learning, there exists a virtual configuration devoid of the simulator and the controller, where the Digital Twin can be operated without the physical FMS environment. Although the sense of a Digital Twin is arguably lost here by definition due to the absence of it physical counterpart, it models the controller precisely as the manufacturing control system hardware. The high maturity in modelling the control system means that for all purposes of discussion the sense of a Digital Twin is still maintained. This virtual configuration is being leveraged to train personnel and educate pupils of the concept of the FM System before actually working on the physical system. In this virtual configuration the Control Station interface from the physical system, i.e. the Dashboard and the Data Manager, is made available on via a web browser and it is this configuration that we make use of in this thesis

**Implementation Architecture**

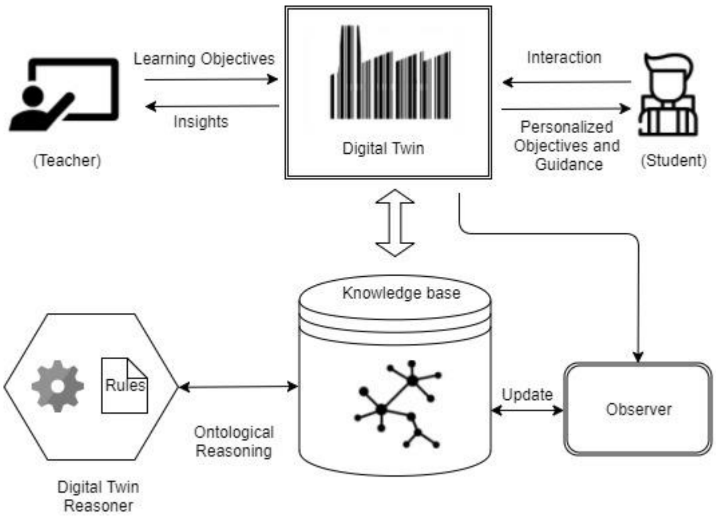


***Figure 23.*** *Implementation System Architecture*

The system architecture (depicted in Figure 23) consists of the Digital Twin of a manufacturing system that interacts with its users (both students and teachers), a knowledge base that consists of the domain ontology. A reasoner engine assesses the learning outcome of the student against the set learning objectives and constantly evaluates the progress of the student towards the set objectives with prior defined rules. The Observer (not implemented in the use case) is the module responsible for capturing additional information that pertains to the user behaviour and thereafter updates the learner's profile. This could be anything from mouse tracking, eye-tracking, time spent navigating the digital twin and so on.

## 5.2 Ontology Modelling

Ontologies have long been used for representing information. We use ontological models of the domain knowledge and other knowledge facets in the digital twin environment. The ontology developed comprises of three main artefacts as a sub class of the general class Thing; the manufacturing system, the pedagogical elements and the learning that occurs as a result of pedagogy.

**Manufacturing System Ontology**

Representation of the manufacturing system is created using ontologies as shown in Figure 24. As can be observed, the manufacturing management system is not modelled in its entirety with the aim to keep things simple. Rather, only those domain information relevant to the context of the learning use case has been modelled in detail while remaining facets have been modelled without any detail at all.

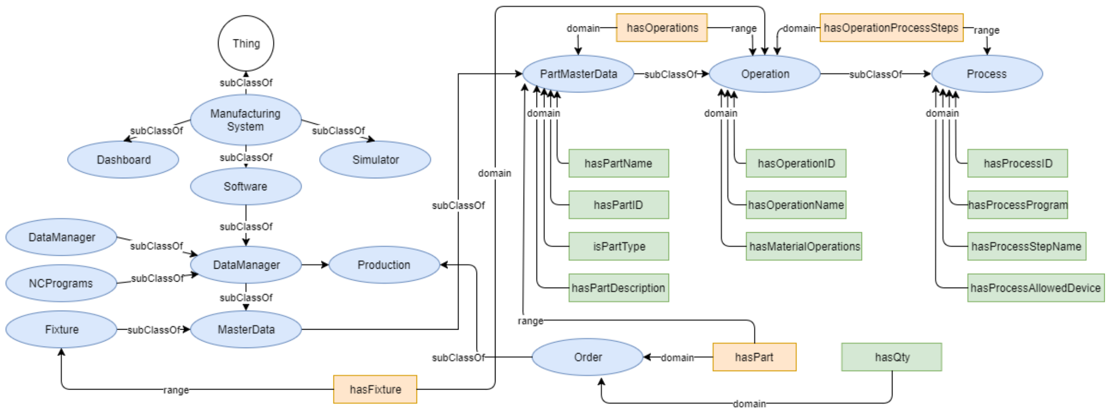


***Figure 24.*** *Manufacturing System Ontology*

Of importance in the domain information is the Manufacturing Control System Data Manager. It consists, *inter alia,* the Master Data and Production details. The Production Class mainly consists of the orders, that has *hasOrderID, hasQty, hasOrderDescription, hasOrderStatus* as data properties and *hasPart* as an object property that has Class Part_Master_Data as its range.

The part master data consists of the Part Master Data and the Fixture. The parts have data properties *hasPartDescription*, *isPartType*, *hasPartName*, *hasPartID* while has an object property *hasOperations* of range Operation. Operation itself consists of data properties *hasFixture*, *hasOperationID*, *hasOperationName* and object property *hasOperationProcessSteps* of range Process. Process further has data properties *hasProcessStepName*, *hasProcessProgram*, *hasProcessID* and *hasProcessAllowedDevices*.

The structure underlying the ontology mentioned above is that the Order class has one or more orders (instances of class Order) that contain a part. Each part is manufactured by one or more operations (instances of class Operation). These operations further are materialized by a sequence of process steps (instances of class Process) that include a combination of loading, machining, washing, manual and unloading steps.

### The Learning Ontology

The learning ontology (super class Learning) comprises of the learning outcomes, learning objectives and the learning taxonomy (Figure 25). The theoretical aspects underlying these terminologies have been covered in Section 2.5.2 and Section 2.5.3.

The *LearningObjectives* class represents the ABCD model (Section 2.5.2). Hence, it comprises of properties *hasAudience*, *hasBehaviour*, *hasCondition* and *hasDegree.*

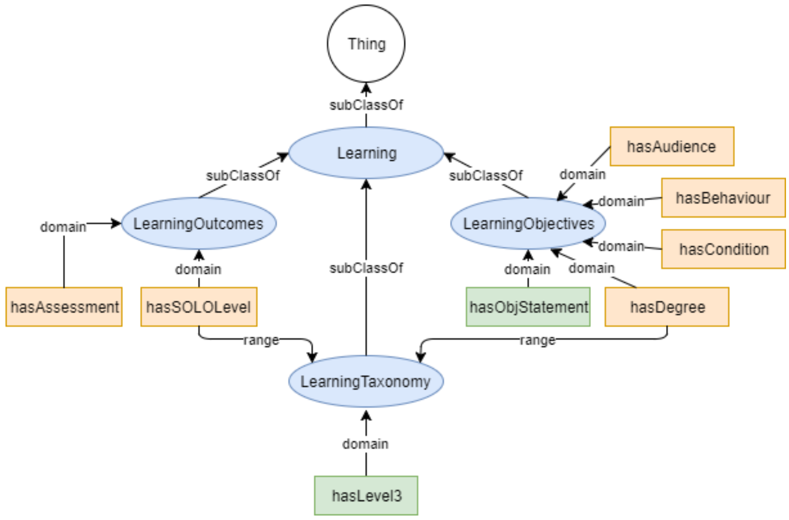


***Figure 25.*** *Learning Ontology*

The *LearningTaxonomy* Class represents the SOLO Taxonomy and has five properties that are its characteristic (covered in Section 2.5.3). These are *hasLevel1, hasLevel2, hasLevel3, hasLevel4, hasLevel5*.

The *LearningOutcomes* Class represents the outcome of learning and is modelled to have two properties, *hasAssessment* and *hasSOLOLevel*.

**Pedagogy Ontology**

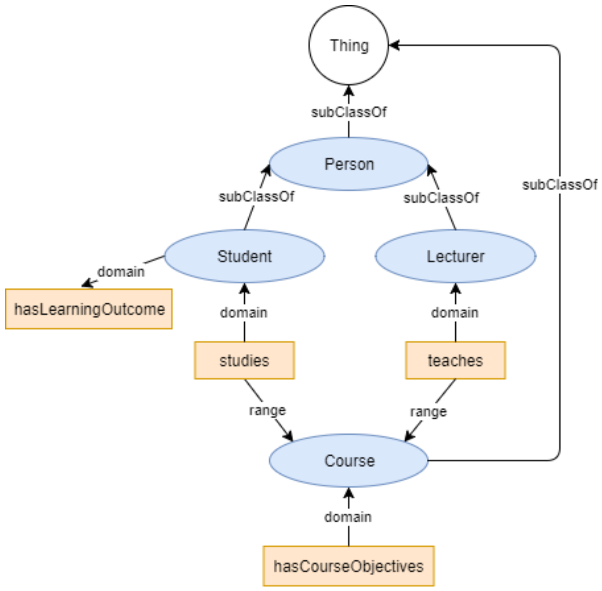


***Figure 26.*** *Pedagogy Ontology*

The pedagogy ontology (Figure 26) consists of Class Person and Class Course. The Person Class constitutes of sub-Class *Teacher* that *teaches* (object property) a Course and a sub-Class *Student* that *studies* (object property) the *Course*.

Further discussion of the ontology takes place in Section 6.1.3.

