Spring 2023

A Semantic Web Approach to Fault Tolerant Autonomous Manufacturing

Fadi El Kalach

A SEMANTIC WEB APPROACH TO FAULT TOLERANT AUTONOMOUS MANUFACTURING

by

Fadi El Kalach

Bachelor of Engineering
Lebanese American University, 2020

Submitted in Partial Fulfillment of the Requirements

For the Degree of Master of Science in

Computer Engineering

College of Engineering and Computing

University of South Carolina

2023

Accepted by:

Amit Sheth, Director of Thesis

Ramy Harik, Reader

Amitava Das, Reader

Cheryl L. Addy, Interim Vice Provost and Dean of the Graduate School

Acknowledgments

I would like to express my sincere gratitude to my Major Professors and mentors, Dr. Amit Sheth and Dr. Ramy Harik for the continuous support and guidance of my master's study and research, for their patience, motivation, enthusiasm, and immense knowledge. The work done in this thesis would not have been possible without the support of the neXt Future Factories team who I have worked with throughout the last two years. I would like to especially thank Clint Saidy, Kaishu Xia, and Max Kirkpatrick. This work would also not have been possible without assistance from my colleagues at the Artificial Intelligence Institute (AIISC) and more specifically Ruwan Wickramarachchi. I would also like to recognize the funding and support of institutions and companies that believed in our work: the South Carolina Research Authority (SCRA), National Science Foundation (NSF), Siemens, IBM, and Yaskawa.

Abstract

The next phase of manufacturing is centered on making the switch from traditional automated to autonomous systems. Future Factories are required to be agile, allowing for more customized production, and resistant to disturbances. Such production lines would have the capability to reallocate resources as needed and eliminate downtime while keeping up with market demands. These systems must be capable of complex decision making based on different parameters such as machine status, sensory data, and inspection results. Current manufacturing lines lack this complex capability and instead focus on low level decision making on the machine level without utilizing the generated data to its full extent. This thesis presents progress towards autonomy by developing a data exchange architecture and introducing Semantic Web capabilities applied to managing the production line. The architecture consists of three layers. The Equipment Layer includes the industrial assets of the factory, the Shop Floor Layer supports edge analytic capabilities converting raw sensory data to actionable information, and the Enterprise Layer acts as the hub of all information. Finally, a full autonomous manufacturing use case is also developed to showcase the value of Semantic Web in a manufacturing context. This use case utilizes different data sources to complete a manufacturing process despite malfunctioning equipment. This provides an approach to autonomous manufacturing not yet fully realized at the intersection of three paradigms: Smart Manufacturing, Autonomous Manufacturing, and Semantic Web.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Smart Manufacturing has taken a front seat in the advancement of manufacturing production lines. This vision of Smart Manufacturing (SM) will be powered by Industrial Internet of Things (IIoT), Big Data Analytics, and Artificial Intelligence. These capabilities can have a substantial effect on the profitability of industry as a whole since manufacturers must respond to fast-changing requirements through productivity advancements and agility while maintaining high standards.

One aspect of manufacturing that SM seeks to tackle is the ability to minimize machine downtime. Downtime refers to the period of time that production is halted for a variety of reasons, one of which could be to perform maintenance on equipment. According to Forbes, unplanned downtime can cost up to $50 billion a year to industrial manufacturers. This presents a challenge to manufacturers in finding solutions to such problems. Correspondingly, industries are embracing Digital Transformation and SM to construct manufacturing systems capable of overcoming faults within and maintaining a continuous production line hence increasing efficiency and throughput. These solutions utilize IIoT, Data Analytics, and Artificial Intelligence as the driving force.

This thesis aims at creating a system capable of utilizing the generated data within the production line to tolerate faults or failures and hence minimize the downtime required for maintenance. More specifically, this thesis addresses how Semantic Web can be utilized to create a fault tolerant autonomous manufacturing system to minimize downtime.

We propose a novel approach to fault tolerant autonomous manufacturing that is capable of eliminating downtime of a production line through data and domain knowledge utilization. The proposed method uses Semantic Web technologies within manufacturing to bridge the physical and digital worlds. This technology takes advantage of Knowledge Graphs to incorporate sensor data with domain knowledge to react to machine failures appropriately while maintaining production.

This work can lead to a substantial broader impact on the manufacturing industry as it is a step towards adopting innovative technologies and standards for SM that can tackle one of the costliest issues manufacturers face. This work has applicability in any manufacturing site as issues with equipment failure can be present in any manufacturing facility whether pharmaceutical, automotive, or aerospace.

The thesis is organized as follows: Chapter two provides a review of research topics relevant to this thesis. This review covers topics such as Edge Computing, Digital Twins, Semantic Web, and Autonomous Manufacturing, before narrowing down the focus of this thesis. Chapter three describes the use case developed in this thesis and outlines some main capabilities that the assembled Future Factories testbed has. This also provides the theoretical framework for the developed use case. Chapter four dives into the details of the infrastructure of the testbed and the application development process. The results from the application are highlighted with a brief discussion of the effect this work has on manufacturing and its limitations.

Chapter 2

Literature Review

2.1 Introduction

We provide an introduction and literature review of three research areas relevant to this thesis. First, the field of Smart Manufacturing is introduced alongside the key topics of Edge Computing and Digital Twins. Secondly, Semantic Web concepts are introduced with an emphasis on different Semantic Web technologies and the integration methodology within manufacturing. Finally, the Autonomous Manufacturing research field is introduced highlighting its relevant capabilities. The final section introduces research at the intersection of these three topics that this thesis will cover in the remaining chapters.

2.2 Smart Manufacturing

2.2.1 Background

Smart Manufacturing or Industry 4.0 refers to the fourth industrial revolution in the manufacturing industry. Previously, the first to third Industrial Revolutions correlated to steam power, mass production, and finally IT automated production, respectively (Thoben et al., 2017). However, Smart Manufacturing aims towards building upon these iterations and integrating innovative technologies into manufacturing settings such as Internet of Things (IoT). With this, Industry 4.0 is replacing traditional automated systems with autonomous systems capable of information exchange, decision-making, and independent control. Research in Smart Manufacturing aims at the eight key areas of standardization, infrastructure, safety, work organization, continuous professional development, managing complex systems, regulatory framework, and resource efficiency (Dagerman, Wahlster, et al., 2013).

2.2.2 Edge Computing

Edge Computing introduces processing capabilities closer to the data source. It utilizes devices that are on the periphery of data generation rather than relying solely on cloud-based technologies. This is due to the efficiency of processing the data closer to the point of generation (W. Shi et al., 2016). Its main characteristics include fast processing and swift response time (Khan et al., 2019).

Edge Computing does not refer to one specific device, but rather processing location. It can encompass a plethora of resource-constraint devices such as Raspberry Pi's (Wan et al., 2018), Gateways (Wang et al., 2020), and regular computers (Zhang & Ji, 2020). Edge Computing complements Cloud Computing in latency-sensitive applications (G. Carvalho et al., 2021). Most of these applications require quick processing times to determine actionable decisions. Table 2.1 showcases different implementations of three distinct use cases for Edge Computing: Real Time Analytics, Job Scheduling, and Anomaly Detection.

Table 2.1: Use Cases for Edge Computing

Use Cases	Paper	Role of Edge	Device Used
Real Time Analytics	(Cao et al., 2017)	Descriptive analytics are used to uncover meaningful patterns from real-time data streams.	Cisco IR829 Industrial Integrated Services Router

	(Sirojan et al., 2019)	Event detection and analysis.	NI single-board controller (NI sbRIO-9637)
Job Scheduling	(Wan et al., 2018)	Energy consumption model related to the workload.	Raspberry pie board, UDOO board, ESP8266
	(Wang et al., 2020)	Algorithm execution	Smart Gateways
Anomaly Detection	(Zhang & Ji, 2020)	Abnormal situations detected using the trained LSTM.	Computer equipped with a 3.2 GHz Intel Core i7 processor and 8GB RAM.
	(Liu et al., 2021)	Anomaly detection algorithm.	MSP430 or ARM

This figure represents only a small subset of the work being done within Edge Computing but intends to showcase the variety of applications that can benefit from it, whether it was through deployment of machine learning models straight on edge (Schneible & Lu, 2017; Zhang & Ji, 2020) or data analysis algorithms with quick processing times (Cao et al., 2017; Liu et al., 2021; Wang et al., 2020).

2.2.3 Digital Twin

The first notion of a Digital Twin (DT) was introduced in 1964 with the development of Ivan Sutherland's *Sketchpad* (Ivan Sutherland, 1963). Since then, DTs have been adopted in different fields including manufacturing, construction, aerospace, automobile, and electricity (Qi et al., 2021). Even with this rich history, the term "Digital Twin" was not explicitly coined until 2011 by Michael Grieves (Michael Grieves, 2015). With the advancement of technology, DTs began to gain capabilities more complex than mere digital representations, but systems capable of analysis and affecting the physical system as well. Table 2.2 details a few applications of DTs in manufacturing.

Table 2.2: Digital Twin Implementations

Title **(Citation)**	**Description**
A Cyber-Physical Machine Tools Platform using OPC UA and MTConnect (Liu et al., 2019)	OPCUA based approach to integration of machine tools into a cyber physical data collection system
A Reconfigurable Modeling Approach for Digital Twin-based Manufacturing System (Zhang et al., 2019)	Presents modeling approach for reconfigurable manufacturing systems. Allows use of digital twin to explore possible options and issue corrective actions.
Analyzing bearing faults in wind turbines: A data-mining approach (Kusiak & Verma, 2012)	Machine learning approach to fault prediction in wind turbines. Predicted over-temp faults up to 1.5 hrs before fault. Models based on real data from wind turbines, using input parameters such as voltage phase, current phase, torque, and temperatures.
Motion planning and scheduling for human and industrial-robot collaboration (Pellegrinelli et al., 2017)	Dynamic path planning for human robot interaction. Optimization based on cycle times and equipment availability.
Petri-net-based dynamic scheduling of flexible manufacturing system via deep reinforcement learning with graph convolutional network (Hu et al., 2020)	Deep reinforcement learning approach to manufacturing operation scheduling. Multiple learning network approaches used against digital model.
A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces, and intelligence (Xia, Sacco, et al., 2021a)	Initial results towards using deep reinforcement learning applied to manufacturing systems – dynamic scheduling, vision, etc.
Machine Learning Based AFP Inspection: A Tool for Characterization and Integration (Sacco et al., 2019)	Presents methodology for automatic recognition of AFP defects using ML techniques.

2.3 Semantic Web

In the early years of the Web, information was designed strictly for human consumption. However, this meant that it was not interpretable by machines as well. Consequently, the Semantic Web was introduced as an extension to define information allowing interoperability between different systems and machines (Cardoso & Sheth, 2006). Semantic Web provides a framework to deal with massive scale, heterogenous, and dynamic data (Sheth & Ramakrishnan, 2003). Applications for Semantic Web can range from e-Learning (Markellou et al., 2005) to the EventWeb (Jain, 2008) (Sheth & Perry, 2008). This section will provide a brief background on Semantic Web concepts before exploring it in the context of manufacturing.

2.3.1 Background

2.3.1.1 Knowledge Graphs

The building blocks of the Semantic Web include entities linked together through relationships. This creates a Knowledge Graph (KG) that represents information in a structured form (Sheth et al., 2019). KGs can be used for enhanced search, browsing, integration and analysis of data.

A KG is made up of two main components, nodes, and edges. A node can represent an asset, person, place, or object. Edges on the other hand connect different nodes together and create the relationship between them. A KG has two parts, the Schema/Ontology and the Instantiations. A schema or ontology defines the classes to be used in the KG (Parsons, 2009). In essence, ontology defines the domain of discourse, and it consists of a finite list of terms and relationships. As for applications, KGs can be used in various fields from Deep Learning (Gaur et al., 2021), to Robot Knowledge Representation (Kho et al., 2014), to the medical domain (Xu et al., 2020).

2.3.1.2 Resource Description Framework

Resource Description Framework (RDF) is an international standard for Semantic Web data or represent Web resources. RDF provides a common framework for representing information to allow interoperability between applications. RDF resources are documented through Uniform Resource Identifiers (URI). It is made up of three types of resources: subject, object, and predicate as shown in Figure 2.1.

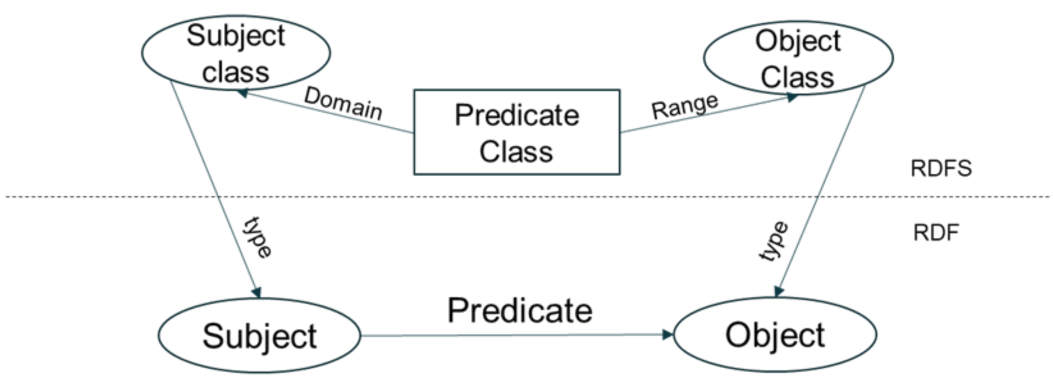

Figure 2.1: RDF Model

In the context of Semantic Web, RDF is used to define ontologies and create the instances needed to generate the overall KG.

2.3.2 Semantic Web in Manufacturing

Semantic Web technologies deliver explainable results to decision making and offer an approach to autonomous systems that provide the accountability and transparency required for manufacturing environments. Similarly, technological advancements of different types of sensors, equipment, and robots have led to the generation of large amounts of data during the manufacturing process that can provide insights on the status of the production line. Much work has been done to merge these two paradigms to support insights and decisions for manufacturing applications.

Ontologies were developed that can be adopted within a manufacturing context. The ManuService ontology was developed (Lu et al., 2019), based on service-oriented business interactions. An ontology was developed focusing more specifically on additive manufacturing (Dinar & Rosen, 2017). An engineering ontology model was developed on an enterprise level (Lin & Harding, 2007). The Bosch Industry 4.0 KG was created to aid with interoperability between different systems such as products, machines, equipment, and processes (Grangel-González et al., 2020). All these works focus on creating the schema to be adopted as a standard when creating KGs for manufacturing purposes.

Some works oriented these frameworks towards business planning and production order. A Semantic Web based architecture was developed to create an integrated business process model using OWL (Yang et al., 2005) and integration between a design house and manufacturer was established through a Semantic Web service for business interactions (Kulvatunyou et al., 2005). Different works also adopted Semantic Web to enable collaboration and resource allocation within a manufacturing facility (Cai et al., 2009)(Alsafi & Vyatkin, 2010)(Yang et al., 2005)(Balakirsky, 2015) .

Within these KGs, information is stored from different sources. One main advantage of KGs is the ability to integrate domain knowledge which provides data from external sources rather than from the shop floor. This information can be found publicly and integrated within a KG.

2.3.2.1 Manufacturing Use Cases

Based on the above studies, manufacturing use cases for Semantic Web can be identified to showcase the potential of the intersection of these two domains. These capabilities can be split into three distinct categories: Vertical Integration, Domain Integration, and Autonomous Systems. Each category will adopt traditional manufacturing use cases to show the enhanced capabilities that can be achieved. These use cases are Monitoring, Predictive Maintenance, and Quality Control and will be expanded upon within each category listed previously. Monitoring traditionally encompasses the act of supervising the manufacturing process and ensuring all assets are operating as they should. Predictive Maintenance is a field of research on its own that deals with the analysis of equipment data to predict whether and when in the future the maintenance will be required so as to minimize downtime and extend the assets' lifetime. Quality Control measures the quality of the product being manufactured and studies how best to maintain that specified or targeted standard of quality.

Vertical Integration

Vertical integration deals with the abstraction of heterogenous data sources in the manufacturing facility into one unified data model. It is the abstraction of all components of the factory from the raw data level to a data model that can be integrated with other components in the factory to create one homogenous and standardized data model, represented as a KG, that can be queried to obtain any information that the user requires. Unlike traditional IT systems that require significant manual effort to integrate data systematically.

Monitoring

With the introduction of the standardized data model that can be queried, monitoring becomes simple to achieve. One example is to create a centralized dashboard with signals obtained from the equipment and sensors alongside process signals and deduced knowledge. A concrete example could be the inclusion of sensor data, process signals (indicators such as job number and operating equipment), and deduced knowledge such as product state.

Predictive Maintenance

Since all data has been unified into a standard model, predictive maintenance algorithms can now have a more holistic view of the signals being produced by the equipment. For example, a certain asset equipped with a vibration sensor and temperature sensor and monitored by a thermal camera can all be easily utilized to derive a well-developed predictive maintenance model for that asset.

Quality Control

Traditional quality control relies on multiple variables obtained from the completed product to test the validity and quality before shipping. The unification of all sensor data aids in obtaining further indicators from the product testing. For example, a traditional quality control method might test for smooth surfaces using one sensor. However, an enhanced quality control method can utilize data from the sensors used in the quality check and during the manufacturing procedure to produce more holistic quality indicators.

Domain Integration

Domain knowledge denotes information from different fields obtained from external sources and not generated from the factory. This category outlines examples where the fusion of these two information sources can be of advantage for the traditional manufacturing use cases adopted in this section.

Monitoring

Dashboards would not only include signals that originate from the factory but can work towards adding further context. For a cross domain example, a dashboard can include current electricity rates, so technicians decide how best to run equipment to lower the cost of electricity usage. This can also include market trends and demands to help optimize manufacturing accordingly.

Predictive Maintenance

Manufacturer specifications provide ideal conditions and output ranges that can be used in training predictive maintenance models. The inclusion of these specifications alongside other domains knowledge such as temperature and humidity can allow models to predict maintenance schedules more accurately. An example of this can be the maintenance of a sensor placed in a rugged environment. The knowledge obtained about temperature, manufacturer specifications, and the generated data can all be combined in the training set.

Quality Control

A factory can focus more of its resources depending on market demands. More specifically, a factory that produces face masks and gloves can produce both in equal amounts. However, when market demand for masks increases while that of glove decreases, instead of increasing production of masks only and produce lower quality masks from over working the equipment, the factory will use the knowledge acquired from the market to reallocate resources and produce more masks and less gloves while maintaining the same standard of quality.

Autonomous System

This category is the culmination of all the abilities mentioned above. It combines the unification of local data sources with domain knowledge acquired from external sources. The following examples outline how these two paradigms can interact to create a full autonomous system within a manufacturing environment.

Monitoring

For monitoring, the system has concrete contextual information of all equipment in the factory. On top of that, it has domain knowledge about the equipment. One example of the use of this is to monitor the performance of sensors present in the factory. Should one be acting outside the manufacturer specified operational range, the system will suggest different sensors that can be ordered instead that might be more cost effective.

Predictive Maintenance

Predictive maintenance in an autonomous system introduces a new layer of capabilities that can minimize downtime of the production facility. The multimodal data generated can be used to create health indicators for different assets. The autonomous system can then allocate jobs based on the equipment health indicators, maintenance schedules, factory workload and orders. This can aid in prolonging equipment lifetime and minimize needed maintenance.

Quality Control

Quality indicators are derived from heterogenous sources in the manufacturing facility while keeping up with the different trends of the market. An autonomous system will allocate jobs while maintaining a certain standard of quality by considering market trends, machine health indicators and generated sensor data.

2.3.3 Integration of Semantic Web in Manufacturing

This section will look at the intersection between Semantic Web and Smart Manufacturing, more specifically detailing the integration methodology of Semantic Web technologies into manufacturing facilities.

2.3.3.1 Semantic Annotation

In manufacturing, sensors generate large volumes of which need to be annotated to be used. Raw sensor data demands annotation with semantic metadata to obtain contextualized information. Standardized annotation allows greater interoperability between machines, and more accessible data models. This annotation effort represents the translation of the raw sensor data to RDF, based on specific ontology mapping. Different ontologies for sensor data mapping have been developed in the context of manufacturing.

The OGC Sensor Web Enablement Ontology (Botts et al., 2008) was established by the Open Geospatial Consortium (OGC). It includes Observation & Measurement (O&M), Sensor Model Language (SensorML) and Sensor Observation Service (SOS). O&M and SensorML provide a standard model and XML schema for captured measurements. The SOS model offers a mechanism to query the observations and sensor metadata. Developed by W3C, the Semantic Sensor Network (Compton et al., 2012) ontology is a standard for modelling sensor devices, knowledge of the environment and observations and sensor platforms. Semantic Sensor Observation Service (SemSOS) (Henson et al., 2009) provides the knowledge base to derive higher level abstractions from the annotated sensor data. It retains the standard SOS specifications/service interactions while offering a semantic backend. SemSOS is the principal component of the Semantic Sensor Web (Sheth et al., 2008), the precursor to the SSN.

2.3.3.2 Integration Methodology

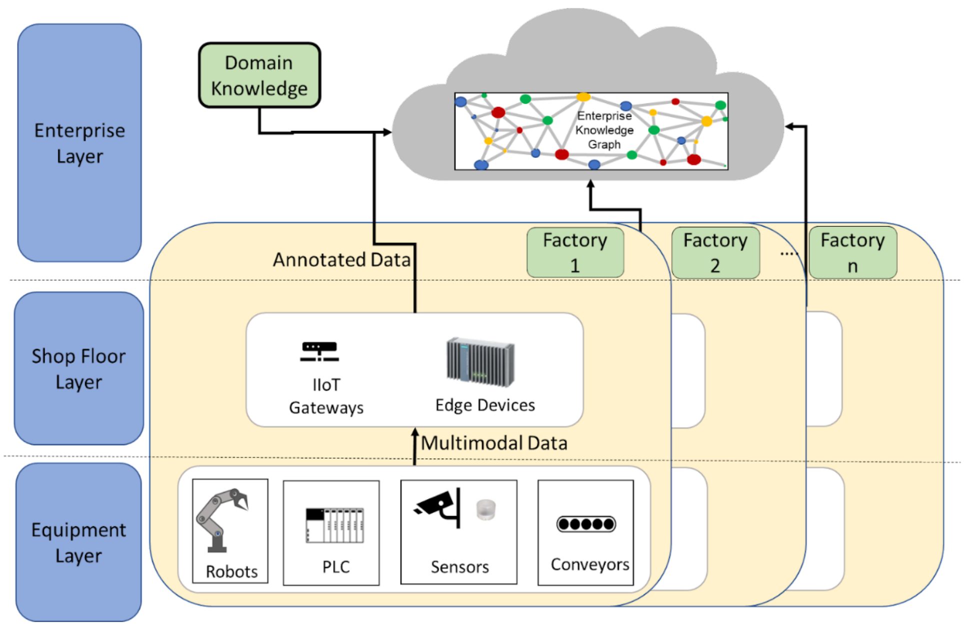

Figure 2.2: Layered Equipment Architecture for Factories

This section will outline the vision for Semantic Web integration in manufacturing that can be deployed on different devices and equipment. It consists of three layers shown in Figure 2.2. The first layer is the Equipment layer which contains all the manufacturing equipment such as Robots, Conveyors, and Programmable Logic Controllers (PLC). The second is the Shop Floor layer which introduces computational and processing capabilities and includes devices such as edge devices and gateways. Finally, the Enterprise layer introduces the cloud level and adds another and more powerful computational level. The Enterprise layer also acts as the hub of all knowledge whether it be domain knowledge or generated locally.

Equipment layer

The Equipment layer oversees communication within the industrial assets of the factory, providing the needed logic to advance manufacturing lines. State-of-the-art digital transformation techniques are fundamental to automate rapid understanding of acquired data supplied by the physical infrastructure. For instance, industrial sensors can be installed on legacy equipment to acquire process variables (throughputs, wastes, product levels, material feeds, visual inspection, etc.) and machine health variables (temperature, vibration, force, etc.).

PLC's can monitor process signals such as control logic signals which indicate the machine states, torque signals, safety signals, and actuator signals. The Open Platform Communications (OPC) protocol (Leitner & Mahnke, 2006) can be used to send these variables to different equipment which allows interoperability between manufacturing systems (Xia et al., 2019a). On top of that, advanced sensing technologies can allow state acquisition. An example of this would be visual monitoring systems (Saidy et al., 2020) (Xia, Saidy, et al., 2021a) which include high-resolution security cameras, thermal infrared cameras, and wireless inspection cameras.

Shop Floor layer

The Shop Floor layer introduces edge analytic capabilities on real-time collected data. Data from sensors on industrial assets are typically in a raw format and do not provide contextualized information. Annotating the data at the level of devices near the source could be sufficient to integrate the raw data from heterogeneous sources.

RDF can be utilized in this scenario. Adding Semantic Web technologies to resource constraint devices and the study of multiple semantic representations for sensor measurements with regards to energy efficiency of data communication and processing were evaluated (Su et al., 2015) and concluded that Entity Notation (EN) (Su et al., 2012) and JSON for Linked Data (JSON-LD) are adequate representations of RDF.

Different formats can be transformed to RDF using SDM-RDFizer (Iglesias et al., 2020) for uniform reasoning of sensor data. One approach for resource constraint devices is to format RDF from Sensor Markup Language (SenML) which supports JSON and Efficient XML Interchange (Su et al., 2014). In addition to this, a binary XML format was developed due to the difficulty in storing RDF on resource-constrained devices from the textual representation of RDF. However, prevalent reasoning mechanisms such as Jena reasoning engine (Ameen et al., 2014), Pellet (Sirin et al., n.d.), RacerPro (Haarslev et al., 2011), and Fact++ (Tsarkov & Horrocks, 2006) used with KGs have limited use within current edge devices due to processing complexity.

Enterprise layer

The Enterprise layer can perform complex decision-making using edge analytics, domain expertise, and global knowledge bases. It is also responsible for communication among devices at the Shop Floor layer, allowing them to build their own networks and perform autonomous decision-making. The Enterprise layer consists of processing-intensive cloud components with the help of the Enterprise KG (EKG). The EKG acts as the hub of information from joint departmental efforts involving backlog of events, analytic outputs, sensor readings, and domain expertise. This knowledge is required for any emergent production tasks and future occurrences of similar tasks. The EKG may also be dynamic and capture new knowledge.

Streamed Linked Data (Le-Phuoc et al., 2011) allows the storage and processing of continuous streams of factory states which provides the ability to update physical and virtual components of the framework to enhance productivity and remove faulty/irrelevant segments (Tao et al., 2017). This introduces the concept of Dynamic KGs (Pujara & Getoor, n.d.) (Das et al., 2018)(Padhee et al., 2018) that can update a triple in a KG when new sensor data is acquired.

2.4 Autonomous Manufacturing

2.4.1 Background

Autonomous systems can execute high level tasks without human intervention (Rosen et al., 2015). It refers to the complete automation of decision making (Bourne & Mark S. Fox., 1984). Different definitions have been given to the term autonomous manufacturing such as the “ability to identify their status and capability autonomously, to collect data, and to take decisions according to the changes of the manufacturing system”(Ding, Lei, Chan, et al., 2020), “the autonomous communication and collaboration between the WIP and the machines during production”(Park & Tran, 2011), and employing “Industry 4.0 technologies like the Industrial Internet of Things (IIoT), artificial intelligence (AI), and data analytics to modify and optimize production on the run”(David Rand, 2021). Each of these definitions tackles autonomous manufacturing from different perspectives whether it is the ability to autonomously create decisions, communicate with other equipment, or optimize production.

2.4.2 Approaches to Autonomous Manufacturing

Several earlier efforts have sought to implement Autonomous systems in a manufacturing environment. Many of these works adopt the smart manufacturing concepts discussed in this chapter and integrate further AI principles to create a decision-making model for manufacturing lines. This section will highlight a few of these works grouped together by the technologies used to create the autonomous system.

As autonomous systems involve decision making processes, and need to be deployed without explicit programming, Machine Learning (ML) was widely adopted as a common approach to build an autonomous manufacturing process. In fact, a large portion of the research community is focused on the development of new algorithms in manufacturing (Sharp et al., 2018). ML is a field of research focused on learning systems and algorithms (Qiu et al., 2016). ML aims towards providing the machine with the capability to create data-driven decisions by inputting a large amount of data so the created ML model can leverage that data.

ML models can be split into three categories: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning requires a training data set with labeled inputs and their corresponding outputs. During training, the model will learn the relationship between the inputs and outputs and will eventually be able to deduce the output for new inputs based on the learned inputs. Unsupervised learning attempts to detect patterns not outlined previously with minimal human supervision. Unsupervised learning does not use labeled data but models' probability densities based on the input data set. Reinforcement learning relies on an agent learning the correct decisions by reward accumulation.

2.4.2.1 Reinforcement Learning

ML was applied in a manufacturing system that builds parts according to user-specified performance indicators (Alam et al., 2020). The work developed supervised and reinforcement learning algorithms deployed on digital counterparts of the manufacturing process to assess performance with little data generation. A reinforcement learning model was developed as a means of successful collaboration between different autonomous robots working within the same shop floor (Agrawal et al., 2021). Similarly, a reinforcement learning model was trained on a DT of a manufacturing shop floor to ensure no collisions occur (Xia, Sacco, et al., 2021).

2.4.2.2 Supervised Learning

Supervised learning models were developed in the context of manufacturing to reduce cost and waste while increasing productivity of a cutting tool in machining lines (C. P. de Carvalho & Bittencourt, 2021). A supervised learning approach was adopted that learns from previous events to allow variable levels of autonomy for a robotic arm conducting object relocation tasks (Wheeless & Rahman, 2021). Supervised learning was used to inspect the elements in an 'inspection by exception' methodology that only inspects manufactured parts that cannot be categorized above a certain confidence level whether the part is healthy or unhealthy (Papananias et al., 2020).

2.4.2.3 Unsupervised Learning

The above model was also coupled with an unsupervised learning model that is trained to categorize the healthy and unhealthy parts as well (Papananias et al., 2020). An unsupervised learning model was adopted as a tool for dynamic task allocation of unmanned surface vehicles in constrained environments. This was done through splitting up the task management into two categories, task allocation and task execution (Ma et al., 2021). Supervised and unsupervised learning models were coupled to monitor performance by classifying defects. The traditional supervised learning model is trained to detect any defect while the unsupervised learning model is used to ensure that the data set that the first model is trained on is still relevant and add new defects that may arrive achieving a truly autonomous inspection capability (Banf & Steinhagen, 2022).

2.4.3 Characteristics of Autonomous Manufacturing

Autonomous manufacturing systems can be summarized by some characteristics that highlight their capability. An autonomous system can be fault-tolerant to a faulty sensor by preventing a temporary breakdown of a line or factory by suggesting a replacement sensor with similar functionality (Thuluva et al., 2017). This system can also dynamically allocate resources at runtime (self-organization), rather than pre-allocating. This can be done while considering factors such as the machines' current conditions, machine availability, maintenance schedule, and customer orders. Resource allocation could also consider external electricity rate data reduce factors such as energy consumption and carbon footprint by offsetting carbon emissions. Furthermore, when production demands lead to the introduction of a new machine in factories, it can simply participate by announcing its services and features during the resource allocation process (agile manufacturing). This illustrates the agility of a factory, where a new machine can be integrated in a plug-and-produce fashion according to market demands with minimal downtime. Based on these characteristics, a total of twenty-two autonomous manufacturing publications were selected and categorized. These papers are shown in Table 2.3.

Table 2.3: Categorized AM Implementations

Paper	Description	Self-Organizing	Agile Manuf.	Fault Tolerant
(Iwamura & Sugimura, 2010)	Real Time Scheduling of tasks for AGV's	x		
(Grundstein et al., 2017)	Control of order release, sequencing, and capacity control	x		
(Hildebrandt et al., 2016)	Flexible production system in complex pick and place tasks.		x	
(Wada & Okada, 2002)	Execution control system for dispatching tasks	x		
(Moergestel et al., 2011)	Agile production of different equipment.	x	x	
(Tharumarajah, 1998)	Adaptable Scheduling of Manufacturing Lines	x		
(Goldsmith & Interrante, 1998)	Distributed scheduling systems for collaboration and improved performance	x		
(Bourne & Mark S. Fox., 1984)	Control and planning of shop floor jobs	x		
(Cao et al., 2020)	Autonomous Distributed Systems for personalized product manufacturing.	x		
(Aly et al., 2010)	Workspace design to optimize autonomous production.		x	
(Jarvis, Jarvis, Lucas, et al., 2001)	Autonomous execution of robot assembly processes	x	x	
(Ding, Lei, Zhang, et al., 2020)	Collaboration based on optimal production decisions.	x		
(Jarvis, Jarvis, McFarlane, et al., 2001)	Resources-dependent autonomous assembly process	x	x	
(Schuster et al., 2017)	Autonomous assembly process by two robotic arms	x		
(B. Scholz-Reiter et al., 2009)	Compensate for manual intervention in manufacturing process.			x
(Park & Tran, 2011)	A manufacturing process capable of adapting to disturbances			x
(Hu et al., 2020)	Dynamic Scheduling for flexible manufacturing systems	x		
(Agrawal et al., 2021)	Collaboration between different autonomous robots working within the same shop floor	x	x	

(Xia, Sacco, et al., 2021a)	Ensure no collisions in manufacturing shop floor	x		
(C. P. de Carvalho & Bittencourt, 2021)	Reduce cost and waste while increasing productivity of a cutting tool in machining lines	x		
(Wheeless & Rahman, 2021)	Variable levels autonomy for a robotic arm carrying out object relocation tasks	x		
(Alam et al., 2020)	Builds parts according to user-specified performance		x	
Total		17	7	2

The selected papers were published between 1984 and 2021 to attempt and achieve a holistic view of the history of research in autonomous manufacturing. The publications are also collected from a wide variety of sources ranging from IEEE journals to the Journal of Manufacturing Systems to Conference Proceedings. This was in part to attain a sample pool representing all different fields of research that autonomous manufacturing can be relevant in. Table 2.3 displays a stark contrast between the number of efforts aimed towards implementing a self-organizing and flexible system compared to fault tolerant systems. The capability of self-organization and flexibility can be found in 17 and 7 publications respectively, whereas fault tolerance can only be found in two of them.