**Domain + Manufacturing + Pedagogy: The Combined Ontology and Course Alignment**

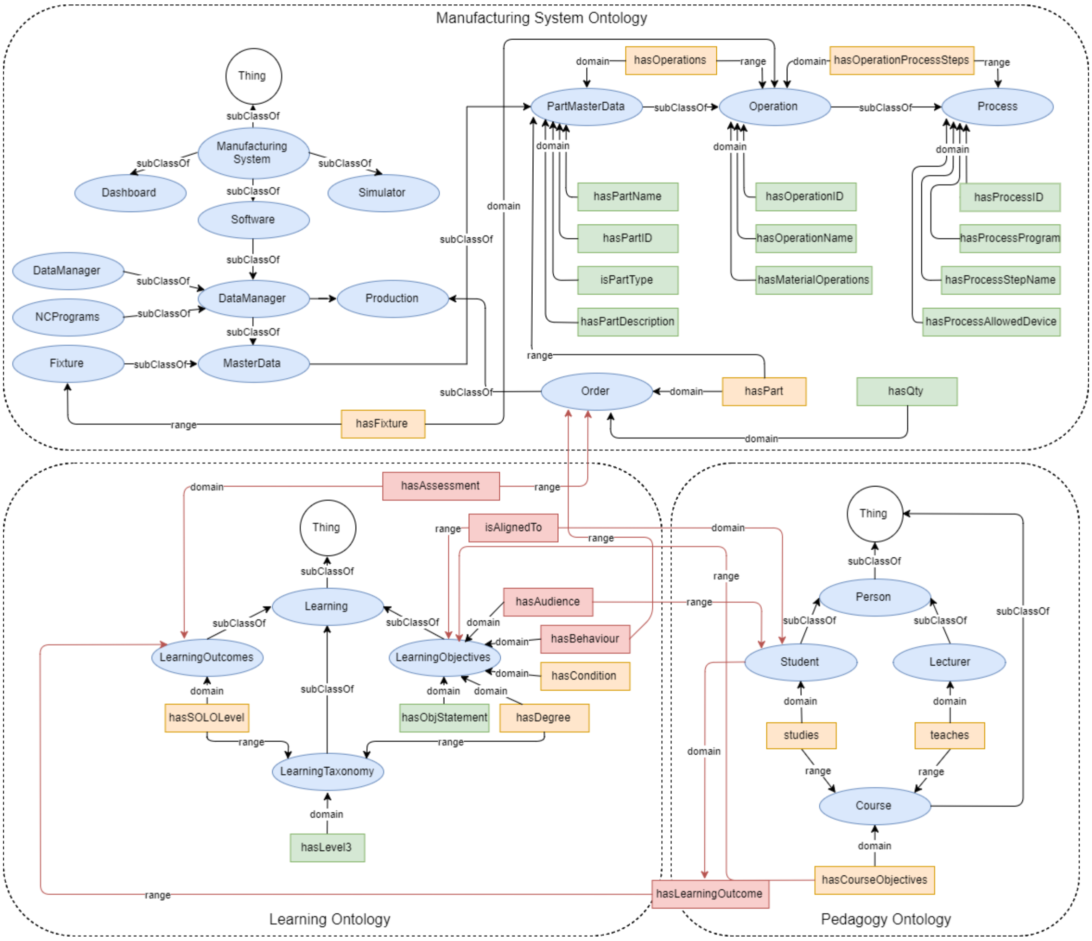


*Figure 27.* *Combined Ontology*

The combined ontology (Figure 27) is a merger of the above three ontologies representing the manufacturing system, pedagogy and the learning that occurs within the domain. The main objective of this merger is the alignment of the course, and by doing so making sure that the course objectives (and thereby the course goals) are in harmony with the learning activities and the assessment.

This merger makes way for a few associations. These associations are marked in red. A student that takes the course *hasLearningOutcome* (object property) of Class *LearningOutcome*. The course *hasCourseObjectives* (object property) of Class *LearningObjectives* which inturn *hasAudience* as the student. The *LearningObjectives* Class also has an object property *hasBehaviour that* for the current use case is modelled as Class *Order*. Class *Order* is also modelled as the learning outcome that has *hasAssessment* which in the context of this use case is the *Order* Class of the Manufacturing Management System. The significance of this modelling is discussed in Section 6.1.3.

## 5.3 Software Design Considerations and Conventions

**Software used:** Visual Components

**Software version:** Premium 4.1

**Programming Language:** Stackless Python v2.7.1

### 5.3.1 Software Design Principles:

1. **Modular Programming - Separation of Concerns**

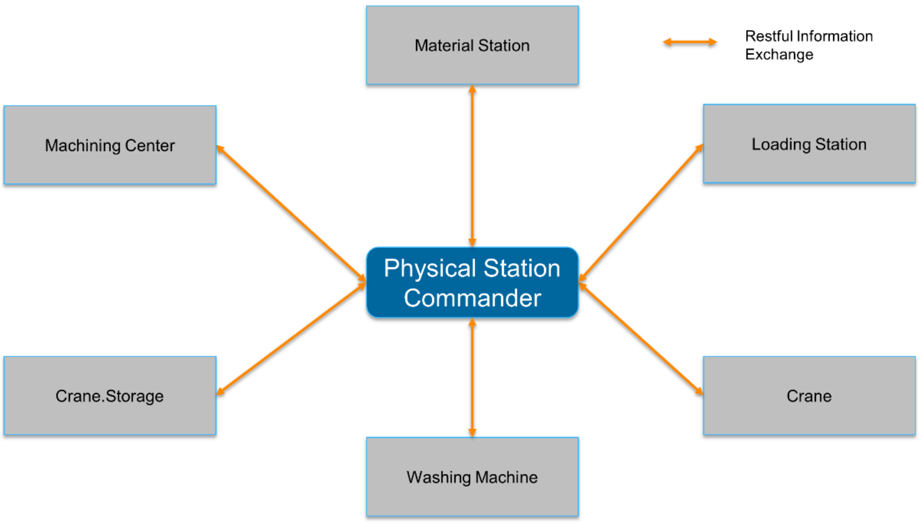


***Figure 28.*** *Modular Programming: RESTful Exchange between VC component*

Each instance of a component is independent in terms of its functionality and does not depend on any other component. Each component is responsible for invoking services relevant to it. (Figure 28) For example when the program resumes in between a running production order, each component is responsible for checking of any existing pallets within its containers. In other words, there is no central device, broker or component that assigns tasks to other components.

2. **Omission of 3rd Party Modules**
The programming is done using libraries that are built-in Python so that they work out of the box on distributed systems.

3. **Don't Repeat Yourself**
Functions are used to avoid repetitive code to make it easy to maintain and keeping it concise

4. **Keep it Simple, Silly**
   Logics have been formulated to be simple with complexity just enough to obtain desired results

   However, at occasions where there was a tradeoff between efficiency and clarity, efficiency was pursued due to the nature of the resource-hungry digital twins.

### 5.3.2 Introduction to Visual Component Terminologies

The sections aims to introduce the reader to certain Visual Components terminologies that are used in the thesis (Figure 29). What is covered in this section is by no means a comprehensive explanation of all the terminologies but only a subtle introduction to what is necessary in understanding the subsequent sections.

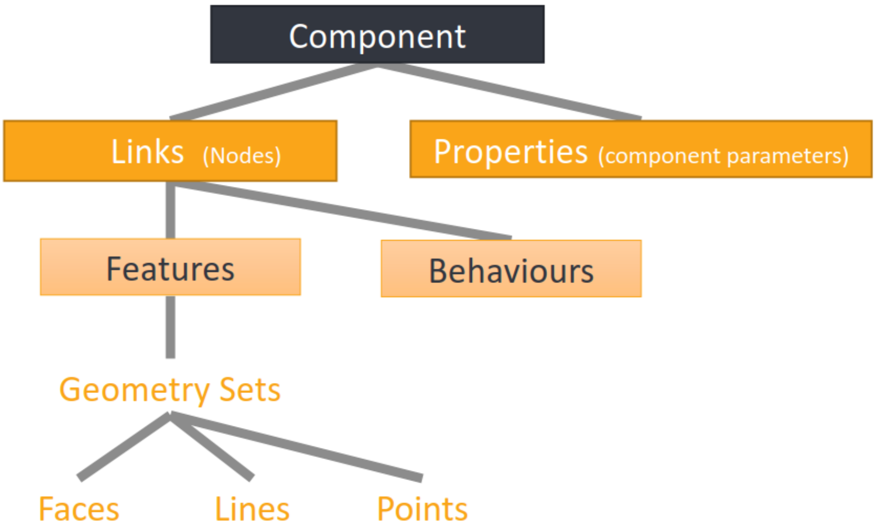


***Figure 29.*** *Visual Components Terminologies*

Each machine or object in 3D world in Visual Components is referred to a component and will be referred to as a VC component henceforth. Each component is made of links (nodes) that form the kinematic chain of the component and properties that define the component such as length width and height. Each link has its own features and behaviour. Features are the 3D CAD features and can include faces, lines or points, collectively known as geometry sets. Behaviors define the behaviour of the component during the simulation and there are many ways this can be done, the most popular of which are python scripts. Python Scripts are scripts that let us code the behaviour of components using the python API of Visual Components.

### 5.3.3 VC Component Script Behaviors

All VC components contain links and properties as described in the previous section. However, a certain methodology has been followed when naming script behaviors which will be introduced in this sub-section.