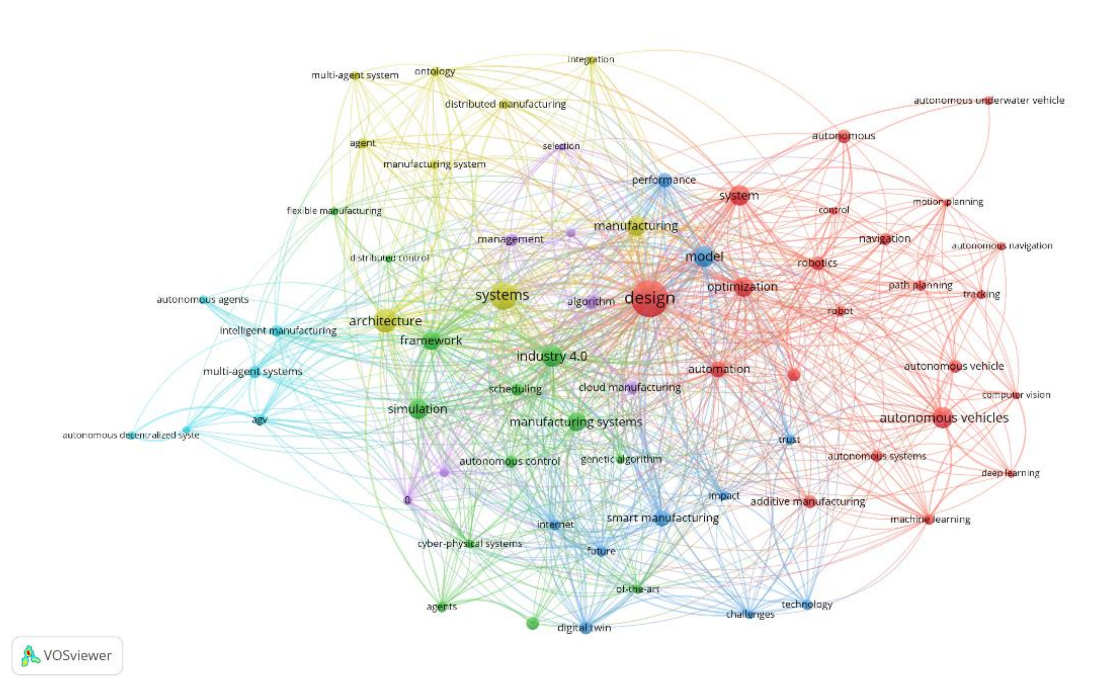

Figure 2.3: AM Research Field Overview

To gain a more complete perspective of the autonomous manufacturing research field, the network in Figure 2.3 was generated to give a more comprehensive perspective of the field of the autonomous manufacturing research. This network is a compilation of keywords from one thousand autonomous manufacturing papers. These papers were gathered from the Web of Science search engine and all the keywords were extracted. These keywords were then visualized using VOSviewer (https://www.vosviewer.com/). The keyword must have appeared at least ten times to show up in this network. A link between keywords signifies that they have appeared in the same paper. From this network, different research areas can be pinpointed including Autonomous Vehicles, Smart Manufacturing, Multi-Agent Systems, Distributed Manufacturing Systems, and Industry 4.0. The main outcome from this network is the shortage of Semantic Web terminologies with the only one present being "ontology." This does not mean that no work has been done at all but that the work has not been explored excessively or has not produced many significant outcomes.

2.5 Discussion and Gap Assessment

From the above analysis, there are different research gaps that could be addressed in this thesis. The first being research work aiming at accomplishing fault tolerance. Among the publications reviewed, only two previous implementations had relevant work to fault tolerance. This represents a small sample size of the overall pool even though this capability is of significant value to manufacturers. Hence, this thesis will attempt to achieve an autonomous manufacturing use case that can showcase the fault tolerance capability. In addition to that, Semantic Web has not been a widely adopted approach to witness such capability, with previous works adopting Semantic Web for collaboration (agile manufacturing) and resource allocation (Self-Organizing).

Additionally, there are five phases in utilizing KGs: Knowledge Acquisition, Knowledge Fusion, Knowledge Processing, Knowledge Storage, and Knowledge Utilization. Within a sample pool of collected works involving Semantic Web and manufacturing, none had fully undergone the five phases. (Teern et al., 2022). In addition to that, a lack of effort has been observed regarding integrating heterogenous sources such as sensors, material required, work orders, and quality of material (Yahya et al., 2021). This thesis will be tackling all five of the above phases within a manufacturing environment and integrating different sources into one central KG. To be able to achieve that goal, this thesis aims to find the intersection between the three research fields discussed in this chapter by utilizing Smart Manufacturing and Semantic Web techniques to tackle the identified gap. The approach is detailed in Figure 2.4.

Within Smart Manufacturing, this thesis covers the implementation of the state-of-the-art testbed infrastructure in the Future Factories lab. This includes Edge Computing and Digital Twin capabilities that will be utilized for the use case. The Semantic Web integration methodology outlined in this chapter will also be employed to realize this use case. Chapter three will outline the process of developing this use case in greater detail.

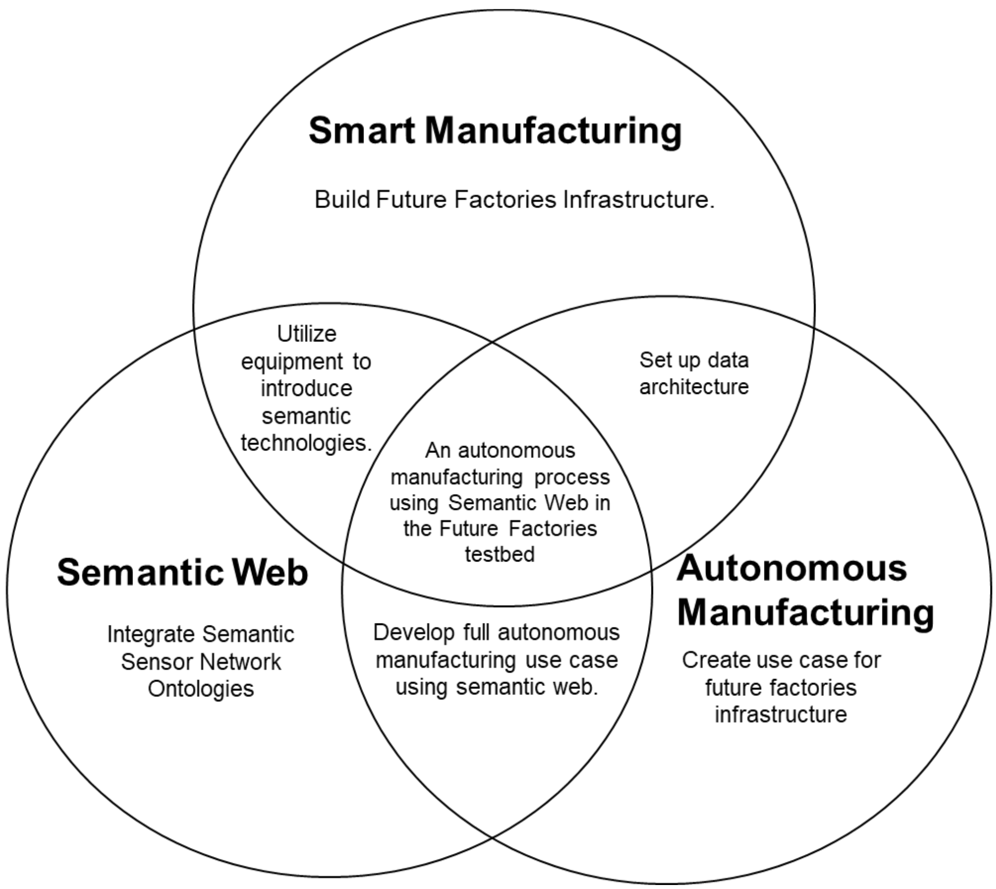

Figure 2.4: Contribution Outline

Chapter 3

Theory and Development

3.1 Introduction

As described in Chapter two, this thesis focuses on creating a manufacturing testbed in the Future Factories (FF) lab at the University of South Carolina and utilizing Semantic Web technologies on this testbed to realize an autonomous manufacturing use case. This section will narrow down the use case adopted in this thesis. Following that, various aspects of the developed testbed are introduced that showcase its capabilities and how they fit in with the overall use case. Finally, this section will cover the implementation plan of the application developed in this thesis.

3.2 Autonomous Manufacturing

3.2.1 Use Case

To showcase an autonomous manufacturing process, this thesis narrows down one characteristic of those outlined in the previous section and adapts it to the FF lab. More specifically, a fault tolerance use case is developed.

The developed application can be broken down into the significant parts shown in Figure 3.1. In order for fault tolerance to be realized, the existing manufacturing assets within a factory floor must react appropriately to a sudden malfunction of a certain sensor. This reaction allows the process to continue without any downtime for maintenance or replacement. For this to happen, assets must be aware of the different data sources available that can be utilized. Semantic Web presents an ideal solution to this issue as all the different information mentioned can be stored in interoperable fashion ready to be used whenever needed. It also presents greater advantages than the minimal existing cases of fault tolerance which do not utilize Semantic Web as it provides a standardized approach and model which can be modified to different knowledge discovery applications.

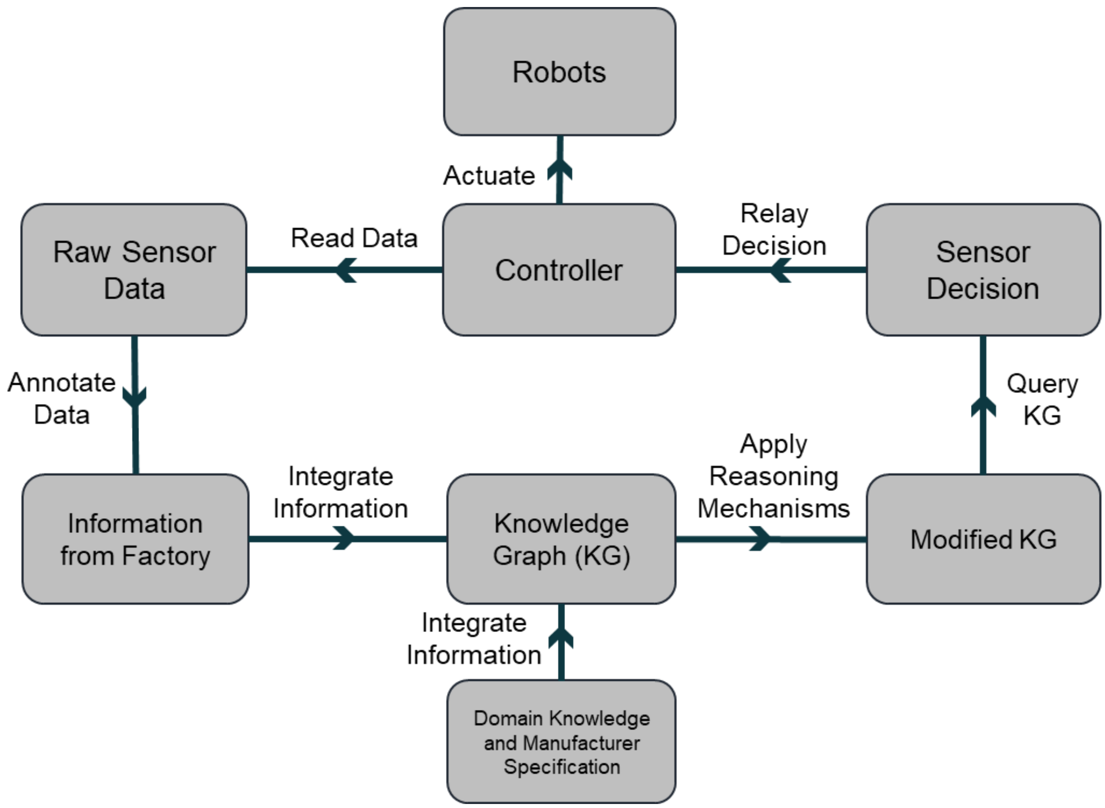

Figure 3.1: Semantic Web Application in Manufacturing

This application begins with the reading of the raw sensor data by the controller present within the facility. Once that data is acquired it is semantically annotated using a user defined mapping derived from the Semantic Sensor Network ontology(Compton et al., 2012), a standard ontology to represent sensor data. This information is then integrated with other information including the domain knowledge of the facility and manufacturer specifications. In this case, the domain knowledge includes the different sensors that can be used for a certain process. The manufacturer specifications outline the output range of a normally functioning sensor. Once all this information is integrated into one KG, a reasoning process can help deduce whether the sensor is functioning properly. This is done by comparing the data read from the actual sensor with the manufacturer specification output range present in the KG. A corresponding triple is added to the KG which shows the sensors functionality status. Based on this, the query engine decides which sensor to use based on the domain knowledge present in the KG. This decision can then be relayed to the controller. Semantic Web also improves upon traditional exception handling which requires every exceptions to be hard coded. This leadng to numerous lines of code. The different components built in this application can also be used for different capabilities minimizing the need for redundant work.

3.2.2 Requirements

To realize this autonomous manufacturing use case in the FF lab, a certain set of requirements are needed. The first requirement deals with the physical infrastructure needed to implement the use case on. To satisfy this requirement, work must be accomplished that integrates different manufacturing and processing equipment together to achieve a full testbed. These devices should have open communication channels and must have equipment with the processing ability to handle the required data. These requirements are met through the developed testbed. The second requirement is from the Semantic Web aspect. In this requirement, the work must integrate the needed data. It will show how locally generated data can be semantically annotated, then integrated to create the KG. This KG is then used to come up with decisions for the manufacturing process. These steps are discussed in more detail in the remainder of this chapter.

3.3 Future Factories Testbed

Figure 3.2: Future Factories Lab Set Up

The testbed in the FF lab at the University of South Carolina's McNair Aerospace Research Center is designed to be a testing environment for different technologies to be integrated in smart manufacturing processes. The testbed is a platform to introduce state of the art tools in robotics, the Industrial Internet of Things (IIoT), data analytics, and edge computing. This testbed is made up of four conveyor belts (Figure 3.2) assembled to provide maximum flexibility with respect to manufacturing processes. Currently, this testbed can accomplish a simple assembly process of a custom 3D-printed model rocket. This assembly process is achieved through cooperation between the different equipment in the testbed. The procedure, while simple, is intended to show the capabilities of the different platforms and assets in the testbed. The testbed exhibits all the requirements needed for the autonomous manufacturing use case to be applied to, with communication between devices set up and the availability of processing-intensive devices. This section will cover the characteristics of the testbed alongside a background on the relevant tools integrated within.

3.3.1 Characteristics

The testbed created was built with a few characteristics in mind. The infrastructure is intended to be reusable, meaning it can be utilized to realize different use cases. Since this testbed is supposed to be the foundation for future manufacturing research, it can be exploited depending on the required use case. For example, this testbed can be used to undergo research on edge computing or digital twin. The testbed uses communication protocols which can be configured with devices from different manufacturers. This testbed has the capability to be autonomous such that different Machine Learning techniques can be deployed on the equipment for autonomous decision making. The testbed is also cross-domain since it utilizes equipment from various manufacturers such as Siemens, IBM, Dell, Yaskawa and others.

3.3.2 Digital Twin

Digital Transformation has become more prevalent in the manufacturing industry over the last years. One aspect of Digital Transformation is the concept of a Digital Twin (DT). Essentially, a DT is the virtual counterpart of a physical facility. It can be anything from a simple CAD model of the facility to a complex system capable of two-way communication from the virtual to the physical entity that can undergo real-time decision-making or predictive analysis that affects the physical system. Currently, the FF testbed has some DT capabilities with the integration of *Siemens Tecnomatix Process Simulate* as the primary DT tool. *Process Simulate* is a software capable of modelling the physical testbed in virtual space as seen in Figure 3.3.

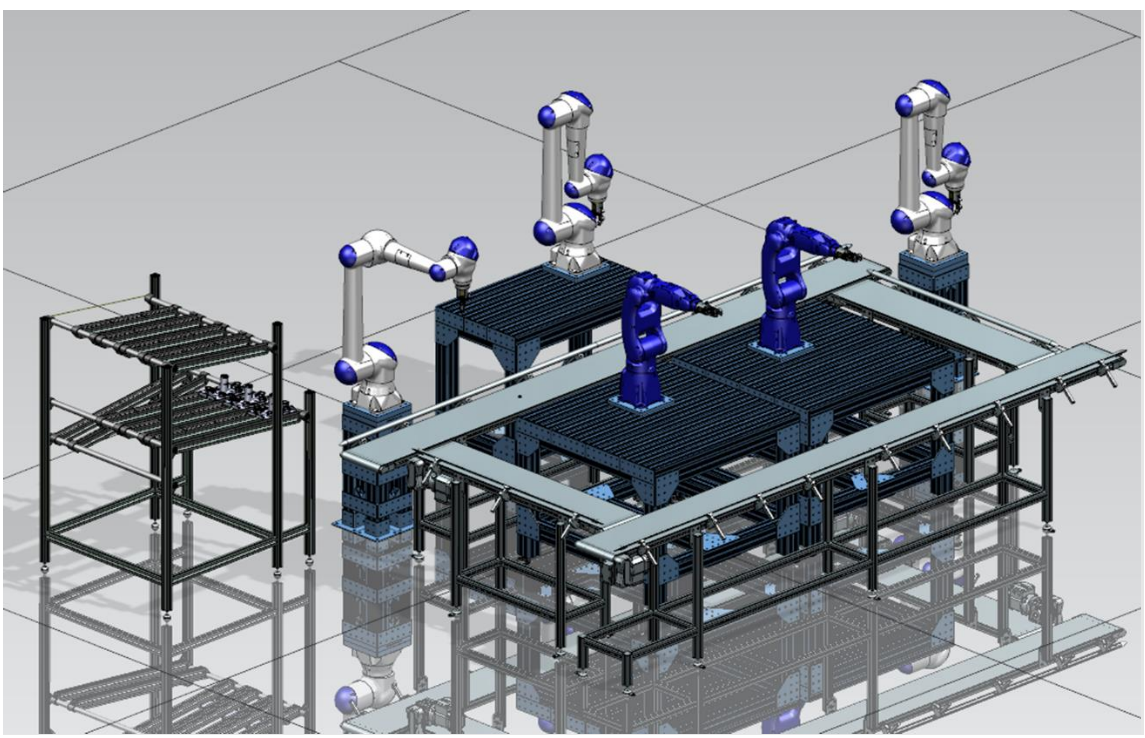

Figure 3.3: DT of the FF Testbed

It can also be used to program robot arm paths to be deployed onto the physical systems and assess feasibility of different processes. In the FF testbed, *Process Simulate* was first used to design the lab before the physical infrastructure was put in place. However, it is currently being used to create different robot paths. The capabilities of this tool go beyond that, including real-time DT systems and more custom functions. This highlights the reusable and autonomous characteristics of the testbed. In relation to the autonomous manufacturing use case, this DT technology permitted the initial design of the testbed alongside the formulation of the path that the robot must undertake.