Figure 30 shows the basic architecture of scripts contained in components.

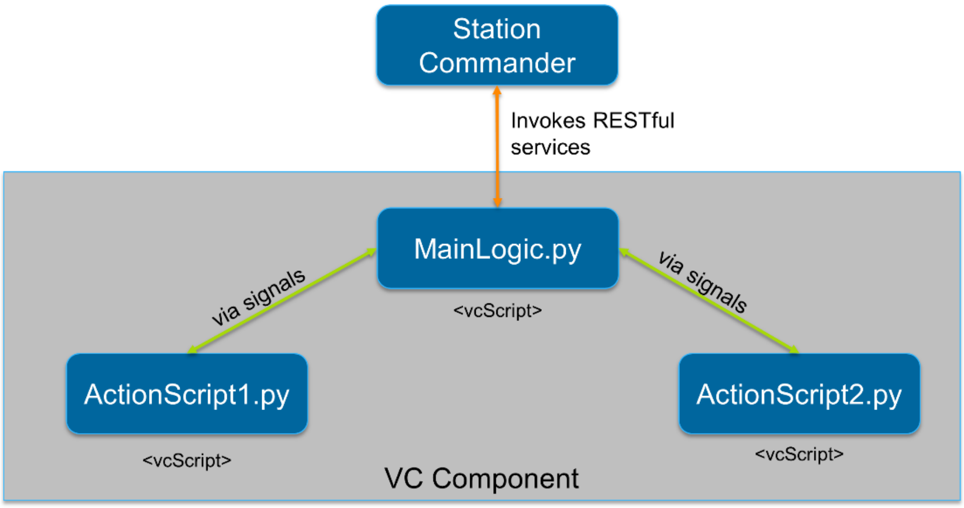


***Figure 30.*** *Visual Components Scrips in Components*

**This thesis uses python script behaviours with two distinct names, "MainLogic" and "ActionScript" and their difference is as follows:**

1. Scripts named "ActionScript" contain source code to cause animations such as opening of doors, changing of changer table, etc. They are named following the Upper Camel Case convention of behaviors mentioned in the next section. When more than one Action script exist in a component a number is appended, for e.g. ActionScript1.

2. Scripts named "MainLogic" contain source code that contain the complex scripts responsible for functioning of the components, including invoking of services and handling of the returned information. They too follow the Upper Camel Case convention of behaviors mentioned in the next section and are named as "PythonScript"

Another behaviour used in the software development is that of motion paths. A motion path allows you to contain and move components forward and/or backwards along a path defined by Frame features.

### 5.3.4 Software Design Styles and Documentation

This section introduces the reader to the programming style maintained throughout the software development of this thesis. Design considerations pertaining to VC Components and those pertaining to python script behaviours are documented in separate subsections below.

**Naming Conventions of VC Components (Including their Properties and Behaviors)**

**Component Names**

Naming of the Components were governed by the corresponding names in the Simulator. Six Major Components exist in the Digital Twin

- Crane (Includes Storage)
- Material Stations with the names MX where X = {1,2}
- Loading Stations with the names LX where X = {1,2,3,4,5}
- Machine Centers with the names MCX where X = {1,2,3,4,5,6}
- Washing Machine with the name WM1
- Control Stations with the names SCX where X = {1,2,3,4} with the main Control Station denoted by Just SC

**Properties, Behaviors and Links**

Properties, Behaviors and Links of VC Components are denoted by Upper Camel Case (also known as Pascal Case) where the first letter of every word is capitalized. For e.g.: DoorHeight, PythonScript, OperatorDoorOperate, etc.

**Containers and Paths**

Naming of the Containers were governed by the corresponding names in the Simulator. Table 3 shows the names of the existing containers along with their components.

***Table 3.*** *Container and Path Naming Conventions*

| Component | Component Notation | Container(s) | |
|---|---|---|---|
| | | | |
| Crane | Crane | Crane.Storage.X | X = {1,2,3..84}<br>X= {1000,1001…1041} |
| Crane | Crane | Crane.Forks | |
| Material Station | MX | MX.User | X = {1,2} |
| Loading Station | LX | LX.User | X = {1,2,3,4,5} |
| Machine Center | MCX | MCX.Changer<br>MCX.Table | X = {1,2,3,4,5,6} |
| Washing Machine | WM1 | WM1.Table | |
| Control Station | Control Station | - | |

## Documentation and Style Conventions of Python Scripts Behaviors in VC Components

### External Documentation or Program Header:

1. Every Python Script behaviour includes at the beginning, a comment block with the following information:

- **Author Name and Details** that display the name and other details of the programmer
- **Date** when the program was written
- **Language of the program** including the version number
- **Component Name** which is the VC component in which the script is contained
- **Description** of what the script does.
- **Comments** if any

### Internal Documentation

1. A Block comment is placed in the beginning of every method ( function or subroutine). This includes:

- Type and Description of the function; a function can be of two types user-defined or built-in
- Arguments received and returned parameters
- Global variables used by the function

2. Variables or Identifier names are selected to be self-explanatory that communicate its purpose to the reader. Variable names that are not self-explanatory has a comment line along with its declaration that explains its usage.

3. Any complex sections of code has explanations ahead of them.

### Programming Style:

1. Variable names and identifiers are denoted by lower came case where the first letter of the entire word is lowercase, but subsequent first letters are uppercase. E.g.: palletNumber, encodedData, etc.
2. User defined methods are in lower camel case as well.
3. Methods and Properties from Visual Components Python API are of upper camel case.
4. All letters of constants are capitalized. IP = 192.168.9.10, PORT = 8760, etc.

### Miscellaneous Styles

1. There is only one statement per line.
2. White spaces are used before and after to set off logically related sections of code.

## 5.4 Digital Twin Component Design in Visual Components

### 5.4.1 Material Station

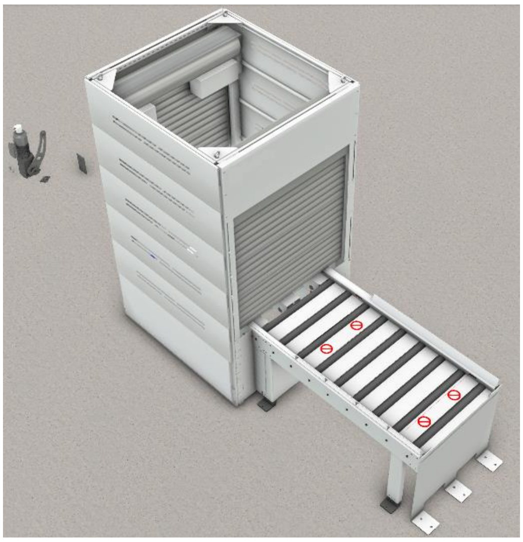

***Figure 31.*** *Material Station model*

***Table 4.*** *VC Properties of Material Station*

| Name | Type | Description |
|---|---|---|
| AisleDoor | <vcProperty> of type Button | A Button property than toggles between opening and closing of the aisle door |
| OperateDoor | <vcProperty> of type Button | A button property that toggles be-tween opening and closing of the operator |
| RefNode[7] | <vcProperty> of type <Ref<Component>> | RefNode is a property that allows the Material Station to be "linked " to a Loading Station. |

7 Loading Station typically requires materials that comes from material pallets from material stations. RefNode is a property of the Material Station that should be used to reference a Loading Station to which it acts as the Material Station, typically the closest Loading Station.

***Table 5.*** *VC Behvaiours of Material Station*

| Name | VC Type | Description |
| --- | --- | --- |
| MX.User | <vcMotionPath> | A 2-way path that allows to contain and move pallets forward and backwards along a path defined by Frame Features. It serves as the conveyor in the Material Station. |
| OperatorDoorServo | <vcServoController > | A controller used for driving the joints of the operator door. |
| AisleDoorServo | <vcServoController > | A controller used for driving the joints of the Aisle door. |
| OperatorRaycastSensor | <vcBehaviour> of type <rRayCastSensor> | A sensor located at the operator end of the conveyor to stop the conveyor. |
| AisleRaycastSensor | <vcBehaviour> of type <rRayCastSen-sor> | A sensor located at the aisle end of the conveyor to stop the conveyor. |
| ReceiveComponent | <vcComponentSignal> | A component signal used by the crane to signal that a component (value of the signal) has been placed in the container of the material station. |
| OperatorDoorOperate | <vcBoolSignal> | A Boolean signal that causes the Operator Door to open (TRUE) or close (FALSE). |
| OperatorDoorDone | <vcBoolSignal> | A Boolean signal that pulses TRUE when an operation, either open or close is completed. |
| AisleDoorOperate | <vcBoolSignal> | A Boolean signal that causes the Aisle Door to open (TRUE) or close (FALSE). |

| | | |
|---|---|---|
| AisleDoorStatus | <vcBoolSignal> | A Boolean signal that has the status of the last completed operation of the aisle door. (TRUE = open, FALSE = closed) |
| AisleDoorDone | <vcBoolSignal> | A Boolean signal that pulses TRUE when an operation, either open or close is completed. |
| AisleRaycastSense | <vcBoolSignal> | A Boolean signal that is used by the Aisle Raycast sensor to signal the detection of a component. This is used to stop the conveyor |
| AisleRaycastPulse | <vcBoolSignal> | A Boolean signal that drives the Aisle Raycast sensor. This is necessary for performance reasons as the Aisle Raycast sensor is made active only when there is a pallet. |
| OperatorRaycastSense | <vcBoolSignal> | A Boolean signal that indicates the detection of a component by the Operator Raycast Sensor |
| ActionScript1 | <vcScript> | A python script behaviour that andles the functioning of the Aisle Door of the Material Station and also defines the Aisle Door button for the same |
| ActionScript2 | <vcScript> | A python script behaviour that handles the functioning of the Operator Door of the Material Station and also defines the Operator Door button for the same |
| MainLogic | <vcScript> | A python script behaviour that handles the functioning for the functioning of the Material Station like signaling appropriate |

| | | |
|---|---|---|
| | | signals, initialization of state, invoking services and so on. |

**Functioning Logic**

The material station is where materials required to be loaded on machining pallet is brought on material pallets. The material station has two doors, one on the operator side and the other on this aisle side. The aisleDoorOperate signal is triggered by the crane when it is assigned a task to either pick from or place to the material station a pallet. The material station has two ray cast sensors that detect when the material pallet has reached either ends of the conveyor.