3.3.3 Off-line Programming

DTs are often utilized to program and assess different robot paths before deployment into the physical system. Off-line Programming (OLP) refers to the act of using the DT to program a robot or any other system through the virtual model in the DT. OLP is used to program industrial robot arms in systems containing multiple instances of robot arms that must perform different tasks. This is because hand-training multiple robots would be time-consuming relative to the time OLP requires. This testbed uses *Siemens Tecnomatix Process Simulate* as the OLP tool as well. As such, this testbed can easily introduce new paths and processes for the robot arms. This aids with the reusable characteristic listed above, as different use cases can be implemented with the introduction of new paths for the robots. OLP was used to program the robot with the required path to realize the autonomous manufacturing use case.

3.3.4 PLC Programming

A Programmable Logic Controller (PLC) acts as the brain of the operational equipment in the FF Testbed. A PLC is a controller that has been adapted to manufacturing environments to control manufacturing processes. It can control equipment such as robotic devices, assembly lines, and other machines. A PLC uses ladder logic meaning it continuously monitors the inputs to generate with the required output. FF utilizes the Siemens S7-1516F CPU to run the PLC logic with multiple I/O modules. The PLC program was made through Siemens Totally Integrated Automation (TIA) Portal. This software allows the user to program using ladder logic, functional block diagrams, statement list, and structured control language. The FF testbed uses PLC programming to create the different processes needed. The capability of writing the PLC code from scratch allows this testbed to be reusable and interoperable as the code can be altered to different use cases and sets up communication between the different operational devices. A custom PLC code was written that integrates the sensor data and the robot paths to create the necessary logic to implement the autonomous manufacturing use case.

3.3.5 Virtual Commissioning

Virtual Commissioning (VC) is a means of integrating both PLC Programming and Digital Twins. In VC, a DT is commissioned to run the logic usually deployed on a PLC. This ability allows users to test the PLC code on the virtual environment before the physical counterpart. VC can be accomplished by providing a simulation or emulation of a PLC (Software in the Loop) or coupling a physical PLC with the DT (Hardware in the Loop). In the FF testbed, *Process Simulate* is coupled with a virtualized PLC using *Siemens PLCSim Advanced* software. VC has been used to test the PLC code created before deployment. VC allows the testbed to be reusable by reconfiguring the code and DT design to fit the use case desired with ease. VC was used to test the custom PLC code written for the autonomous manufacturing use case.

3.3.6 Internet of Things (IoT) and Sensors

Internet of Things (IoT) refers to a system of interconnected computing devices with other assets for the purpose of data exchange and processing. While IoT deals mainly with consumer appliances such as smart home devices and prototyping equipment such as Arduino processors, Industrial Internet of Thing (IIoT) refers to assets in manufacturing systems such as robot arms, industrial sensors, and other machines. In this testbed, IIoT was developed through the integration of various sensors on different equipment and the introduction of processing equipment capable of data collection and analysis. Currently, IIoT techniques are utilized in the FF testbed to enable data communication between sensory devices and logic controllers with more processing intensive devices such as the edge device. One of the main components of the autonomous manufacturing use case are the different sensors needed which are integrated into the FF testbed.

3.3.7 Edge Computing

As described in chapter two, Edge Computing is a field of research that introduces processing capabilities closer to the data source rather than relying on cloud computing technologies. This allows computing operations to be performed on generated data in real time. This is usually done with edge devices located on premise, directly communicating with the operational assets. Currently in the FF testbed, Edge Computing is being employed to filter data before being sent to the cloud. This filtering is done to minimize the amount of redundant data being stored. One current use case also includes closed loop control of the assembly process using image processing. This thesis will explain how edge computing is employed to introduce Semantic Web capabilities by mapping the raw data to contextualized information for the fault tolerance use case.

3.3.8 Cloud Computing

Cloud Computing describes the ability to perform processing intensive tasks over the internet ("the cloud"). Generally, this capability requires equipment such as servers to be introduced. Cloud Computing can be used for data storage, data analytics, and complex intelligence. Cloud Computing can be useful in connecting data from multiple physical locations into one centralized server. The testbed at the FF lab has different cloud computing platforms currently being integrated. These platforms include Siemen's *Mindsphere* and IBM's *Maximo Application Suite*. Current use cases for Cloud Computing in the testbed are straightforward dashboarding and data visualization as work is still being done on the connection of data from the PLC up to the different platforms. Cloud Computing will be integrated as the platform for further reasoning of contextualized information from the edge for the autonomous manufacturing use case.

3.4 Semantic Web in the Future Factories Testbed

This section will delve deeper into the fault tolerance use case. It will cover the integration of Semantic Web concepts into the FF testbed and the process taken to implement the use case.

3.4.1 Abstraction of Data

For the use case to be realized, proper abstraction of the manufacturing data must be implemented.

Figure 3.4 describes an example of data abstraction. The Data, Information, Knowledge, Wisdom (DIKW) pyramid has four layers. The lowest layer being the data layer describes the raw data that is acquired by the sensor. The raw data provides no other information but the actual analog value that the sensor generates. The information level, the second layer, connects this sensor data to spatial, temporal, or thematic elements.

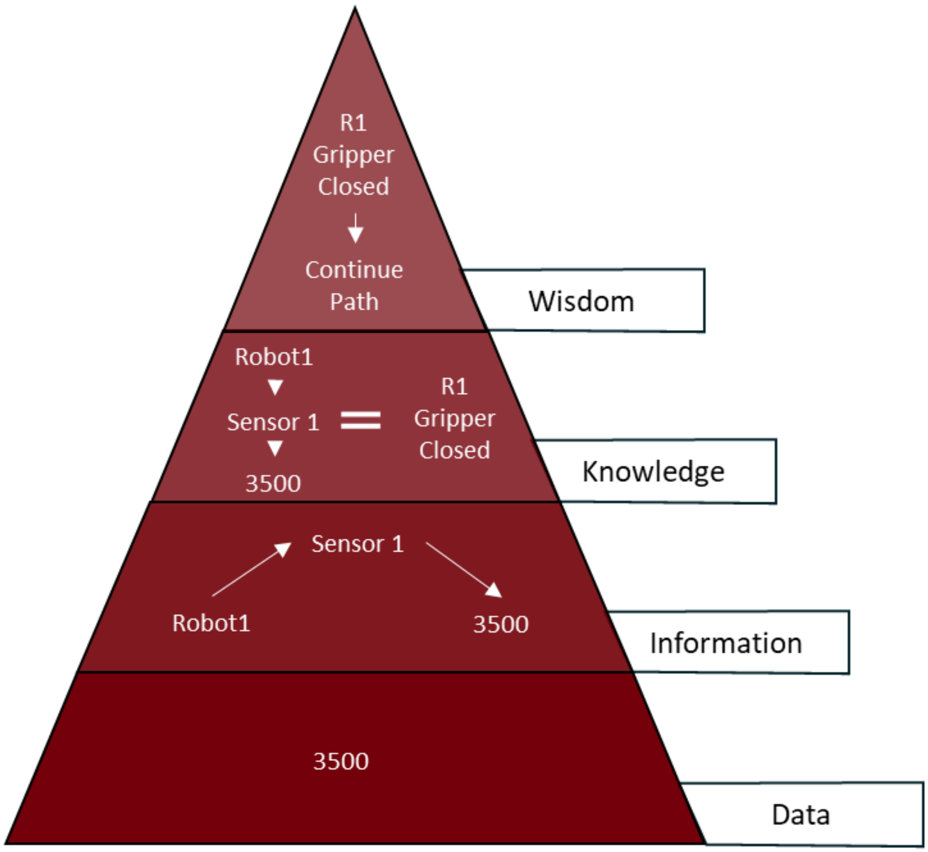

Figure 3.4: Example with different abstraction levels

In this example the information level connects the raw data value to the spatial context within our manufacturing cell by connecting the value to the sensor generating it and robot one, the larger asset that the sensor is connected to. The third layer is the knowledge layer which utilizes the contextualized information to deduce knowledge that cannot be measured directly. In this example, the measured value from the sensor linked to robot one allows us to deduce that the robot gripper is closed. Finally, the fourth layer makes use of the deduced knowledge from the previous layer to come up with decisions regarding the process. In this example, the deduced knowledge reaffirms that the robot can continue its path in the manufacturing process. Even though the above is a very simple example, this methodology can be used at a much more complex level. The aggregation of multiple sensors in similar fashion can allow more complex monitoring of processes or prediction capabilities. This same abstraction process will be implemented to realize the autonomous manufacturing use case.

3.4.2 Development Methodology

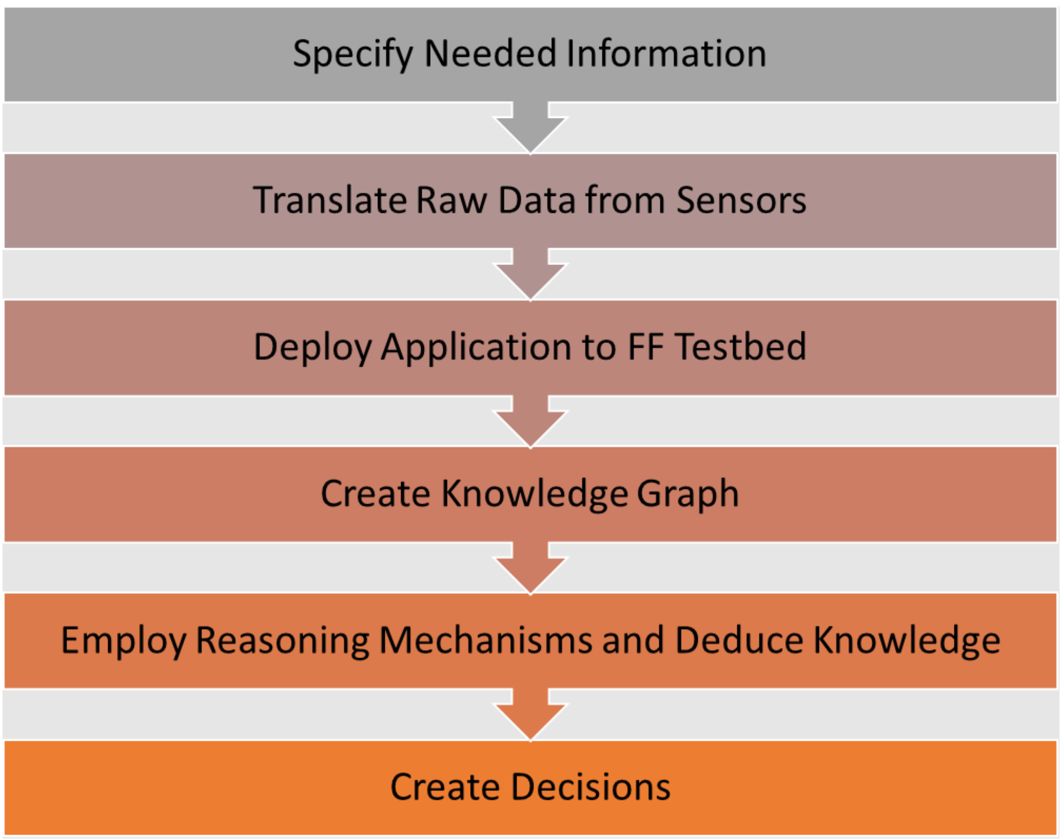

Figure 3.5: Implementation Plan

Figure 3.5 outlines the steps taken to execute the full autonomous manufacturing use case in the FF testbed. The first step details the information that the use case will utilize. The second step focuses on acquiring this information in RDF to be able to manipulate it further using Semantic Web technologies. The data coming from the sensor is in raw format and thus needs to be mapped out accordingly. To do this, an application was developed that takes as an input this raw data and outputs the value mapped out to the correct entities each time this value is updated. Once this application was developed, the next step was to deploy it onto the FF testbed. The tools integrated into the testbed mentioned previously were leveraged to deploy this application. The output of this application was then integrated with the other information in a KG. Reasoning Mechanisms were then deployed on the KG to deduce whether the potentiometer was functional or not. Once this was known, the system can then decide whether to use the potentiometer or the timer to continue with the required path. This provides a high-level structure of the implementation plan adopted throughout this thesis.

3.4.3 Data Modelling

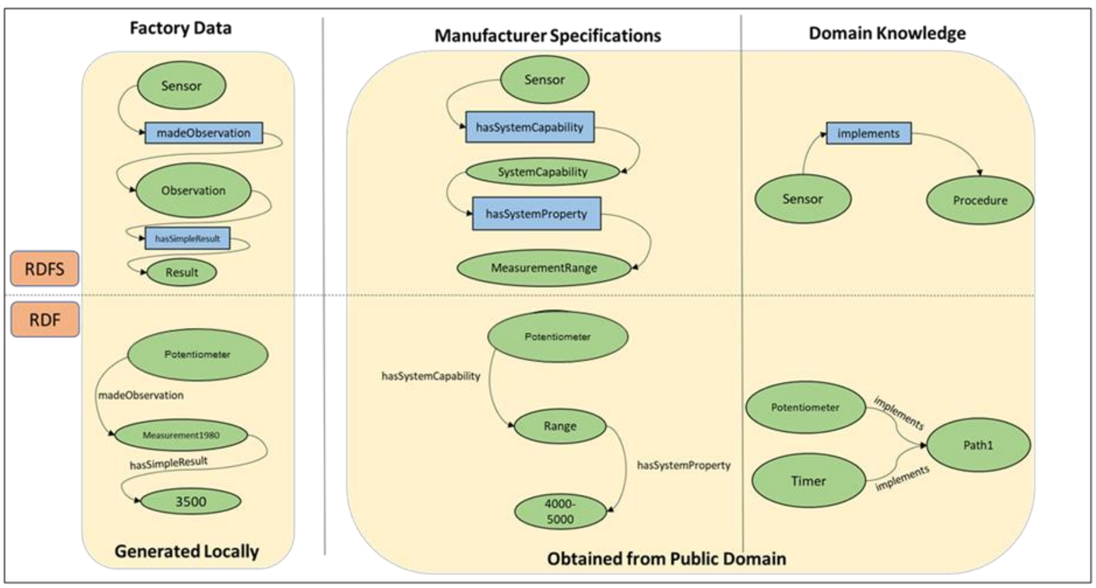

Figure 3.6: Information for the AM Use Case

The first step was to identify the necessary information for the use case to be possible. Figure 3.6 outlines the different information leveraged to produce the final decision. All the information presented is modelled using the Resource Description Framework Schema (RDFS). The top half of the figure represents the ontology adopted which is the Semantic Sensor Network (SSN) ontology.

The first set of information is the local data generated by the sensor in the testbed. This data is mapped out to the potentiometer. This set has three separate entities in the schema which are the Sensor, Observation, and Result entities.

1) *sosa:Sensor*: This entity represents a device that responds to a stimulus and generates a result which relates to the potentiometer which is a sensor that generates different results depending on the change in linear motion of its extrusion.
2) *sosa:Observation:* This entity represents the value of a property which in our case is the measurement of the potentiometer. These two entities are linked together using the *sosa:madeObservation* property.
3) *sosa:Result:* This entity represents the actual value of the observation made which is the value given by the potentiometer which is linked to the Observation using the *sosa:hasSimpleResult* property. The lower half of the figure illustrates the instances created from the described ontology for this use case.

The second data source is the information obtained from the manufacturer. This source provides information about the lifetime of the sensor and the normal output range that the sensor should be yielding. This is required as it provides the information set that is used to deduce the functionality of the potentiometer. The information is mapped as follows:

1) *sosa:Sensor*: Similarly, to the first set, this entity represents the potentiometer again and will be unified into one entity instance in the subsequent KG.

2) *ssn-system:SystemCapability:* This entity represents a property of the *Sensor* entity. In this use case, this represents the Range attribute of the potentiometer. This entity is mapped to the *Sensor* entity through the *ssn-system:hasSystemCapability* property.

3) *ssn-system:MeasurementRange:* This entity is the set of values that a sensor can return which is the normal output range that the manufacturer specified for the Potentiometer. This entity is mapped to the above entity using the *ssn-system:hasSystemProperty* property.

Finally, the third data source is the domain knowledge about the testbed. As described above, a robot arm can use the data from two different sensors to continue with the implemented path. Either the potentiometer or a timer can provide the needed data to deduce whether the gripper is in the required state or not. This set of information is integral as it will be the basis of which the autonomous system will decide which sensor to rely on for the process to move forward. This information is mapped as follows:

1) *sosa:Sensor*: Once again this entity describes the sensors used and will be instanced twice for this source, once for the potentiometer and once for the timer.

2) *sosa*:*Procedure*: This entity describes a workflow, plan, or algorithm that makes a change to the state of the world. In our use case, this will be instanced as Path1 which is the path programmed on the robot to pick up the object. Path1 moves the robot from its home location to the location of the part needed to be picked up. Once the part is picked up which is indicated by the potentiometer value, this path finishes by moving that part to a second predetermined location. This entity is linked to the sensor entity using the *ssn:implements* property.

Chapter 4

Implementation

4.1 Introduction

This chapter will provide a detailed view of all the steps taken as part of the contributions in the Thesis. Section two will go over the development of the FF Testbed and outline all the equipment available and the overall communication set up. Section three will detail the Autonomous Manufacturing use case presented in Chapter three using a Semantic Web approach.

4.2 Future Factories Testbed

To discuss all the assets available at the FF Testbed, this section will be broken down into the three factory layers introduced in Chapter 2.

4.2.1 Equipment Layer

4.2.1.1 Robotic Arms

The FF testbed hosts five Yaskawa six-axis Robots (two HC10s and three GP8s) and are controlled by YRC1000 and YRC1000micro robot controllers. Figure 4.1 shows the Yaskawa GP8's (on the right) and the Yaskawa HC10's (on the left). The two GP8s are primarily used for assembly tasks.

Their high speed and high repeatability accuracy allows them to cooperatively assemble a wide range of items. The three HC10s are utilized to facilitate material intake and output.

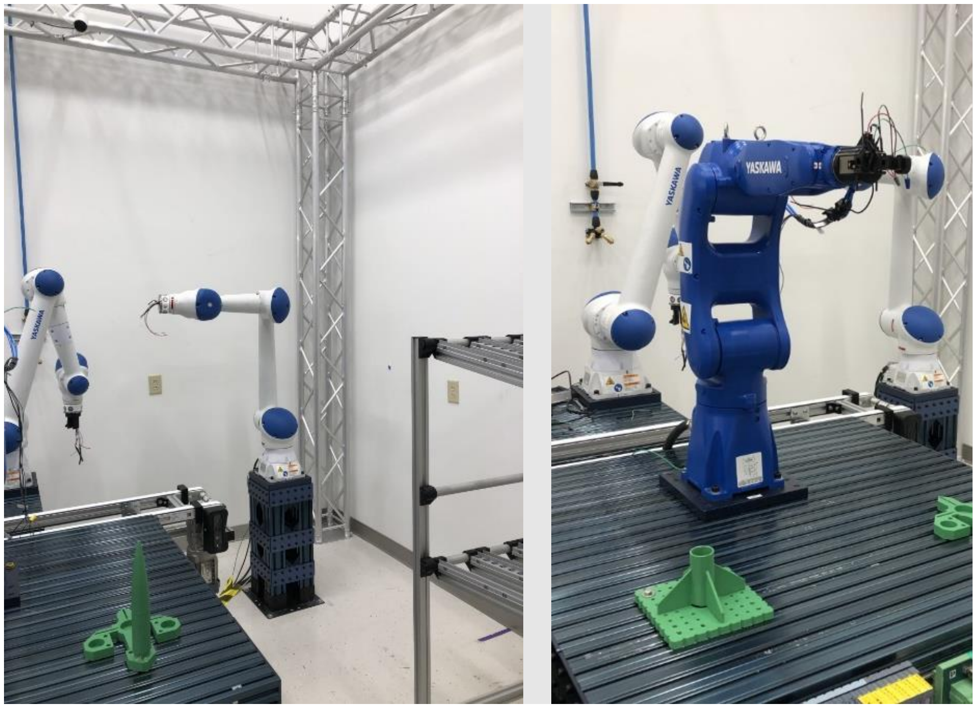

Figure 4.1: Robotic Arms in Future Factories

Additionally, they perform inspection procedures and some collaborative tasks with the other robots. They can either be operated manually with smart pendants or remotely through the PLC system. Custom designed pneumatic end-effector tools are 3D printed in the lab and attached to the end of the robot arms to grip the rocket parts.

4.2.1.2 Conveyor System

The FF testbed also contains a four-conveyor system all connected to achieve full loop around the different robots as seen in Figure 4.2. The C4N Conveyor Belts and Stands are used to transport the manufactured or assembled products throughout the different stations available in the testbed. They are controlled by the Sinamics GS120 Variable Frequency Drives which in turn are connected to the Programmable Logic Controller. The conveyor system is pivotal for allowing collaboration between the different robotic arms available.

Figure 4.2: FF Testbed Setup

4.2.1.3 Programmable Logic Controller

The devices and machines in the lab communicate to each other through a Siemens SIMATIC CPU 1517F-3 PN S7-1500 PLC. PLC code is written in Siemens Totally Integrated Automation (TIA) Portal engineering software, where the layout of devices in the PLC network is configured and managed. The PLC communicates to connected devices using a wide variety of communication protocols, the most notable being the Profinet

Industrial Ethernet standard. The PLC is connected to three ET200SP Distributed I/O devices, which are essentially extended I/O modules with larger numbers of available inputs and outputs, and they are mounted near the controlled device as opposed to being seated directly on the PLC rail. The Distributed I/O modules are mounted by the robot stations and communicate to the PLC network via a Profinet connection. Distributed I/O provides an advantage when communicating to devices such as robots that require a large amount of I/O signal allocation. In the TIA Portal software, PLC data tags have their data type defined, and they are mapped to the robots' concurrent I/O signals. These PLC tags represent the actual logic state of the concurrent I/O signal. By creating a function block - which essentially works like a class from a standard programming language - input, output, and temporary variables can be created and assigned to the PLC tags. By writing PLC code (whether using ladder diagrams, Structured Control Language, or Function Block Diagrams), logic can be performed on the variables to manipulate the logic state of the PLC tags upon triggering of the specified conditions moving the manufacturing process forward.

4.2.1.4 End Effectors

A pneumatic end-effector gripper tool was designed, and 3D printed in the Future Factories lab and attached to the end of each robot. The design of the end-effector allows the attachment of different gripper configurations depending on the shape of the workpiece being manipulated, increasing the flexibility to address a broader range of manufacturing use-cases. The end-effector hosts several sensors - accelerometers, potentiometers, and load sensors, etc. - that report real-time status data to the Siemens Distributed I/O modules mounted on the shop floor.

4.2.1.5 Sensors

Motion Capture Cameras

Future Factories incorporates an OptiTrack motion capture system, Figure 4.3, into the manufacturing cell using an optical motion capture software called Motive. Retroreflective markers are placed at various locations within the cell to be tracked by OptiTrack motion capture cameras that are mounted on the truss surrounding the cell. The marker locations are calibrated in Motive, and once they are mapped within the generated virtual 3D space, their coordinates can be linked within the software to create rigid bodies to represent physical objects within the cell. OptiTrack plugins can be used to transmit real-time motion data into other programs such as Unity or Unreal Engine; in the scope of manufacturing, OptiTrack motion data is imported into Process Simulate for virtual commissioning purposes.

Potentiometer

One of three sensors mounted on the Robot's end effector, the Sensata-BEI potentiometer measures the linear position of the extruding rod shown in Figure 4.4. The main function of the potentiometer is to provide data about the status of the gripper. The value given by the sensor when the gripper is closed differs from the one given when open and thus the value provides knowledge about whether the gripper is open or closed. The resistive sensor outputs a voltage that feeds directly into our PLC I/O modules.

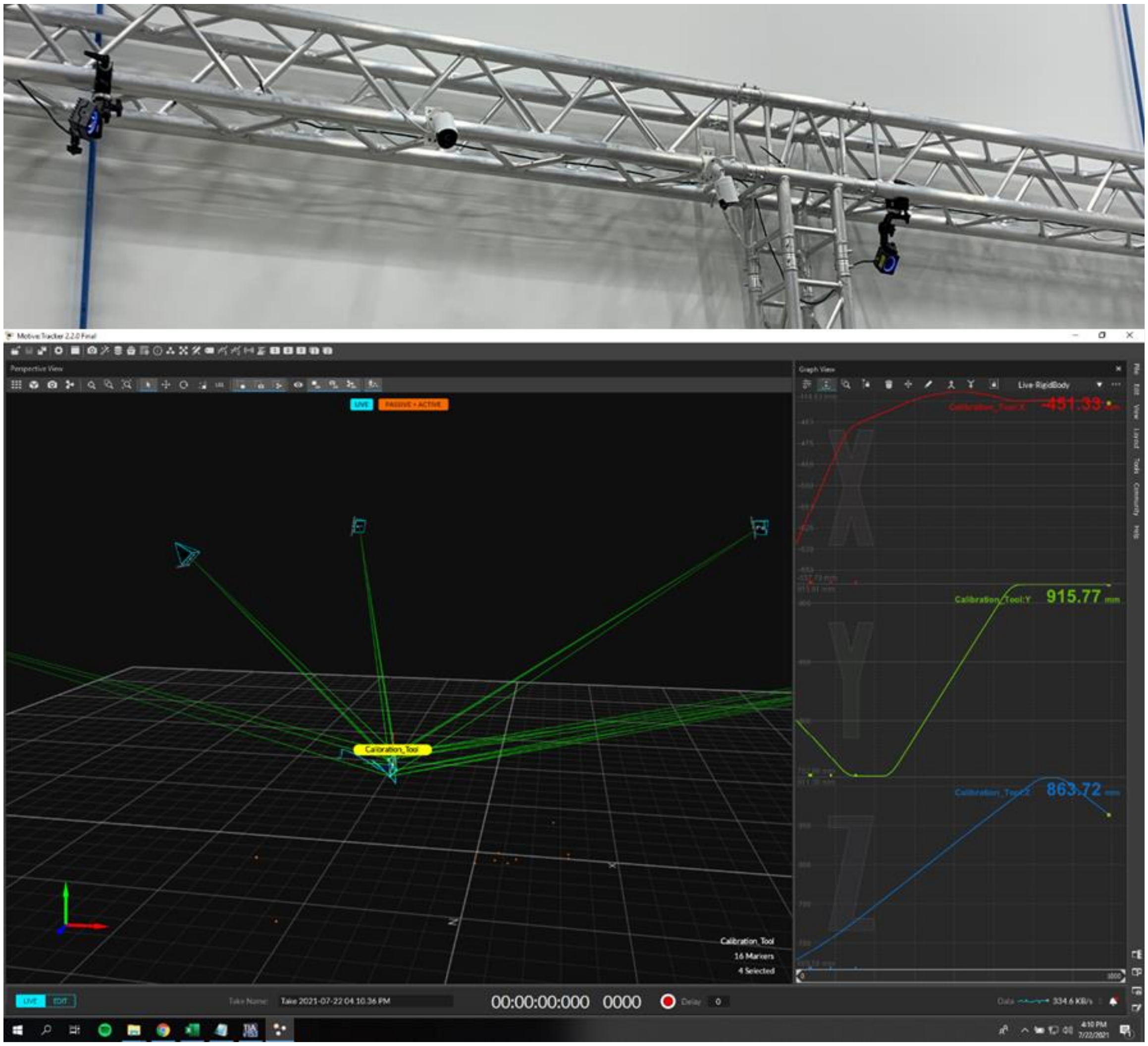

Figure 4.3: Opti Track Motion Capture System

Accelerometer

The second sensor mounted on the end effector is the TE Connectivity accelerometer. The accelerometer measures the vibrations of the gripper. The main function of this sensor is to determine the time interval where the gripper is closing or opening. The opening and closing of the gripper will cause a sudden spike in vibrations and thus when the accelerometer values are minimal, we can deduce that the gripper is in a stable state, and we can continue with the process. The accelerometer is connected directly into the PLC I/O modules as well.

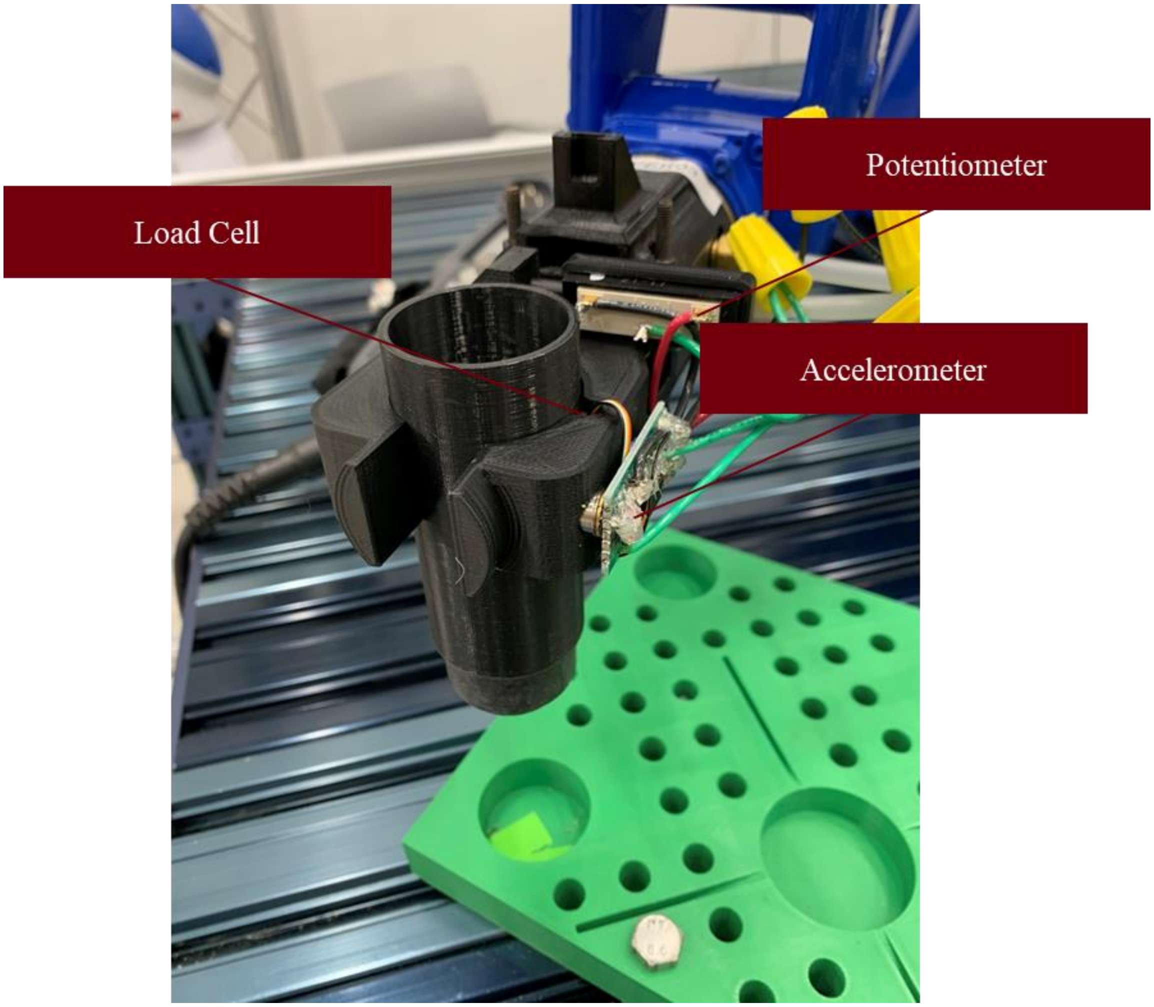

Figure 4.4: End Effector Sensors

Load Cell

The final sensor mounted on the end effector is the TE Connectivity amplified load cell. The load cell measures the force or pressure that is applied to it. The main function of the load cell in our testbed is to determine whether an object is currently being held by the gripper or not. When the gripper is closed and the object is being held, then the load cell value will increase with the increase of pressure. However, when there is no object then the value remains minimal. The load cell is also directly connected to the PLC I/O module.

Figure 4.5. MistLX Vibration Sensors

Inductive Proximity Sensors

As part of the manufacturing processes, multiple stations were created on the conveyors for flexibility of scheduled paths. However, to integrate this station onto the conveyors, inductive proximity sensors had to be placed at every station. These sensors allow us to recognize when an object is at the station and ready to be manipulated with. The proximity sensor is a digital sensor that is either on when an object is detected or off when nothing is detected. These sensors are also directly connected to the PLC I/O modules.

Wireless Vibration Sensors

The MistLX Advanced Wireless Monitoring vibration sensors were installed on the motor drives of every conveyor. These sensors are used to analyze the vibration frequencies of the different conveyors and deduce equipment health indicators. The data is relayed onto a gateway wirelessly and pushed onto the edge device for processing. Both the sensor and gateway can be seen in Figure 4.5.

4.2.2 Shop Floor Layer

4.2.2.1 Simatic IPC227E

The IPC227E (Figure 4.6) is the main edge device used in the Future Factories lab. It includes 240GB SSD storage, 8GB RAM, and an Intel Celeron N2930 processor. The device has two ethernet ports as well. One port is connected to the PLC system through Profinet connection, the other port is connected to the UofSC network. The second connection allows users to access the edge device through the Siemens IPC Management System. This management system is installed on the UofSC network and allows users to access and configure many IPC’s while on the local network. Within the Future Factories lab, the IPC has multiple uses including data collection and visualization, connection with cloud services, and hosts the application developed as part of this thesis to transform raw data to RDF triples.

Figure 4.6. SIMATIC IPC227E (Left), Karbon 700 (Right)

4.2.2.2 KARBON 700

Another edge device present in the FF Lab is the Karbon 700 Rugged Edge Computer (Figure 4.6). This device has 64GB SSD of storage, 4GB RAM and an Intel Celeron G4900T Processor. This edge device is connected to the gateways and mainly functions as a sink for data generated by the wireless sensors before sending them to the cloud platform.

4.2.3 Enterprise Layer

The Enterprise layer in FF is still in the early stages of configuration and set up. Currently, this layer includes two cloud platforms with limited connectivity to the data generated within the testbed.

4.2.3.1 Siemens Mindsphere

One of the cloud platforms utilized is *Siemens Mindsphere*. *Mindsphere* is a Siemens solution created as an IoT platform to seamlessly integrate the generated data from a factory environment. *Mindsphere* has a variety of capabilities built within the software alongside the ability to deploy custom made applications. Within the FF lab, *Mindsphere* is currently being utilized as a data lake alongside simple visualization of metrics.

4.2.3.2 IBM Maximo Application Suite

IBM *Maximo Application Suite* (MAS) is another cloud platform being integrated into the FF lab. This platform is broken up into three different suites; *Maximo Manage, Health, and Monitor*. *Manage* is intended to add assets to the platform which represent the different equipment in the factory floor alongside the topology of their connection. *Health* reads different data generated from the equipment and sensors then outputs different health performance indicators and maintenance prediction. *Monitor* can be utilized to create dashboards and visualizations of all the data connected to the platform. In the FF lab, Maximo is currently being configured and as such is not being fully utilized for any specific use case.