Once the pallet is placed and the crane has retracted its forks, the aisleDoorOperate is signaled a False value by the crane to close the Aisle door. Once the pallet is inside and the aisle door has closed, the operator door opens. Once the operator door has opened completely, the material pallet is carried forward on the conveyor towards the operator. Once the material pallet reached the operator end, it triggers the Operator Raycast Sensor that causes it to stop the conveyor.

Once the machining pallet is "loaded" in the corresponding Loading Station (value of RefNode Property, see in Footnote 7), the material station sends the material pallet backwards towards the aisle. The Operator Door closes and the Aisle Door opens once the material pallet has reached the Aisle Side.

## 5.4.2 Loading Station

***Figure 32.*** *Loading Station model*

*Table 6. VC Properties of Loading Station*

| Name | Type | Description |
|---|---|---|
| AisleDoor | <vcProperty> of type Button | A Button property than toggles between opening and closing of the aisle door |
| OperateDoor | <vcProperty> of type Button | A button property that toggles between opening and closing of the operator |

*Table 7. VC Behaviours of Loading Station*

| Name | VC Type | Description |
|---|---|---|
| LX.User | <vcComponentContainer> | A Component Container is the location at which a pallet is temporarily stored. |
| OperatorDoorServo | <vcServoController > | A controller used for driving the joints of the operator door. |
| AisleDoorServo | <vcServoController > | A controller used for driving the joints of the Aisle door. |
| ReceiveComponent | <vcComponentSignal> | A component signal used by the crane to signal that a component (value of the signal) has been placed in the container of the material station. |
| OperatorDoorOperate | <vcBoolSignal> | A Boolean signal that causes the Operator Door to open (TRUE) or close (FALSE). |
| OperatorDoorDone | <vcBoolSignal> | A Boolean signal that pulses TRUE when an operation, either open or close is completed. |

| | | |
|---|---|---|
| AisleDoorOperate | <vcBoolSignal> | A Boolean signal that causes the Aisle Door to open (TRUE) or close (FALSE). |
| AisleDoorStatus | <vcBoolSignal> | A Boolean signal that has the status of the last completed operation of the aisle door. (TRUE = open, FALSE = closed) |
| AisleDoorDone | <vcBoolSignal> | A Boolean signal that pulses TRUE when an operation, either open or close is completed. |
| ActionScript1 | <vcScript> | A python script behaviour that handles the functioning of the Aisle Door of the Loading Station and also defines the Aisle Door button for the same |
| ActionScript2 | <vcScript> | A python script behaviour that handles the functioning of the Operator Door of the Loading Station and also defines the Operator Door button for the same |
| MainLogic | <vcScript> | A python script behaviour that handles the functioning for the functioning of the Loading Station like signaling appropriate signals, initialization of state, invoking services and so on. |

**Functioning Logic**

The Loading Station is used to load machining pallets. It has two doors, the Aisle Door, just like the Material Station and the operator door, that unlike that of the material station, swivels on hinges. The aisleDoorOperate signal is triggered by the crane when it is assigned a task to either pick from or place to the loading station a pallet. This causes the Aisle Door to open.

Once the pallet is placed and the crane has retracted its forks, the aisleDoorOperate is signaled a False value by the crane to close the Aisle door. Once the pallet is inside and the aisle door has closed, the operator door opens.

After the machining pallet has been loaded, the operator door is closed first, followed by the opening of the aisle door.

### 5.4.3 Machining Centre

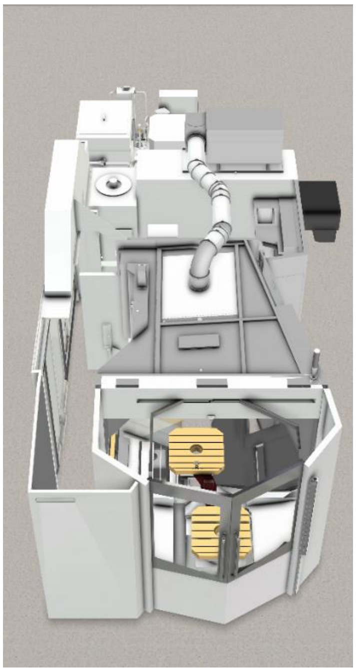

***Figure 33.*** *Machining Center model*

***Table 8.*** *VC Property of Machining Center*

| **Name** | **Type** | **Description** |
|---|---|---|
| Status | <STR> | Current Status of the Machining Centre, either, “ASideOut” or “BSideOut” * |

***Table 9.*** *VC Behaviours of Machining Center*

| **Name** | **Type** | **Description** |
|---|---|---|
| APCServo | <vcServoController > | A Servo controller used for driving the joints of APC. |

| | | |
|---|---|---|
| DoorServo | <vcServoController > | A Servo controller used for driving the joints of AisleDoor. |
| AisleDoorOperate | <vcBoolSignal> | A Boolean signal that causes the Aisle Door to open (TRUE) or close (FALSE). |
| AisleDoorDone | <vcBoolSignal> | A Boolean signal that pulses TRUE when an operation, either open or close is completed. |
| AddTask | <vcStringSignal> | A String Signal that TaskScheduler uses to signal a task to the MainLogic |
| ReceiveComponent | <vcComponentSignal> | A component signal used by the crane to signal that a component (value of the signal) has been placed in the container of the material station. |
| InitComplete | <vcBoolSignal> | A Boolean Signal that indicates that initialization of the component is complete on starting or resuming the simulation |
| ChangeStatus | <vcBoolSignal> | A Boolean Signal behaviour that toggles between the two available Status configurations of the Machining Center (ASideOut/BSideOut) |
| ActionScript1 | <vcScript> | A python script behaviour that handles the functioning of the Automatic Pallet Changer (APC). |
| ActionScript2 | <vcScript> | A python script behaviour that handles the functioning of the (Aisle) Door. |
| TaskScheduler* | <vcScript> | A python script behaviour that handles the scheduling of tasks for the Machining Center |

| | | |
|---|---|---|
| MainLogic | <vcScript> | A python script behaviour that functions as the main script of the Machining Center and takes care of its initialization and executing tasks.. |

**Functioning Logic**

The machining center processes the machining pallets. The machining center has the AisleDoor door the opens to the aisle. . The aisleDoorOperate signal is triggered by the crane when it is assigned a task to either pick from or place to the machining center a machining pallet. This causes the Aisle Door to open.

Each machining center is such that it has two pallet holders (A and B). At any given time, the machining center can have one of two statuses, either “ASideOut” or “BSideOut”, or might be in the process of changing between these two statuses. “ASideOut” means that the pallet holder A is out, while “BSideOut” means that the pallet holder B is out. The pallet holder that is out is called the Changer while that what’s inside is known as the Table. Hence, when the machining center is in the “ASideOut” status, it the pallet holder A is the changer and B the table and vice-versa.

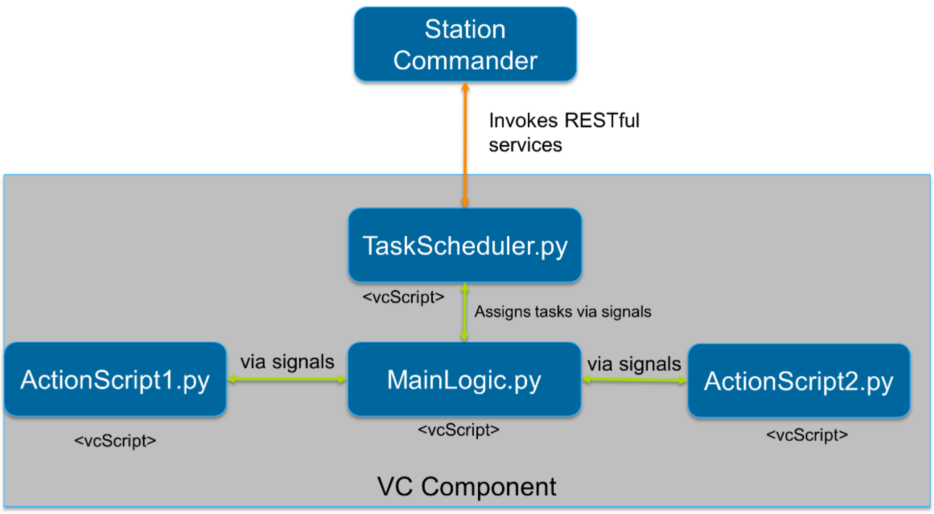


***Figure 34.*** *Scripting mechanism in the machining center model*

Once the machining center receives a pallet in the pallet changer, it executes all the required NC programs. The rotation of the pallet holders, and thus the changing of the status of the machining center is also a program known as “PalletChange.nc”. Sometimes, the machining center may tend to skip out on NC programs that are to be executed on the machining center due to its delay of arrival to the machining center. To avoid this, a TaskScheduler is implemented (as shown in Figure 34) that records all the tasks in real-time from the simulator and makes sure its carried out once the pallet arrives, although needless to say, if the pallet already has arrived on time it NC Programs are executed immediately.