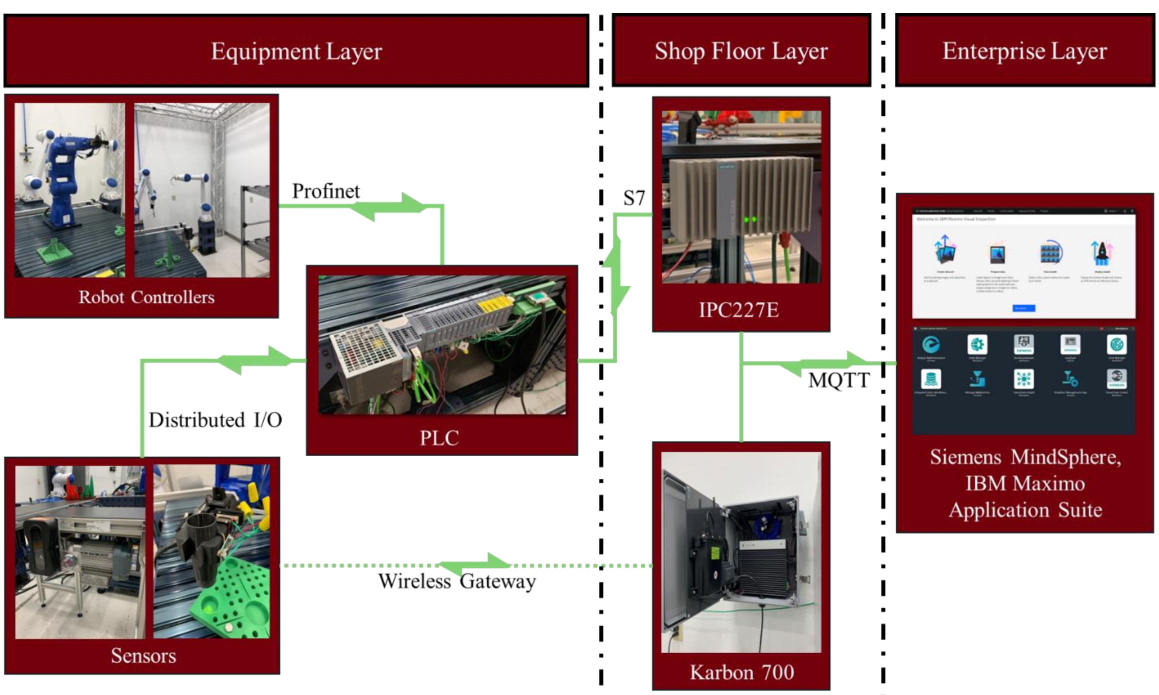

Figure 4.7. Communication Architecture in FF testbed

4.2.4 Communication Architecture

The overall communication architecture of the FF testbed is shown in Figure 4.7. The figure outlines the variety of communication protocols that are implemented ranging from wired Profinet and Distributed I/O connections to wireless Gateway and MQTT communication. These connections allow data to be sent up through the defined layers from the Equipment to the Enterprise layer and vice versa.

4.3 Fault Tolerance Use Case

With the FF testbed fully built the next step was to develop the use case presented previously. Chapter three specified in detail the information that will be used in this use case. With the requirements defined, the raw sensor data generated had to be translated.

4.3.1 Semantic Annotation

This translation process is called semantic annotation. In order to translate the raw data to contextualized information, the generated sensor values had to be connected to existing entities in the KG created for this use case. To do so, one of the Semantic Web technologies describes in Chapter two was used and customized to fit into our case, SDM-RDFizer (Iglesias et al., 2020). SDM-RDFizer is an interpreter that transforms unstructured data into RDF KGs. For this interpreter to function properly, the user must perform some steps. The first step is to define the mapping that the interpreter must abide by when functioning. This mapping must be defined in a Terse RDF Triple Language (TTL) file, a common file format used to express RDF data. Within this file, triples are defined using RDF Mapping Language (RML) (Dimou et al., 2014). RML allows the user to define custom mapping rules for heterogeneous data structure to an RDF data model. The raw data received is sent within a JSON object.

```json
{
  "id": "101",
  "qc": 3,
  "ts": "2021-11-11T17:53:21.487Z",
  "val": 5000
},
```


Figure 4.8. Sensor Data in JSON

Figure 4.8 showcases a snippet of the JSON object that the raw data is sent within. This object has four separate key-value pairs. The first is the unique ID of the sensor which allows us to identify which sensor this value corresponds to. QC refers to the quality code or otherwise known as quality of service (QoS) which determines the status of the message delivery. Ts is the timestamp of the generated value and finally "val" is the actual value that the sensor is generating. The unique ID's are assigned to sensors based on the deployment of the SIMATIC S7 Connector (*SIMATIC S7 Connector Configurator Operating Manual V1.1*, 2021) which provides connectivity for the Edge Device being used to different PLC data tags which are set up for each sensor. When configuring the S7 Connector, an ID is assigned to the PLC data tag based on the order the tags are added starting from 101. In the mapping rules, the only key-value pairs that are of interest for our use case are the ID and value.

RML functions through defining the triples and connecting them to the input data. The first triple defined in this file is the triple connecting the *sosa:Measurement* entity to the *sosa:Result* entity. The Measurement entity is appended with the time stamp of the data then connected to the value from the JSON object through *the sosa:hasSimpleResult* relationship.

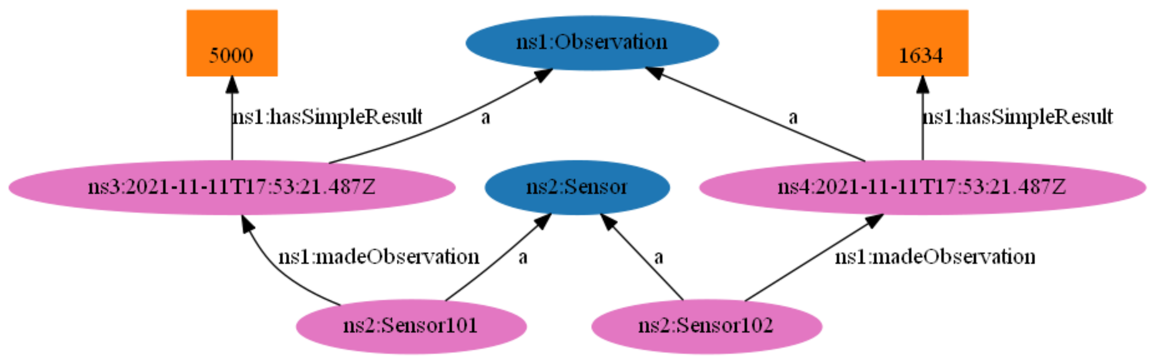

Figure 4.9. Output of Annotated Data

After that triple is defined, the ID of the sensor is instantiated as a *sosa:Sensor* entity and connected to the *sosa:Measurement* entity created in the previous triple through the *sosa:madeObservation* relationship. Figure 4.9 shows the generated KG from the data received for two sensors. This KG has four triple instantiations alongside the entity classes.

4.3.2 Deployment of Application

With the annotation application ready, it now needed to be deployed onto the FF testbed. Referring to the three-layered architecture, the Equipment layer did not represent a good option for such an application since it was more complex than simple logic processing. Therefore, the Shop floor layer was chosen as the location of deployment. More specifically, the IPC227E Edge device described above. The first step in deploying an application onto the edge device requires containerization of the application. This step essentially places all the written functions and their dependencies into one container that can be run independently on any device. To accomplish this, Docker for Windows (Cook, 2017) was used. Running within a container also allows the application to run infinitely which in turn permits the translation of new data each time a new message payload is received.

Once the application is containerized, the deployment process requires uploading the docker image onto the edge device management system through the *Siemens Application Publisher*.

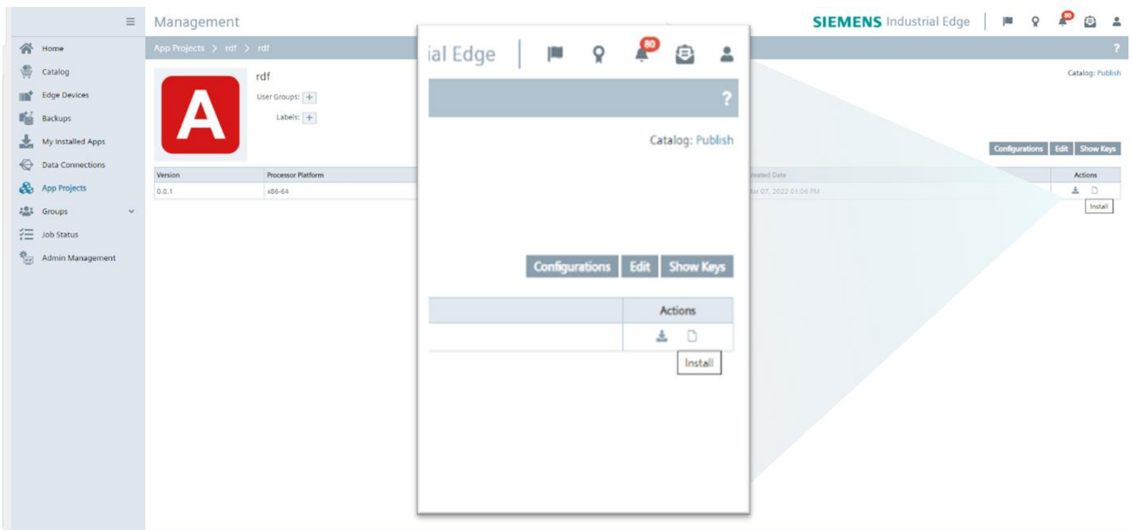

Figure 4.10. View of Application on Management System.

Figure 4.10 shows the application once uploaded to the management system. After that, the application can be installed by pressing the install button on the right side of the screen. Once completely installed, the application architecture is shown in Figure 4.11.

Sensors, conveyors, and robots are all connected to different I/O modules on the PLC with unique data tags. The data associated with each data tag is then sent to the edge device through the S7 Connector. This connector then publishes all this data in the internal Industrial Edge data bus within the edge device through an MQTT connection. At this point, the data is still raw and present in JSON. The deployed translator application subscribes to the topic that the S7 Connector publishes to. Once subscribed, the application translates the data as described previously and publishes the annotated data back onto the data bus through a different topic. This translation happens in near real time as the RDF triples can be seen on the data bus as soon as the raw data is.

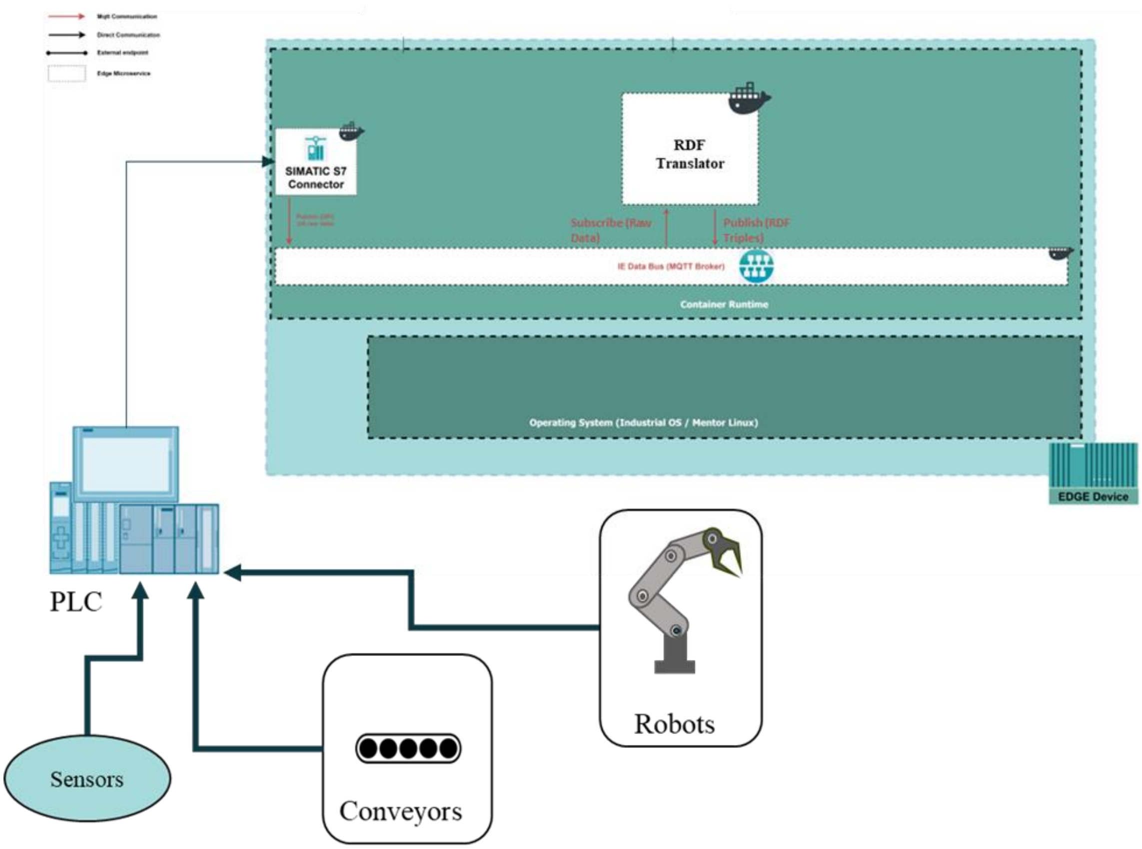

Figure 4.11. Edge Device Application Architecture

4.3.3 KG Generation

With the RDF triples being generated on the edge level, the next step is the creation of the needed KG. This is done on a separate machine to simulate cloud level processing. At this level, the Jena Reasoning Mechanism (Ameen et al., 2014) is utilized to be able to integrate all the different information into one model.

```java
// STEP 1 : LOAD SEMANTIC SENSOR DATA for Jena
Model model = ModelFactory.createDefaultModel();

ReadFile.enrichJenaModelOntologyDataset(model,
        FACTORY_DATA);



// GENERIC APPLICATION
GenericApplication generic_appli = new GenericApplication(model);

// STEP 2: SPECIFIC DOMAIN ONTOLOGIES AND DATASETS

ReadFile.enrichJenaModelOntologyDataset(generic_appli.model,
        MANUF_SPECS);
ReadFile.enrichJenaModelOntologyDataset(generic_appli.model,
        DOMAIN_KNOWLEDGE);
```

Figure 4.12. KG Generation using Jena

Figure 4.12 shows the steps required to generate the KG. Step one creates the model and integrates the generated triples. Step two combines those triples with the domain knowledge and manufacturer specifications present in different files.

```
# if Sensor101 value LESS THAN 4000 then Sensor needs changing
[NeedsChange:
    (sosa:Measurement1980 sosa:hasSimpleResult ?v)
    (ssn-system:Range ssn-system:hasSystemProperty ?r)
    lessThan(?v, ?r)
    ->
    (ssn:Sensor101 sosa:NeedsChange "True")
]

[DoenstNeedsChange:
    (sosa:Measurement1980 sosa:hasSimpleResult ?v)
    (ssn-system:Range ssn-system:hasSystemProperty ?r)
    greaterThan(?v, ?r)
    ->
    (ssn:Sensor101 sosa:NeedsChange "False")
]
```

Figure 4.13. Jena Rules

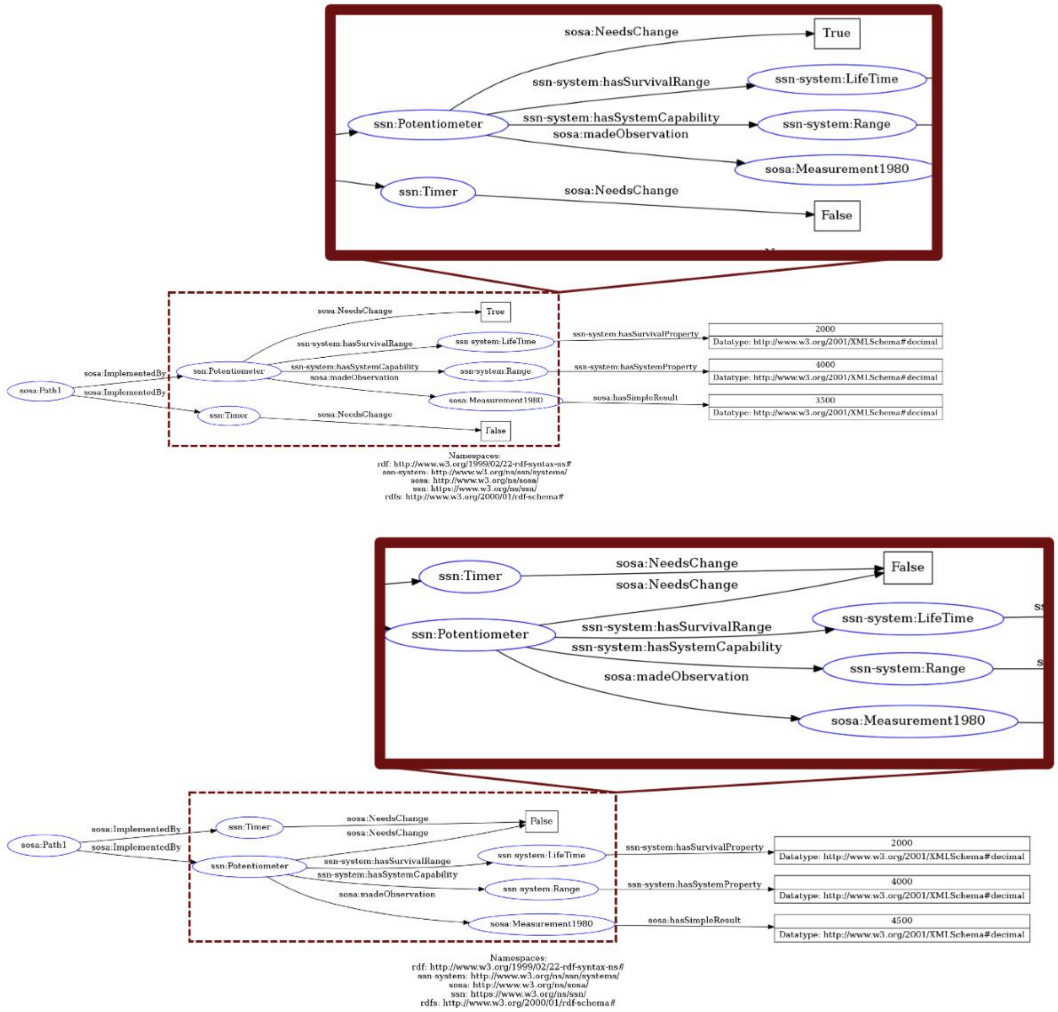

Figure 4.14: Final generated KG with functioning or malfunctioning potentiometer

4.3.4 Knowledge Deduction and Decision Making

With the KG created, reasoning was introduced to deduce knowledge from the information present. This reasoning was introduced in the form of rules that allow the creation of new entities which will be used in the decision-making process. Figure 4.13 shows the two rules that were implemented in this use case to deduce knowledge from the information provided in the KG. Should the generated KG have a Potentiometer value of less than that given in the manufacturer specifications, then the sensor needs changing. The visualized KGs whether the potentiometer is malfunctioning or not can be seen in Figure 4.14.

With the KG manipulation accomplished, the next step was to come up with the final decision regarding the best way to complete the required path. To iterate through the KG SPARQL (shorturl.at/QWZ35) was used which is a query language for RDF. In the query statement in Figure 4.15, the status of the sensor is extracted. If the status returned was that the sensor needs changing, then the different sensors that can be used are discovered. The final projected result of the query statement is the decision made on which sensor value to use shown in Figure 4.16 and Figure 4.17 .

```
SELECT ?Change WHERE{
        ssn:Potentiometer sosa:NeedsChange ?bool.
        sosa:Path1 sosa:ImplementedBy ?Sensors.
        BIND(IF(?bool = "True", ?Sensors  , ?bool) AS ?Change)

}
```

Figure 4.15. SPARQL Query

```
    <ssn-system:hasSurvivalProperty rdf:datatype="http:/
  </rdf:Description>
</rdf:RDF>
No Change  Needed
```

Figure 4.16: Output if Sensor Operational

```
    <ssn-system:hasSurvivalProperty rdf:datatype="http://www.
  </rdf:Description>
</rdf:RDF>
Potentiometer needs changing
Will use [https://www.w3.org/ns/ssn/Timer] instead
```

Figure 4.17: Output with Malfunctional Sensor

4.4 Discussion

In this thesis, the incorporation of Semantic Web technologies was described within a manufacturing context. The use case follows through with the full process of semantic annotation of raw data, knowledge graph generation, and deduction of implicit knowledge. For this thesis, Semantic Web was chosen as it presents a technology capable of centralizing all the information required in one central database. It can also help provide more fault tolerance capabilities beyond sensor malfunction, as different equipment can work dynamically to cover for larger machinery failure by using the information present about machine capabilities, production line layout, and scheduled jobs. This approach also presents different advances to manufacturing in general.

4.4.1 Standardized Data Integration Process

The steps taken within this thesis to integrate the different information sources can be applied within many different domains and use cases. Even though this use case focuses mainly on one certain instance of a malfunctioning sensor, this process can be generalized to encapsulate whichever capability required. This thesis also showcases the applicability of undergoing real time translation of the raw data. This work presents a way forward for incorporating heterogenous data sources from shop floors.

With this procedure outlined alongside the technologies, further work can be undergone to integrate more information and address other issues that may occur on the manufacturing shop floor. This provides a great advantage over traditional exception handling which requires numerous lines of code to cover every instance that may occur. Within this approach, exceptions such as that tackled in this thesis, can be dealt with dynamically rather than coded before deployment.

4.4.2 Interoperability

With more data being generated on manufacturing shop floors, more issues begin to arise when addressing interoperability of that data. In that sense, Semantic Web provides a step towards solving such issues with the emerging standards being adopted for structuring the raw data in an accessible fashion between machines and equipment. These standards include ontologies such as the one used in this thesis, the SSN ontology.

4.4.3 Domain Knowledge

Finally, KGs present a great opportunity for integrating domain knowledge for manufacturing processes. As such, traditional specification mediums such as manuals can be analyzed to include much of the information in there within the KG. This use case focused only on output range but there are endless possibilities of information that can be extracted whether it be operating or set up instruction. Having all this integrated into one central KG can lead to different data accessibility and autonomous manufacturing capabilities.

4.5 Limitations

One limitation with the developed application is the current lack of a user interface. To develop the mapping required in the edge application, the user must have knowledge of

RML coding for the application to output the correct RDF triples. Within the future work, an interface can be created where users can create the mapping through in a graphical drag and drop fashion. This can allow a broader range of users to collaborate in the data integration effort as more assets can be mapped out.

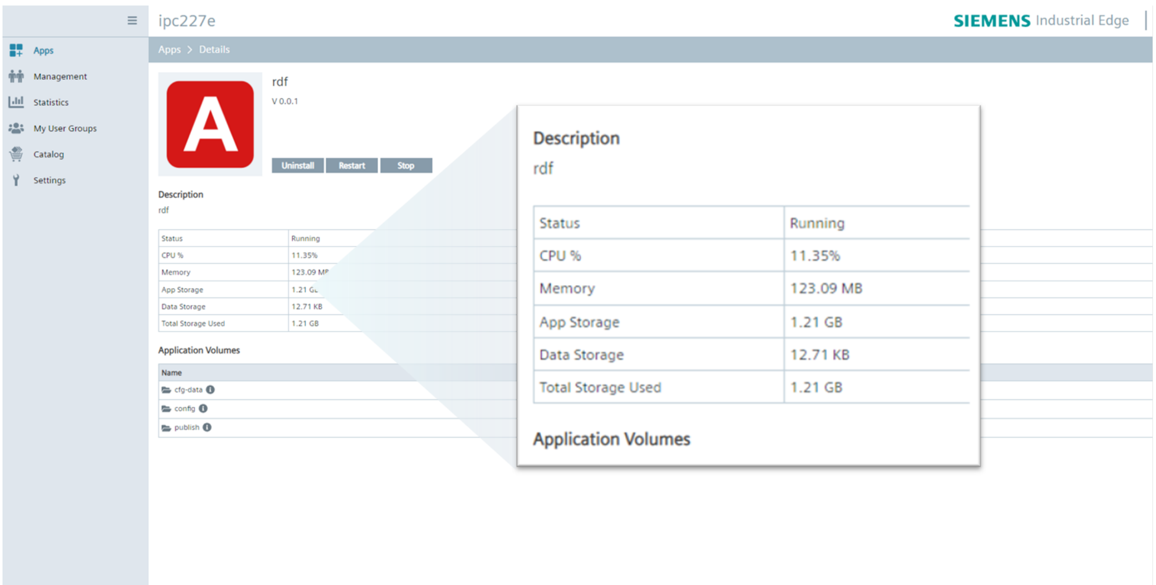

Figure 4.18: Application Installed on Edge

Another limitation this application has is the computational requirement needed to process the data whether in the annotation or reasoning stage. Currently, the application deployed on the edge device only reads two analog values and translates them into the RDF triples outlined in the mapping. Even with the minimal data tags, the application still requires 11% of the CPU and 1.21GB in storage as seen in Figure 4.18. Within manufacturing facilities, hundreds of data tags would have to be processed to create a true Knowledge Graph that encapsulates the full facility. This will cause an issue for resource constraint devices such as the one used in this thesis as there will not be enough processing capabilities to be allocated. For such an integration process to be possible, the application needs to be optimized at first to minimize the needed storage. Multiple edge devices could also be utilized, each focusing on translating a range of data tags and thus distributing the processing throughout the shop floor.

Chapter 5

Conclusion

The need for autonomous manufacturing has become more apparent in the age of Industry 4.0. With an ever-changing market and added focus on customized production, factory floors must be agile and dynamic to adapt to different needs. In that regard, the added capabilities of autonomous manufacturing can allow this new era of manufacturing to be introduced at a greater scale. As discussed earlier, an autonomous manufacturing system can be fault-tolerant to a faulty sensor by preventing a temporary breakdown of a line or factory by suggesting a replacement sensor with similar functionality. It can dynamically allocate resources at runtime (self-organization) and when production demands lead to the introduction of a new machine in factories, it can simply participate by announcing its services and features during the resource allocation process (agile manufacturing). This illustrates the agility of a factory, where a new machine can be integrated into plug-and-produce fashion according to market demands with minimal downtime. All these capabilities can lead to optimized, customized, and faster manufacturing lines in different factory floors.

As a first step towards this reality, this thesis narrowed down the implementation to one specific use case derived from the above characteristics. Fault tolerance was a capability that was not tackled thoroughly with only a couple of previous works addressing it from the full number of works gathered. In addition to that, from the works cited, none utilized Semantic Web to showcase it. As such, fault tolerance was identified as the necessary contribution. To be able to achieve that goal, this thesis aimed at finding the intersection between the three research fields discussed in chapter two by utilizing Smart Manufacturing and Semantic Web techniques to tackle the identified gap in autonomous manufacturing.

The adopted use case was a specific capability that can be applied to the FF testbed. The FF testbed has processes and paths that depend on certain sensor readings to continue. In our case, the robot requires sensor readings from a potentiometer to know whether a gripper is closed and if it can continue its scheduled path. However, to be fault-tolerant, the system should be able to continue with the process when the potentiometer malfunctions. This brought up two main requirements. The first requirement deals with the physical infrastructure needed to implement the use case on. To satisfy this requirement, work was accomplished that integrates different manufacturing and processing equipment together to achieve a full testbed. These devices have open communication channels and equipment with the processing ability to handle the required data. This testbed has advanced capabilities and equipment as outlined in this thesis. The second requirement is from the Semantic Web aspect. In this requirement, the work integrates the needed data. It shows how locally generated data can be semantically annotated then integrated to create a KG. This KG is then used to come up with decisions for the manufacturing process.

This thesis provides a foundation for different research directions moving forward. From one aspect, the developed testbed is a state-of-the-art environment for testing and developing new technologies that could benefit the manufacturing sector. With the available infrastructure, future research directions could lean towards different AI use cases from image recognition and inspection, predictive maintenance and modelling, to complex event understanding. The possibilities are endless as new equipment can be easily integrated and platforms configured.

Within Semantic Web, future work should focus on further integration into manufacturing settings. This thesis contributes with the real time annotation of raw data. However, more complex processing such as reasoning is done outside the limits of the testbed due to the lack of processing power of equipment at the time. The next step in this use case involves integrating the full Semantic Web capabilities within the equipment in the testbed allowing the knowledge to truly be infused into the process as the robots are operational. When that step is achieved, then different capabilities are needed to realize autonomous manufacturing beyond fault tolerance can be attained.

Following from that, more complex systems can be created that aim at creating machine event understanding. This means that all equipment can deduce knowledge about different events happening around them and reacting accordingly. On a broader aspect, this work can be a springboard to further developing Semantics in manufacturing including building a novel standardized smart manufacturing KG that can be utilized within different factories.

Finally, the contributions laid out within this Thesis provide an intersection between the different research done in the participating research teams, the Artificial Intelligence Institute, and the Future Factories team at the University of South Carolina. Infusing semantics with manufacturing and pushing the research forward in both teams.

5.1 Situation Research

This study in knowledge infusion into manufacturing processes represents an overall goal of the research undertaken at the Future Factories laboratory at the University of South Carolina's McNair Center. These works attempt to recognize mechanical features (Harik et al., 2017) for manufacturability analysis (Y. Shi et al., 2018) and cover topics such as semantic segmentation (Xia, Saidy, et al., 2021b) of images for enhanced event understanding. Previous works also integrate reinforcement learning with digital twins for increased intelligence (Xia, Sacco, et al., 2021b), multi-modal robotic health through IoT (Saidy, 2021), and virtual commissioning (Xia et al., 2019b). All these works aim at increasing autonomy of manufacturing processes similar to this thesis by introducing Artificial Intelligence in manufacturing through different approaches. On top of that, this work was built upon the early design work done to create the Future Factories testbed (Saidy et al., 2020)

References

Agrawal, A., Won, S. J., Sharma, T., Deshpande, M., & McComb, C. (2021). A Multi-Agent Reinforcement Learning Framework for Intelligent Manufacturing with Autonomous Mobile Robots. *Proceedings of the Design Society*, *1*, 161–170. https://doi.org/10.1017/pds.2021.17

Alam, Md. F., Shtein, M., Barton, K., & Hoelzle, D. J. (2020, September 3). Autonomous Manufacturing Using Machine Learning: A Computational Case Study With a Limited Manufacturing Budget. *Volume 2: Manufacturing Processes; Manufacturing Systems; Nano/Micro/Meso Manufacturing; Quality and Reliability*. https://doi.org/10.1115/MSEC2020-8472

Alsafi, Y., & Vyatkin, V. (2010). Ontology-based reconfiguration agent for intelligent mechatronic systems in flexible manufacturing. *Robotics and Computer-Integrated Manufacturing*, *26*(4), 381–391. https://doi.org/10.1016/J.RCIM.2009.12.001

Aly, M. F., Abbas, A. T., & Megahed, S. M. (2010). Robot workspace estimation and base placement optimisation techniques for the conversion of conventional work cells into autonomous flexible manufacturing systems. *International Journal of Computer Integrated Manufacturing*, *23*(12), 1133–1148. https://doi.org/10.1080/0951192X.2010.528033

Ameen, A., Ur Rahman Khan, K., & Rani, Bp. (2014). Reasoning in Semantic Web Using Jena. *Online)*, *5*(4). www.iiste.org

B. Scholz-Reiter, st. Sowade, D. Rippel, M. Teucke, M. Özsahin, & T. Hildebrandt. (2009). A Contribution to the Application of Autonomous Control in Manufacturing. *International Journal Of Computers*, *3*(3).

Balakirsky, S. (2015). Ontology based action planning and verification for agile manufacturing. *Robotics and Computer-Integrated Manufacturing*, *33*, 21–28. https://doi.org/10.1016/J.RCIM.2014.08.011

Banf, M., & Steinhagen, G. (2022). Who supervises the supervisor? Model monitoring in production using deep feature embeddings with applications to workpiece inspection. *ArXiv*, *abs/2201.06599*.

Botts, M., Percivall, G., Reed, C., & Davidson, J. (2008). OGC® sensor web enablement: Overview and high level architecture. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *4540 LNCS*, 175–190. https://doi.org/10.1007/978-3-540-79996-2_10/COVER/

Bourne, D. A., & Mark S. Fox. (1984). Autonomous manufacturing: automating the job-shop. *Computer*, *17*(09), 7686.

Cai, M., Zhang, W. Y., Chen, G., Zhang, K., & Li, S. T. (2009). SWMRD: a Semantic Web-based manufacturing resource discovery system for cross-enterprise collaboration. *Http://Dx.Doi.Org/10.1080/00207540902814330*, *48*(12), 3445–3460. https://doi.org/10.1080/00207540902814330

Cao, H., Wachowicz, M., & Cha, S. (2017). Developing an edge computing platform for real-time descriptive analytics. *2017 IEEE International Conference on Big Data (Big Data)*, 4546–4554. https://doi.org/10.1109/BigData.2017.8258497

Cao, H., Yang, X., & Deng, R. (2020). Ontology-Based Holonic Event-Driven Architecture for Autonomous Networked Manufacturing Systems. *IEEE Transactions on Automation Science and Engineering*, *18*(1), 205–215. https://doi.org/10.1109/TASE.2020.3025784

Cardoso, J., & Sheth, A. (2006). The Semantic Web and Its Applications. *Semantic Web Services, Processes and Applications*, 3–33. https://doi.org/10.1007/978-0-387-34685-4_1

Carvalho, C. P. de, & Bittencourt, P. de M. (2021). Industry 4.0 Machine Learning to Monitor the Life Span of Cutting Tools in an Automotive Production Line. *International Journal of Advanced Engineering Research and Science*, *8*(5), 220–228. https://doi.org/10.22161/ijaers.85.25

Carvalho, G., Cabral, B., Pereira, V., & Bernardino, J. (2021). Edge computing: current trends, research challenges and future directions. *Computing*, *103*(5), 993–1023. https://doi.org/10.1007/s00607-020-00896-5

Compton, M., Barnaghi, P., Bermudez, L., García-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., Huang, V., Janowicz, K., Kelsey, W. D., le Phuoc, D., Lefort, L., Leggieri, M., Neuhaus, H., Nikolov, A., Page, K., … Taylor, K. (2012). The SSN ontology of the W3C semantic sensor network incubator group. *Journal of Web Semantics*, *17*, 25–32. https://doi.org/10.1016/J.WEBSEM.2012.05.003

Cook, J. (2017). Docker. *Docker for Data Science*, 29–47. https://doi.org/10.1007/978-1-4842-3012-1_2

Das, R., Munkhdalai, T., Yuan, X., Trischler, A., & McCallum, A. (2018). Building Dynamic Knowledge Graphs from Text using Machine Reading Comprehension. *7th International Conference on Learning Representations, ICLR 2019*. https://doi.org/10.48550/arxiv.1810.05682

David Rand. (2021, April 7). *Moving from Automated to Autonomous Manufacturing*.