### 5.4.4 Crane

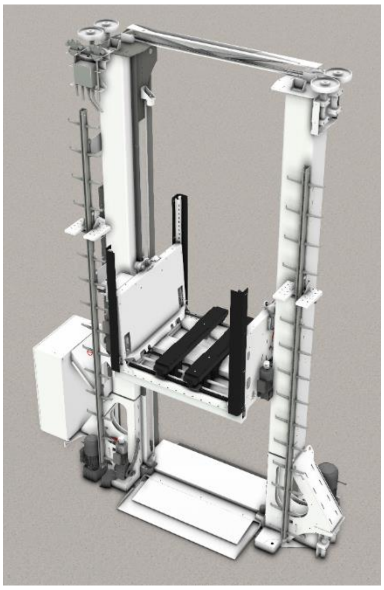

***Figure 35.*** *Crane Model*

***Table 10.*** *VC Properties of Crane*

| Name | Type | Description |
|---|---|---|
| CraneSpeed | <vcProperty> of type <VC_INTEGER> | Defines the Speed of the Crane |
| CraneAcceleration | <vcProperty> of type <VC_INTEGER> | Defines the Acceleration of the Crane |
| RefNode | <Ref<Component> | A property that links the crane to the storage area to which in acts as a means of transport |

***Table 11.*** *VC Behaviours of Crane*

| Name | Type | Description |
|---|---|---|
| RobotController | (vcRobotController) | RobotController is the controller that allows to control the crane and drive its joints |
| RobotExecutor | (vcRobotExecutor) | RobotExecutor is the executor of the crane |
| CartesianKinematics | <vcBehaviour> | CartesianKinematics is a forward and inverse kinematics solver for Cartesian type |
| Pick | <vcBoolSignal> | A Boolean signal that indicates if the current activity is picking. |
| Place | <vcBoolSignal> | A Boolean signal that indicates if the current activity is placing. |
| AddTask | <vcStringSignal> | A String signal that the TaskScheduler uses to signal the MainLogic the tasks. |
| TaskScheduler | <vcScript> | A script that functions s as the task scheduler obtaining tasks<br><br>real-time from the simulator and scheduling it for the crane in VC |
| MainLogic | <vcScript> | A script that handles the main logic for the functioning of the Crane, including initialization of state, execution of Pick and Place tasks and so on. |

**Functioning Logic**

The crane is modelled as a robot whose workspace is the storage area. The crane queries the Manufacturing Control System Simulator for active tasks repeatedly. It may be such that the crane in the simulator finishes tasks at a very rapid pace due to less visual fidelity in the control system web interface. This means that if the crane in the VC model only performs the current active tasks, it is likely to skip tasks from the control system Simulator if the crane there (in the Simulator) gets too far ahead.

For this reason a script called TaskScheduler is added to communicate real-time with the control system Simulator and schedules tasks for the Crane in the VC model. The script architecture is similar to that of the machining center shown in Figure 34.

### 5.4.5 Crane Storage

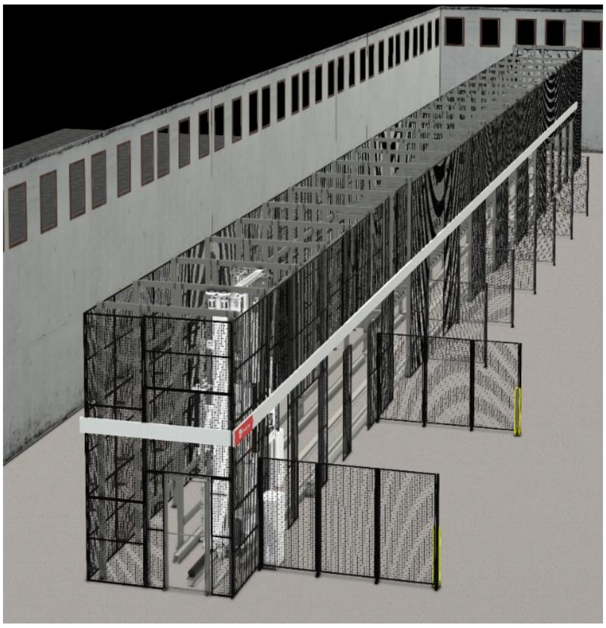

***Figure 36.*** *Crane Storage Model*

***Table 12.*** *VC Behaviours of Crane Storage*

| Name | Type | Description |
|---|---|---|
| Crane.Storage.X | <vcComponentContainer> | A component container is the location at which the pallet is stored in the Buffer, (X= instance of pallet storage,<br><br>e.g: Crane.Storage.1001) |
| MainLogic | <vcScript> | A script behaviour that handles the functioning of |

| | | |
|---|---|---|
| | | the Crane Storage byinitialization of state by creation and transfer of pallets |

### 5.4.6 Control Station

*Figure 37.* *Control Station*

*Table 13.* *VC Property of Control Station*

| Name | Type | Description |
|---|---|---|
| Address | <vcProperty> of type <VC_STRING> | A property that holds the IP address of the simulator for invoking the RESTful services |

*Table 14.* *VC Behaviours of Control Station*

| Name | Type | Description |
|---|---|---|
| MainLogic | <vcScript> | A script behaviour that pings the physical Control |

| | | |
|---|---|---|
| | | Station and checks for connectivity |

### 5.4.7 Completed Layout

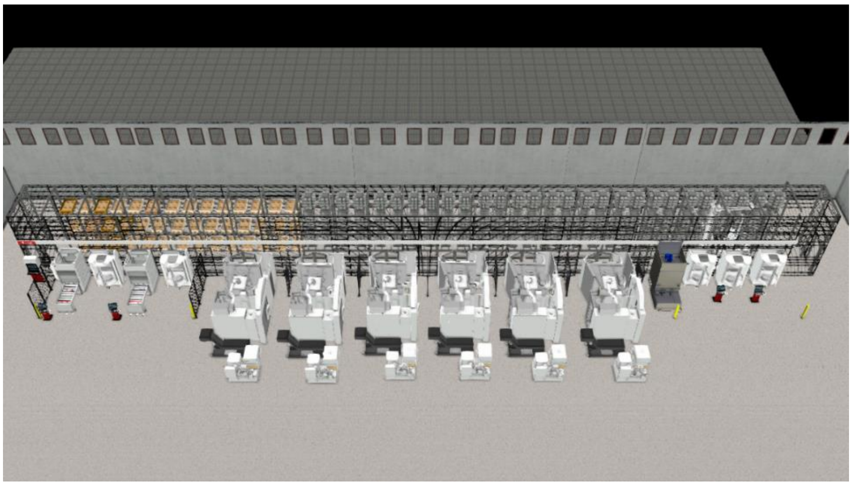

***Figure 38.*** *Completed Layout Models*

# 6. DISCUSSION

This section discusses the reviews the work done in the thesis and how the research questions presented in chapter 1 have been addressed. The subsections in the next section holds discussions pertaining to each of posed research questions.

## 6.1 Research Question: Summary of Findings

### 6.1.1 RQ1: Digital Twin Learning Experience and mitigating Learning Factory limitations

The learning method discussed in the earlier section involves a fine blend of passive and active learning. In the context of Flexible Manufacturing systems or manufacturing systems in general, we believe that starting with a passive approach to learning would help improve the quality of active learning in the subsequent stages by laying the foundation of manufacturing principles as hypotheses from where students can build on from the very first rounds of active learning. The flexibilities provided in an FM system is categorized into machine flexibility and routing flexibility. The use of Digital Twins as an educating tool is justified by the fidelity it comes with; it is a replication of the manufacturing system in its finest detail. The introduction of Digital Twins inside of the classroom allows the student to be able to hone their skills in a more intuitive and realistic manner. It allows for concrete experience and reflective observation as a part of experiential learning right from the class rooms environment. The environment is set up in a way that allows the replication of a system that offers realistic, mobile, and interchangeable situations that a student can use in order to learn about the manufacturing process. During the computer exercises the students have access to the same interfaces (Data Manager and Dashboard) as that in the factory floor. As such the upcoming discussion applies to the learning in the classrooms and also the physical site. The fact that the same discussion applies to both the classroom and site learning clearly brings out the advantages of the Digital Twin. The dashboard promotes learning and enforces a state of situational awareness as it displays real time updated statuses of all the components of the FM system. The dashboard is configurable and can be made to show the KPIs of the user's choice. As such, the students are constantly interacting with the system and engaged in constant analysis of different process-oriented, manual and semi-automated activities. They constantly monitor the cycle times, throughput, lead times and inventories and load material pallets as and when required. By being able to closely follow the unfolding events students in front of them the students understand the principles of flexibility in manufacturing systems; machine and routing flexibility. For example, the students are made to execute orders containing products requiring different machining tools. As the first order is being processed, more students begin to execute assigned orders. The Digital Twin Interface shows a clear picture of orders placed and the resulting workload. As such, the ability of the system to adapt to new orders (routing flexibility) is easily captured. The student is able to analyze how the schedule of the machining centers change with the arrival of the new orders by means of dedicated Gantt-charts for each machine. Sometimes a work piece may require tending from different machine tools (machine flexibly). This too is shown by means of a Tool Data Library that shows the location of each tool with its current status (ok, pre-warning, broken) and remaining life. Since these tools are used across multiple machines, the manufacturing system control automatically sums up the total usage and displays the remaining life to the user. As such the user develop insights on the future predictions of the state of the system, which is one of the key elements of situational awareness. Further, as the orders accumulate, bottlenecks are created in the system. When such changes occur, it automatically reschedules work allotted to the machining centers based on priorities assigned at the time of the creation of the order. It alerts the student operator of any missing resources and displays clear estimates for when the order will stop. Thus the students are able to reflectively observe the developments as in unfolds in real-time via the Digital Twin. Thus it can be seen how both the learning process and situational awareness are intertwined. In addition, as mentioned in the previous section, onto- logical models may be used with the Digital Twin model for the evaluation of learning objectives which in turn help converging the physical world changes with the intended learning objectives. This is the essence of the learning process involving Digital Twins.

As for mitigating the limitations of learning factories, the limited mapping ability of learning factories can be overcome as the digital twin of different manufacturing plants focusing on diverse domains  can be built with lessor effort when compared to having set similar learning factories. Secondly, digital twins have no space related issues as it is a virtual system. The configuration of the plant may be scaled endlessly with a only marginal cost--wise increment when compared to a learning factory. The same is the case when mapping different factory levels in the twin. Thirdly, for production orders having a high cycle time, the digital twin in the virtual configuration can be simulated at a faster speed to avoid waiting the entire duration of the product cycle-time. Fourthly, digital twins have no fixed locations as compared to learning factories as they are virtual systems, they can be further used simultaneously from many locations. Lastly, evaluation of learning outcomes or competencies can be automated in digital twins as opposed to the manual tests and interviews that the learners have to take at learning factories. More on this is covered in Section 6.1.3 and 6.1.4 addressing RQ3 and RQ4.

### 6.1.2 RQ2: Systematic Approach to Developing a Digital Twin

As a major part of the thesis was the creation of the digital twin, the approach in developing a twin of pedagogic significance can be seen as the research methodology (Section 4) pursued during the thesis. It kicked off with the envision stage which answered the first research question as to how the digital twin would prove beneficial in the academic setting. Next, was the identification of the process that would support the learning objectives of the course. These two steps were necessary before getting started to build the digital twin. While the first step was necessary for establishing concrete reasons to invest time and effort in pursuing the rest of the thesis, the second was necessary so that the value that the digital twin provided was aligned with the overall learning experience. This was followed by development of the pilot twin and its subsequent deployment that helped the process of creation from the developer's perspective. Domain knowledge and learning objectives are modelled and integrated into the digital twin to realize its pedagogic potential. The twin is later scaled to encompass the process in entirety. Lastly, the solution is monitored continuously and findings are published.

### 6.1.3 RQ3: Modelling Learning Outcomes in Digital Twin context

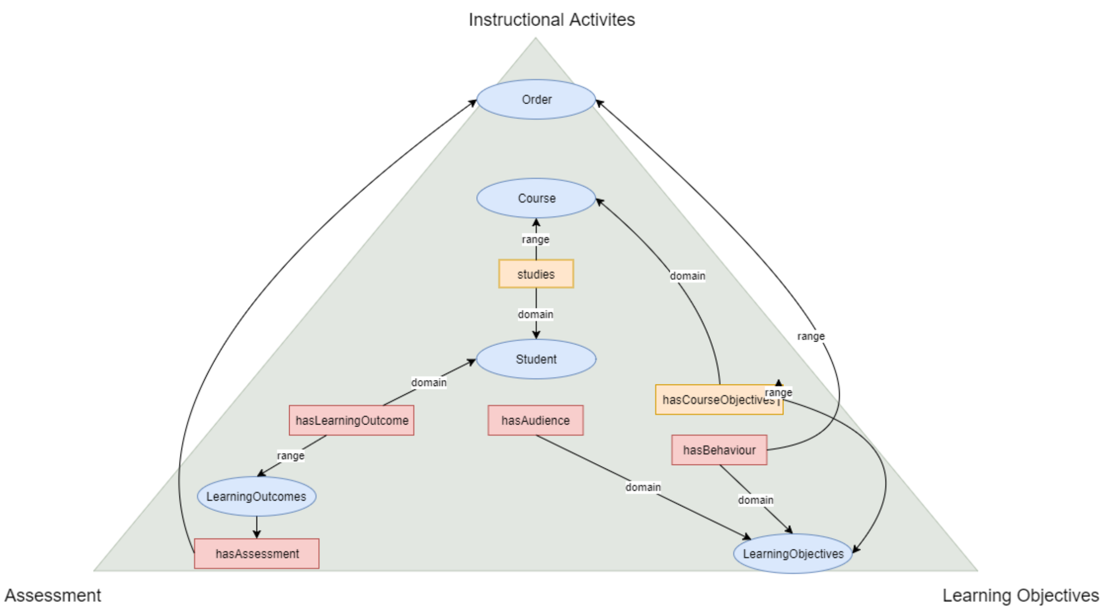


***Figure 39.*** *Instantiation of Bigg's Triangle of Effective Learning in ontology*

The ontological model developed in Section 5.2 essentially models the expected student behaviour (the intended learning objective) in our use case as the Order placed by the student in the manufacturing system. This is represented by the *hasBehaviour* object property of Class *LearningObjectives* and is populated by the digital twin based on the input from the course personnel. More on this is covered in the next section. The *LearningOutcome* Class consists of the actual Order placed by the student in the digital twin and *hasAssessment* as the same class as the behaviour of the learning objective.

If we compare the structure of the ontology schema Figure 39), we see that it conforms to Bigg's triangle of effective learning introduced in Section 2.5.2. Such an instantiation leads the author to believe that the underlying schema is effectively modelled to represent learning objectives and outcomes and more importantly align the assessment towards set objectives. The effectiveness is further accentuated by the fact additional **intended behaviours** of the student can be modelled as classes and attributed to students using the *hasBehaviour* object property of the *LearningObjectives* Class, while the **actual behaviour** of the student, which obviously is modelled by the same class as the intended behaviour , can be attributed to the student by the *hasLearningOutcome* object property of the *LearningOutcome* Class. This means to say any behaviour that can be modelled using ontologies can be instantiated and checked via the ontological reasoning mechanism used in this study.

### 6.1.4 RQ4: Student Performance Evaluation in Digital Twin context

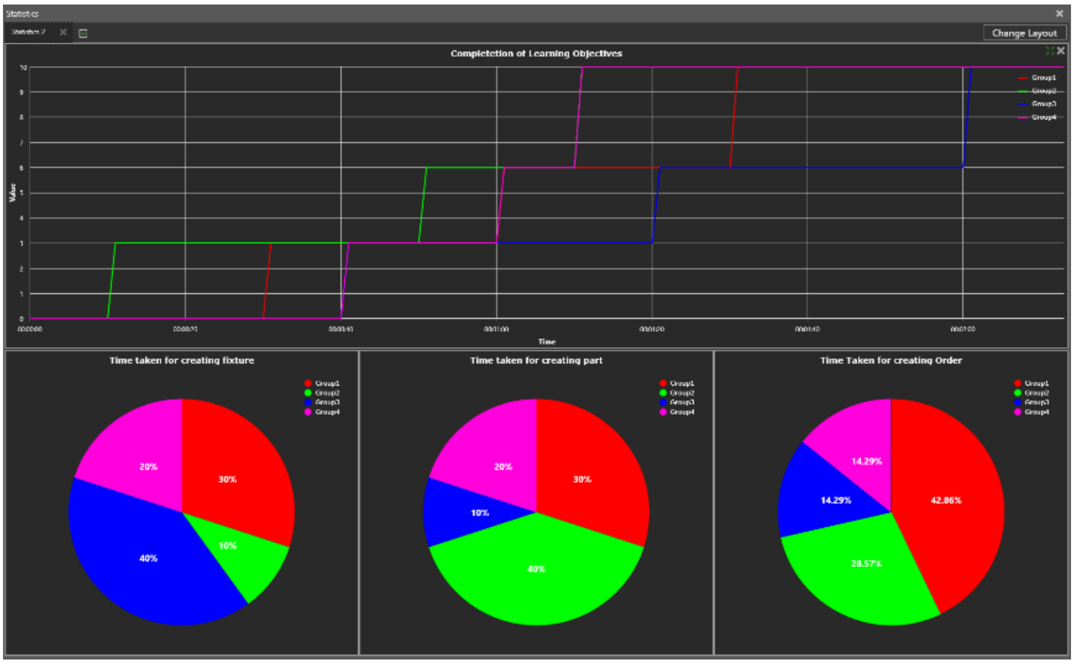


***Figure 40.*** *Envisioned Insights from the digital twin*

The envisioned UI for getting system insights is shown in Figure 40. Students all begin at the bottom of SOLO level 1, i.e. the pre-structural level, where the student has not grasped the concepts and has scattered information regarding the subject (assumption). Upon assessment of aligned course activities, the overall SOLO level of the student advances. After the lectures, SOLO points are assigned depending on the their performance in the online quiz. These are stored in individual student profiles in the knowledge base. The exercise in the laboratory extends on the SOLO points attained in earlier pedagogic step.

One intended learning outcome that is aligned towards the course objective defined would read "*At the end of the course, the student will be (1) competent in navigating through the* *user interfaces of FM systems, manage production requirements and gain key insights*" with a maximum SOLO points of 5. This SOLO points of 5 is attained in 2 parts, the quiz after the lectures is an activity that is of 2 SOLO points and the laboratory exercise that is of 3 SOLO points. For the laboratory exercise, the teacher inputs the order requirement for each students via an excel form generated by the digital twin. The digital twin populates the ontology with the order information as the behaviour of the learning objectives with the corresponding student as the audience. The order as modelled earlier contains parts, each of which contains a set of operations and its entailed processes that make the operation. Each of these tasks contains an assigned SOLO point which eventually contributes to the taxonomy level that the student will have attained have they executed correct tasks on the digital twin. Successful completion of both activities would result in the attainment of 5 SOLO points (not to be confused with SOLO level 5). These SOLO points is accumulated as a weighted average of individual activities and then mapped to the overall SOLO Level of the student at the end of the course.

We define progression in the SOLO levels as attainment of competence as did John Biggs that led to the proposed taxonomy in his work [67]. Figure 40 shows the envisioned insights. We see that as the student progresses with the creation of the order, the SOLO levels assigned to the sub-tasks are accumulated by the respective student. The student may not end up with the maximum SOLO "points" as an indication of not achieving the intended learning objectives due to his/her performance if the digital twin observes any divergence from the set learning objectives. Thus, the SOLO levels of various students may be compared as shown in Figure 40.

### 6.1.5 RQ5: Guiding Students to the desired skill level.

Based on the real-time insights the digital twin is able to guide the student based on the student's current status. For example, if the time taken for doing certain operations passes beyond a threshold set by the course personnel, learner and further task specific assistance may be provided based on the learner profile and the learner's current SOLO level.

The student also has real-time access of his/her progress and he/she materializes the orders in the system. For example, if a part has two operations and the first operation is made on the wrong fixture, the student is able to perceive this information during the time the student is creating the second operation. At this time, the student my choose to go back and edit the earlier configuration.

For the most part, this is a research question is left as future work whilst providing a concrete platform to getting started. The *hasCondition* property of the learning objective can be used to tie different objectives either in a sequence or by considering the difficulty of the task. The student if struggling with the current task may be assigned an easier task on the fly while notifying the course personnel.

# 7. CONCLUSION AND FUTURE RESEARCH

This chapter presents the contribution of the thesis and summarizes the thesis and presents the scope for future work on the same.

### 7.1.1 Thesis Summary

This research started with the observation that manufacturing pedagogy have failed to keep pace with the advancement of technology due to Industry 4.0. Learning Factories , in particular, that aimed at solving this posed several limitation that was identified in the Introduction chapter. Thus, a research journey began on ways to mitigate these limitations and further provide concrete learning experiences seeing digital twins as the ideal environment. The theoretical background, provided the reader with an abstract insight into several theoretical concepts necessary for the comprehension of the subsequent chapters in the thesis. This was also a precursor to the state of the art research in the following chapter and introduced certain tools and softwares used during the implementation. State of the art research was done in warranting digital twin as an educational tool. This also contributed to the envision and envisage stages of the research methodology while also addressing the first research question. The research methodology contributed with a systematic approach for the work done during the study while addressing the second research question. In the implementation chapter, all the details pertaining to the creation of the twin was put together. The following chapter holds a discussion on how all the research questions were addressed.

### 7.1.2 Future Research

As for directions of future research, the author proposes a few different directions taking into account the limitations mentioned in the first chapter.

The pedagogic extensions may be extended to encompass a wider range of competencies.

Although of less pedagogic significance, the performance of the digital twin may be improved using a better communication protocol, for e.g. websockets that leverage the publish-subscribe mechanism instead of polling.

# 8. REFERENCES


[1] J. S. Lamancusa, J. L. Zayas, A. L. Soyster, L. Morell, and J. Jorgensen, "The Learning Factory: Industry-Partnered Active Learning.," *J. Eng. Educ.*, vol. 97, no. 1, pp. 5–11, 2008.

[2] J. Cachay, J. Wennemer, E. Abele, and R. Tenberg, "Study on Action-Oriented Learning with a Learning Factory Approach," *Procedia - Soc. Behav. Sci.*, vol. 55, pp. 1144–1153, 2012.

[3] G. Chryssolouris, D. Mavrikios, and L. Rentzos, "The Teaching Factory: A Manufacturing Education Paradigm," *Procedia CIRP*, vol. 57, pp. 44–48, 2016.

[4] D. Popescu, C. Ionete, R. Aguridan, L. Popescu, Q. Meng, and A. A. Ionete, "Remote vs. simulated, virtual or real-time automation laboratory," *Proc. 2009 IEEE Int. Conf. Autom. Logist. ICAL 2009*, no. August, pp. 1410–1415, 2009.

[5] E. Abele, G. Chryssolouris, W. Sihn, J. Metternich, H. ElMaraghy, G. Seliger, G. Silvard, W. ElMaraghy, V. Hummel, M. Tisch, S. Seifermann, "Learning factories for future oriented research and education in manufacturing," *CIRP Ann. - Manuf. Technol.*, vol. 66, no. 2, pp. 803–826, 2017.

[6] M. Tisch, C. Hertle, J. Cachay, E. Abele, J. Metternich, and R. Tenberg, "A systematic approach on developing action-oriented, competency-based Learning Factories," *Procedia CIRP*, vol. 7, pp. 580–585, 2013.

[7] M. H. M. I. Hamid, M. Masrom, and K. R. Salim, "Review of Learning Models for Production Based Education Training in Technical Education," *2014 Int. Conf. Teach. Learn. Comput. Eng.*, pp. 206–211, 2014.

[8] A. Parrott and W. Lane, "Industry 4.0 and the digital twin," *Deloitte Univ. Press*, pp. 1–17, 2017.

[9] D. Kiritsis, A. Bufardi, and P. Xirouchakis, "Execution of SPARQL Query using Apache Jena Fuseki Server in AISHE domain," *Adv. Eng. Informatics*, vol. 17, no. 3, pp. 189–202, 2003.

[10] S. John, "Product Lifecycle Management," in *Product Lifecycle Management (Volume 2): The Devil is in the Details*, vol. 2, Springer International Publishing, 2016, pp. 1–35.

[11] J. Y. Mule, "Concept and Evolution of PLM," *Int. J. Appl. Inf. Syst.*, vol. 4, no. 3, pp. 25–28, Sep. 2012.

[12] L. Da Xu, W. He, S. Li, S. Member, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Trans. Ind. Informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[13] E. Glaessgen and D. Stargel, "The Digital Twin Paradigm for Future NASA and

U.S. Air Force Vehicles," *53rd AIAA/ASME/ASCE/AHS/ASC Struct. Struct. Dyn. Mater. Conf. AIAA/ASME/AHS Adapt. Struct. Conf. AIAA*, pp. 1–14, 2012.

[14] M. W. Grieves, "Product lifecycle management: the new paradigm for enterprises," *Int. J. Prod. Dev.*, vol. 2, no. 1–2, pp. 71–84, 2005.

[15] D. . M. Grieves, "Product Specification Management ( PSM ): Enabling Manufacturing Quality Whitepaper by Dr . Michael Grieves," no. July, 2008.

[16] D. . M. Grieves, *Virtually Perfect: Driving Innovative and Lean Products through Product Lifecycle Management*. .

[17] M. Grieves and J. Vickers, "DigitalTwin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems (Excerpt)," 2016.

[18] J. D. Hochhalter, W. P. Leser, J. A. Newman, E.H. Glaessgen, "Coupling Damage-Sensing Particles to the Digitial Twin Concept," no. April 2014, 2018.

[19] M. Grieves, "Digital Twin: Manufacturing Excellence through Virtual Factory Replication," pp. 1–7, 2014.

[20] P. Hehenberger and D. Bradley, *Mechatronic Futures*. .

[21] Z. Bi, L. Da Xu, and C. Wang, "Internet of things for enterprise systems of modern manufacturing," *IEEE Trans. Ind. Informatics*, vol. 10, no. 2, pp. 1537–1546, 2014.

[22] S. Kolla, D. Border, and E. Mayer, "Fieldbus networks for control system implementations," *Proc. Electr. Insul. Conf. Electr. Manuf. Coil Wind. Technol. Conf. (Cat. No.03CH37480)*, pp. 493–498.

[23] L. J. Wells, J. A. Camelio, C. B. Williams, and J. White, "Cyber-physical security challenges in manufacturing systems," *Manuf. Lett.*, vol. 2, no. 1, pp. 74–77, 2013.

[24] B. Barzani, "DESIGN AND DEVELOPMENT OF FLEXIBLE MANUFACTURING CELLS - A CASE STUDY," New Jersey Institute of Technology, 1993.

[25] R. T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," UNIVERSITY OF CALIFORNIA, IRVINE, 2000.

[26] "PubNub Websockets." [Online]. Available: https://www.pubnub.com/websockets/. [Accessed: 12-Oct-2018].

[27] "MQTT Essentials Special: MQTT over WebSockets | HiveMQ." [Online]. Available: https://www.hivemq.com/blog/mqtt-essentials-special-mqtt-over-websockets. [Accessed: 12-Oct-2018].

[28] S. Schnieder and T. Burke, "The Inside Story: How OPC UA and DDS Can Work Together in Industrial …." [Online]. Available: https://www.slideshare.net/RealTimeInnovations/the-inside-story-how-opc-ua-and-dds-can-work-together-in-industrial-systems. [Accessed: 12-Oct-2018].

[29] E. F. Codd and E. F., “A relational model of data for large shared data banks,” *Commun. ACM*, vol. 13, no. 6, pp. 377–387, Jun. 1970.

[30] A. Borgida and Alexander, “Description logics in data management,” *IEEE Trans. Knowl. Data Eng.*, vol. 7, no. 5, pp. 671–682, 1995.

[31] R. Reiter, “Towards a Logical Reconstruction of Relational Database Theory,” 1984, pp. 191–238.

[32] P. Agarwal, “Ontological considerations in GIScience,” *Int. J. Geogr. Inf. Sci.*, vol. 19, no. 5, pp. 501–536, May 2005.

[33] by Tim Berners-lee, J. Hendler, and O. Lassila, “The Semantic Web A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities.”

[34] C. Ullrich, “An Ontology for Learning Services on the Shop Floor,” *Proc. 13th Int. Conf. Cogn. Explor. Learn. Digit. Age. IADIS Int. Conf. Cogn. Explor. Learn. Digit. Age*, no. CELDA, pp. 17–24, 2016.

[35] H. Cheng, “Ontology-based web service integration for flexible manufacturing systems.”

[36] L. Fumagalli, S. Pala, M. Garetti, and E. Negri, “Ontology-Based Modeling of Manufacturing and Logistics Systems for a New MES Architecture,” *IFIP Adv. Inf. Commun. Technol.*, vol. 438, no. PART 1, pp. 192–200, 2014.

[37] A. Gómez-Pérez and O. Corcho, “Ontology Languages Ontology Languages Ontology Languages,” *IEEE Intell. Syst.*, vol. 17, no. 1, pp. 1–17, 2002.

[38] F. U. Jose L. Martinez Lastra Ivan M. Delamer, *Domain ontologies for reasoning machines in factory automation*. International Society of Automation (ISA), 2010.

[39] S. Staab and R. Studer, *Handbook on Ontologies*. 2007.

[40] “protégé.” [Online]. Available: https://protege.stanford.edu/community.php. [Accessed: 08-Oct-2018].

[41] R. Panchal and P. R.Swaminarayan2, “International Journal of Advance Engineering and Research Execution of SPARQL Query using Apache Jena Fuseki Server in AISHE domain,” *Int. J. Adv. Eng. Res. Dev.*, vol. 4, no. 9, pp. 387–395, 2017.

[42] Harris and Schwann, “Introduction: learning as a product, process and function,” *Sel. Readings Learn. Process. Oxford Univ. Press*, pp. 1–28, 1991.

[43] S. Freeman, S. L. Eddy, M. McDonough, M. K. Smith, N. Okoroafor, H. Jordt, M. P. Wenderoth, “Active learning increases student performance in science, engineering, and mathematics,” *Proc. Natl. Acad. Sci.*, vol. 111, no. 23, pp. 8410–8415, 2014.

[44] Y. Cui, “An Empirical Study of Learning Outcomes Based on Active Versus

Passive Teaching Styles," *Int. J. Educ. Manag. Eng.*, vol. 3, no. 1, pp. 39–43, 2013.

[45] D. Simandan, "Introduction: Learning as a Geographical Process," *Prof. Geogr.*, vol. 65, no. 3, pp. 363–368, Aug. 2013.

[46] A. Pritchard, *Ways of Learning*. .

[47] W. H. Wu, H. C. Hsiao, P. L. Wu, C. H. Lin, and S. H. Huang, "Investigating the learning-theory foundations of game-based learning: A meta-analysis," *J. Comput. Assist. Learn.*, vol. 28, no. 3, pp. 265–279, 2012.

[48] P. A. Ertmer and T. J. Newby, "Behaviorism, cognitivism,constructivism: Comparing critical features from an instructional design perspective," *Perform. Improv. Q.*, vol. 26, no. 2, pp. 43–71, 2013.

[49] B. F. Skinner, *Technology of teaching*. 1968.

[50] I. P. Pavlov and W. H. Thompson, *The Work of the Digestive Glands*. Charles Griffin, 1902.

[51] E. L. Thorndike, *The elements of psychology.* New York, NY, US: A G Seiler, 1905.

[52] J. B. Watson, *Behaviorism*. University of Chicago Press, 1930.

[53] B. F. Skinner, *Beyond freedom and dignity.* New York, NY, US: Knopf/Random House, 1971.

[54] J. Piaget, *The origins of intelligence in children.* New York, NY, US: W W Norton & Co, 1952.

[55] H. J. Freiberg and C. R. (Carl R. Rogers 1902-1987, "Freedom to learn." Prentice-Hall, Upper Saddle River, NJ, 1994.

[56] A. H. Maslow, *Toward a psychology of being*. New York: Van Nostrand, 1968.

[57] Kollbac and D. A. Kolb, "Experiential learning: Experience as the source of learning and development," *Organ. Sci.*, vol. 1, no. January 1984, 1984.

[58] M. R. Endsley, "Toward a Theory of Situation Awareness in Dynamic Systems," *Hum. Factors J. Hum. Factors Ergon. Soc.*, vol. 37, no. 1, pp. 32–64, 1995.

[59] G. Bedny and D. Meister, "Theory of Activity and Situation Awareness," *Int. J. Cogn. Ergon.*, vol. 3, no. 1, pp. 63–72, Jan. 1999.

[60] K. Smith and P. A. Hancock, "Situation Awareness Is Adaptive, Externally Directed Consciousness," *Hum. Factors J. Hum. Factors Ergon. Soc.*, vol. 37, no. 1, pp. 137–148, Mar. 1995.

[61] N. A. A. Stanton, P. R. G. R. G. G. Chambers, and J. Piggott, "Situational awareness and safety," *Saf. Sci.*, vol. 39, no. 3, pp. 189–204, 2001.

[62] M. R. Endsley, "Theoretical Underpinnings of Situation Awareness: A Critical

Review," *Situat. Aware. Anal. Meas.*, pp. 3–32, 2000.

[63] R. Heinich, *Instructional Media and Technologies for Learning*. Merrill, 1999.

[64] "Potential Assessments: Design ; Teach a Course - Eberly Center - Carnegie Mellon University." [Online]. Available: https://www.cmu.edu/teaching/designteach/design/assessments.html. [Accessed: 01-Nov-2018].

[65] J. Biggs and C. Tang, "Teaching for Quality Learning at University. Society for Research into Higher Education & Open University Press," *New Ed.*, 2003.

[66] S. Bloom Benjamin and D. R. Krathwohl, "Taxonomy of Educational Objectives: The Classification of Educational Goals, by a committee of college and university examiners. Handbook I: Cognitive Domain." New York, Longmans, Green, 1956.

[67] J. B. Biggs and K. F. Collis, *Evaluation the quality of learning: the SOLO taxonomy (structure of the observed learning outcome)*. Academic Press, 1982.

[68] L. W. Anderson, D. R. Krathwohl, P. W. Airasian, K. A. Cruikshank, R. E. Mayer, P. R. Pintrich, J. Raths, M. C. Wittrock "A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives, abridged edition," *White Plains, NY Longman*, 2001.

[69] J. Biggs, "SOLO Taxonomy." [Online]. Available: http://www.johnbiggs.com.au/academic/solo-taxonomy/. [Accessed: 11-Nov-2018].

[70] D. T. Matt, E. Rauch, and P. Dallasega, "Mini-factory - A learning factory concept for students and small and medium sized enterprises," *Procedia CIRP*, vol. 17, pp. 178–183, 2014.

[71] N. Díaz Rodríguez, M. P. Cuéllar, J. Lilius, and M. Delgado Calvo-Flores, "A fuzzy ontology for semantic modelling and recognition of human behaviour," *Knowledge-Based Syst.*, vol. 66, pp. 46–60, 2014.

[72] M. H. M. Noor, Z. Salcic, and K. I. K. Wang, "Enhancing ontological reasoning with uncertainty handling for activity recognition," *Knowledge-Based Syst.*, vol. 114, pp. 47–60, 2016.

[73] J. David, A. Lobov, and M. Lanz, "Leveraging Digital Twins for Assisted Learning of Flexible Manufacturing Systems," *IEEE 16th Int. Conf. Ind. Informatics*, pp. 529–535, 2018.

[74] A. Wolf, "Competency Based Education And Training," Falmer Press, London, 1989, pp. 39–53.

[75] S. M. Robinson and C. Barberis-Ryan, "Competency Assessment: A SYSTEMATIC APPROACH," *Nurs. Manage.*, vol. 26, no. 2, 1995.

[76] C. Bereiter and S. Engelmann, *Teaching disadvantaged children in the preschool*. Prentice-Hall, 1966.

[77] S. G. Magliaro, B. B. Lockee, and J. K. Burton, "Direct Instruction Revisited : A Key Model for Instructional Technology," vol. 53, no. 4, pp. 41–55, 2005.

[78] K.-Y. Ng, L. Van Dyne, and S. Ang, "From Experience to Experiential Learning: Cultural Intelligence as a Learni...," *Acad. Manag. Learn. Educ.*, vol. 8, no. 4, pp. 511–526, 2009.

[79] J. David, A. Lobov, and M. Lanz, "Learning Experiences Involving Digital Twins," *Proc. 44th IEEE Int. Conf. Ind. Electron.*