Dimou, A., Sande, M. Vander, Colpaert, P., Verborgh, R., Mannens, E., & Van De Walle, R. (2014). *RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data*. http://www.gac-grid.de/project-products/Software/XML2RDF.html

Dinar, M., & Rosen, D. W. (2017). A design for additive manufacturing ontology. *Journal of Computing and Information Science in Engineering*, *17*(2). https://doi.org/10.1115/1.4035787/446471

Ding, K., Lei, J., Chan, F. T. S., Hui, J., Zhang, F., & Wang, Y. (2020). Hidden Markov model-based autonomous manufacturing task orchestration in smart shop floors. *Robotics and Computer-Integrated Manufacturing*, *61*, 101845. https://doi.org/10.1016/j.rcim.2019.101845

Ding, K., Lei, J., Zhang, F., Wang, Y., & Wang, C. (2020). Analyzing the cyber-physical system–based autonomous collaborations among smart manufacturing resources in a smart shop floor. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, *234*(3), 489–500. https://doi.org/10.1177/0954405419875340

Gaur, M., Faldu, K., & Sheth, A. (2021). Semantics of the Black-Box: Can Knowledge Graphs Help Make Deep Learning Systems More Interpretable and Explainable? *IEEE Internet Computing*, *25*(1), 51–59. https://doi.org/10.1109/MIC.2020.3031769

Goldsmith, S., & Interrante, L. (1998). An Autonomous Manufacturing Collective for Job Scheduling. *Proceedings of the Artificial Intelligence and Manufacturing Workshop*.

Grangel-González, I., Lösch, F., & Ul Mehdi, A. (2020). Knowledge Graphs for Efficient Integration and Access of Manufacturing Data. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, *2020-September*, 93–100. https://doi.org/10.1109/ETFA46521.2020.9212156

Grundstein, S., Freitag, M., & Scholz-Reiter, B. (2017). A new method for autonomous control of complex job shops – Integrating order release, sequencing and capacity control to meet due dates. *Journal of Manufacturing Systems*, *42*, 11–28. https://doi.org/10.1016/j.jmsy.2016.10.006

Haarslev, V., Hidde, K., Möller, R., & Wessel, M. (2011). *The RacerPro Knowledge Representation and Reasoning System 1*. *1*, 1–5.

Harik, R., Shi, Y., & Baek, S. (2017). Shape Terra: mechanical feature recognition based on a persistent heat signature. *Computer-Aided Design and Applications*, *14*(2), 206–218. https://doi.org/10.1080/16864360.2016.1223433

Henson, C. A., Pschorr, J. K., Sheth, A. P., & Thirunarayan, K. (2009). SemSOS: Semantic sensor observation service. *2009 International Symposium on Collaborative Technologies and Systems, CTS 2009*, 44–53. https://doi.org/10.1109/CTS.2009.5067461

Hildebrandt, A.-C., Schuetz, C., Wahrmann, D., Wittmann, R., & Rixen, D. (2016). A Flexible Robotic Framework for Autonomous Manufacturing Processes: Report from the European Robotics Challenge Stage 1. *2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 21–27. https://doi.org/10.1109/ICARSC.2016.15

Hu, L., Liu, Z., Hu, W., Wang, Y., Tan, J., & Wu, F. (2020). Petri-net-based dynamic scheduling of flexible manufacturing system via deep reinforcement learning with graph convolutional network. *Journal of Manufacturing Systems*, *55*, 1–14. https://doi.org/10.1016/j.jmsy.2020.02.004

Iglesias, E., Jozashoori, S., Chaves-Fraga, D., Collarana, D., & Vidal, M.-E. (2020). SDM-RDFizer. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 3039–3046. https://doi.org/10.1145/3340531.3412881

Ivan Sutherland. (1963). *Sketchpad, a man-machine graphical communication system*.

Iwamura, K., & Sugimura, N. (2010). A study on real-time scheduling for autonomous distributed manufacturing systems. *2010 IEEE International Conference on Systems, Man and Cybernetics*, 1352–1357. https://doi.org/10.1109/ICSMC.2010.5642451

Jain, R. (2008). EventWeb: Developing a human-centered computing system. *Computer*, *41*(2), 42–50. https://doi.org/10.1109/MC.2008.49

Jarvis, D., Jarvis, J., Lucas, A., & Ronnquist, R. (2001). A multi-agent systems approach to collaborative autonomous manufacturing operations. *2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat.No.01CH37236)*, 152–157. https://doi.org/10.1109/ICSMC.2001.969804

Jarvis, D., Jarvis, J., McFarlane, D., Lucas, A., & Ronnquist, R. (2001). Implementing a multi-agent systems approach to collaborative autonomous manufacturing operations. *2001 IEEE Aerospace Conference Proceedings (Cat. No.01TH8542)*, 2803–2811. https://doi.org/10.1109/AERO.2001.931300

Kagermann, H., Wahlster, W., & Helbig, J. (2013). *Recommendations for implementing the strategic initiative INDUSTRIE 4.0*.

Khan, W. Z., Ahmed, E., Hakak, S., Yaqoob, I., & Ahmed, A. (2019). Edge computing: A survey. *Future Generation Computer Systems*, *97*, 219–235. https://doi.org/10.1016/j.future.2019.02.050

Kho, G., Hung, C., & Cunningham, H. (2014). Robo brain: Massive knowledge base for robots. *Cornell Univ., USA, Tech. Rep* .

Kulvatunyou, B. S., Cho, H., & Son, Y. J. (2005). A semantic web service framework to support intelligent distributed manufacturing. *International Journal of Knowledge-Based and Intelligent Engineering Systems*, *9*(2), 107–127. https://doi.org/10.3233/KES-2005-9206

Kusiak, A., & Verma, A. (2012). Analyzing bearing faults in wind turbines: A data-mining approach. *Renewable Energy*, *48*, 110–116. https://doi.org/10.1016/j.renene.2012.04.020

Leitner, S.-H., & Mahnke, W. (2006). *OPC UA-Service-oriented Architecture for Industrial Applications*.

Le-Phuoc, D., Dao-Tran, M., Xavier Parreira, J., & Hauswirth, M. (2011). *A Native and Adaptive Approach for Unified Processing of Linked Streams and Linked Data* (pp. 370–388). https://doi.org/10.1007/978-3-642-25073-6_24

Lin, H. K., & Harding, J. A. (2007). A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration. *Computers in Industry*, *58*(5), 428–437. https://doi.org/10.1016/J.COMPIND.2006.09.015

Liu, C., Su, X., & Li, C. (2021). Edge Computing for Data Anomaly Detection of Multi-Sensors in Underground Mining. *Electronics*, *10*(3), 302. https://doi.org/10.3390/electronics10030302

Liu, C., Vengayil, H., Lu, Y., & Xu, X. (2019). A Cyber-Physical Machine Tools Platform using OPC UA and MTConnect. *Journal of Manufacturing Systems*, *51*, 61–74. https://doi.org/10.1016/j.jmsy.2019.04.006

Lu, Y., Wang, H., & Xu, X. (2019). ManuService ontology: a product data model for service-oriented business interactions in a cloud manufacturing environment. *Journal of Intelligent Manufacturing*, *30*(1), 317–334. https://doi.org/10.1007/S10845-016-1250-X/FIGURES/15

Ma, S., Guo, W., Song, R., & Liu, Y. (2021). Unsupervised learning based coordinated multi-task allocation for unmanned surface vehicles. *Neurocomputing*, *420*, 227–245. https://doi.org/10.1016/j.neucom.2020.09.031

Markellou, P., Mousourouli, I., Spiros, S., & Tsakalidis, A. (2005). Using semantic web mining technologies for personalized e-learning experiences. *Proceedings of the Web-Based Education*.

Michael Grieves. (2015). *Digital Twin: Manufacturing Excellence through Virtual Factory Replication*.

Moergestel, L. van, Puik, E., Telgen, D., & Meyer, J.-J. (2011). Decentralized Autonomous-Agent-Based Infrastructure for Agile Multiparallel Manufacturing.

2011 Tenth International Symposium on Autonomous Decentralized Systems, 281–288. https://doi.org/10.1109/ISADS.2011.38

Padhee, S., Lalithsena, S., & Sheth, A. (2018). Creating Real-Time Dynamic Knowledge Graphs. *Kno.e.Sis Publications*. https://corescholar.libraries.wright.edu/knoesis/1143

Papananias, M., McLeay, T. E., Obajemu, O., Mahfouf, M., & Kadirkamanathan, V. (2020). Inspection by exception: A new machine learning-based approach for multistage manufacturing. *Applied Soft Computing*, *97*. https://doi.org/10.1016/j.asoc.2020.106787

Park, H.-S., & Tran, N.-H. (2011). An autonomous manufacturing system for adapting to disturbances. *The International Journal of Advanced Manufacturing Technology*, *56*(9–12), 1159–1165. https://doi.org/10.1007/s00170-011-3229-2

Parsons, S. (2009). A Semantic Web Primer, Second Edition by Antoniou Grigoris and Harmelen Frank van, MIT Press, 288 pp., $42.00. *The Knowledge Engineering Review*, *24*(4), 415–415. https://doi.org/10.1017/S0269888909990117

Pellegrinelli, S., Orlandini, A., Pedrocchi, N., Umbrico, A., & Tolio, T. (2017). Motion planning and scheduling for human and industrial-robot collaboration. *CIRP Annals*, *66*(1), 1–4. https://doi.org/10.1016/j.cirp.2017.04.095

Pujara, J., & Getoor, L. (n.d.). *Building Dynamic Knowledge Graphs*. Retrieved July 7, 2022, from https://github.com/linqs/psl/tree/online_psl

Qi, Q., Tao, F., Hu, T., Anwer, N., Liu, A., Wei, Y., Wang, L., & Nee, A. Y. C. (2021). Enabling technologies and tools for digital twin. *Journal of Manufacturing Systems*, *58*, 3–21. https://doi.org/10.1016/j.jmsy.2019.10.001

Qiu, J., Wu, Q., Ding, G., Xu, Y., & Feng, S. (2016). A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*, *2016*(1), 67. https://doi.org/10.1186/s13634-016-0355-x

Rosen, R., von Wichert, G., Lo, G., & Bettenhausen, K. D. (2015). About The Importance of Autonomy and Digital Twins for the Future of Manufacturing. *IFAC-PapersOnLine*, *48*(3), 567–572. https://doi.org/10.1016/j.ifacol.2015.06.141

Sacco, C., Radwan, A. B., Beatty, T., & Harik, R. (2019, April 11). Machine Learning Based AFP Inspection: A Tool for Characterization and Integration. *SAMPE 2019 - Charlotte, NC*. https://doi.org/10.33599/nasampe/s.19.1594

Saidy, C. (2021). *Multimodal Robotic Health in Future Factories Through IIot, Data Analytics, and Virtual Commissioning*. University of South Carolina.

Saidy, C., Xia, K., Sacco, C., Kirkpatrick, M., Kircaliali, A., Nguyen, L., & Harik, R. (2020). Building Future Factories: A Smart Robotic Assembly Platform Using Virtual Commissioning, Data Analytics, and Accelerated Computing. *SAMPE 2020 Virtual Series | Digital Modeling Technologies in Manufacturing*.

Schneible, J., & Lu, A. (2017). Anomaly detection on the edge. *MILCOM 2017 - 2017 IEEE Military Communications Conference (MILCOM)*, 678–682. https://doi.org/10.1109/MILCOM.2017.8170817

Schuster, A., Kupke, M., & Larsen, L. (2017). Autonomous Manufacturing of Composite Parts by a Multi-Robot System. *Procedia Manufacturing*, *11*, 249–255. https://doi.org/10.1016/j.promfg.2017.07.238

Sharp, M., Ak, R., & Hedberg, T. (2018). A survey of the advancing use and development of machine learning in smart manufacturing. *Journal of Manufacturing Systems*, *48*, 170–179. https://doi.org/10.1016/j.jmsy.2018.02.004

Sheth, A., Henson, C., & Sahoo, S. S. (2008). Semantic sensor web. *IEEE Internet Computing*, *12*(4), 78–83. https://doi.org/10.1109/MIC.2008.87

Sheth, A., Padhee, S., & Gyrard, A. (2019). Knowledge Graphs and Knowledge Networks: The Story in Brief. *IEEE Internet Computing*, *23*(4), 67–75. https://doi.org/10.1109/MIC.2019.2928449

Sheth, A., & Perry, M. (2008). Traveling the Semantic Web through Space, Theme and Time. *IEEE Internet Computing*, *12*(2), 81. https://doi.org/10.1109/MIC.2008.46

Sheth, A., & Ramakrishnan, C. (2003). Semantic (Web) Technology in Action: Ontology Driven Information Systems for Search, Integration, and Analysis. *IEEE Data Engineering Bulletin*, *26*(4). https://corescholar.libraries.wright.edu/knoesis/970

Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*, *3*(5), 637–646. https://doi.org/10.1109/JIOT.2016.2579198

Shi, Y., Zhang, Y., Baek, S., de Backer, W., & Harik, R. (2018). Manufacturability analysis for additive manufacturing using a novel feature recognition technique. *Computer-Aided Design and Applications*, *15*(6), 941–952. https://doi.org/10.1080/16864360.2018.1462574

SIMATIC S7 Connector Configurator Operating Manual V1.1. (2021).

Sirin, E., Parsia, B., Cuenca Grau, B., Kalyanpur, A., & Katz, Y. (n.d.). *Pellet: A Practical OWL-DL Reasoner*. Retrieved July 7, 2022, from http://www.mindswap.org/2003/pellet/download.shtml

Sirojan, T., Lu, S., Phung, B. T., & Ambikairajah, E. (2019). Embedded Edge Computing for Real-time Smart Meter Data Analytics. *2019 International Conference on Smart Energy Systems and Technologies (SEST)*, 1–5. https://doi.org/10.1109/SEST.2019.8849012

Su, X., Riekki, J., & Haverinen, J. (2012). Entity Notation: enabling knowledge representations for resource-constrained sensors. *Personal and Ubiquitous Computing*, *16*(7), 819–834. https://doi.org/10.1007/s00779-011-0453-6

Su, X., Riekki, J., Nurminen, J. K., Nieminen, J., & Koskimies, M. (2015). Adding semantics to internet of things. *Concurrency and Computation: Practice and Experience*, *27*(8), 1844–1860. https://doi.org/10.1002/cpe.3203

Su, X., Zhang, H., Riekki, J., Keränen, A., Nurminen, J. K., & Du, L. (2014). Connecting IoT Sensors to Knowledge-based Systems by Transforming SenML to RDF. *Procedia Computer Science*, *32*, 215–222. https://doi.org/10.1016/j.procs.2014.05.417

Tao, F., Cheng, J., Qi, Q., Zhang, M., Zhang, H., & Sui, F. (2017). Digital twin-driven product design, manufacturing and service with big data. *The International Journal of Advanced Manufacturing Technology 2017 94:9*, *94*(9), 3563–3576. https://doi.org/10.1007/S00170-017-0233-1

Teern, A., Kelanti, M., Päivärinta, T., & Karaila, M. (2022). *Knowledge graph construction and maintenance process: Design challenges for industrial maintenance support*. http://ceur-ws.org/Vol-3223/paper1.pdf

Tharumarajah, A. (1998). A SELF-ORGANIZING MODEL FOR SCHEDULING DISTRIBUTED AUTONOMOUS MANUFACTURING SYSTEMS. *Cybernetics and Systems*, *29*(5), 461–480. https://doi.org/10.1080/019697298125588

Thoben, K. D., Wiesner, S. A., & Wuest, T. (2017). "Industrie 4.0" and smart manufacturing-a review of research issues and application examples. *International Journal of Automation Technology*, *11*(1), 4–16. https://doi.org/10.20965/IJAT.2017.P0004

Thuluva, A. S., Dorofeev, K., Wenger, M., Anicic, D., & Rudolph, S. (2017). *Semantic-Based Approach for Low-Effort Engineering of Automation Systems* (pp. 497–512). https://doi.org/10.1007/978-3-319-69459-7_33

Tsarkov, D., & Horrocks, I. (2006). FaCT++ description logic reasoner: System description. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *4130 LNAI*, 292–297. https://doi.org/10.1007/11814771_26/COVER/

Wada, H., & Okada, S. (2002). An autonomous agent approach for manufacturing execution control systems. *Integrated Computer-Aided Engineering*, *9*(3), 251–262. https://doi.org/10.3233/ICA-2002-9305

Wan, J., Chen, B., Wang, S., Xia, M., Li, D., & Liu, C. (2018). Fog Computing for Energy-Aware Load Balancing and Scheduling in Smart Factory. *IEEE Transactions on Industrial Informatics*, *14*(10), 4548–4556. https://doi.org/10.1109/TII.2018.2818932

Wang, Y., Hong, K., Zou, J., Peng, T., & Yang, H. (2020). A CNN-Based Visual Sorting System With Cloud-Edge Computing for Flexible Manufacturing Systems. *IEEE*

Transactions on Industrial Informatics, *16*(7), 4726–4735. https://doi.org/10.1109/TII.2019.2947539

Wheeless, S., & Rahman, S. M. M. (2021). *Sensor Fusion-Based Supervised Learning Approach to Developing Collaborative Manipulation System with Variable Autonomy* (pp. 24–30). https://doi.org/10.1007/978-3-030-68017-6_4

Xia, K., Sacco, C., Kirkpatrick, M., Harik, R., & Bayoumi, A.-M. (2019a, April 11). Virtual Comissioning of Manufacturing System Intelligent Control. *SAMPE 2019 - Charlotte, NC*. https://doi.org/10.33599/nasampe/s.19.1403

Xia, K., Sacco, C., Kirkpatrick, M., Harik, R., & Bayoumi, A.-M. (2019b, April 11). Virtual Comissioning of Manufacturing System Intelligent Control. *SAMPE 2019 - Charlotte, NC*. https://doi.org/10.33599/nasampe/s.19.1403

Xia, K., Sacco, C., Kirkpatrick, M., Saidy, C., Nguyen, L., Kircaliali, A., & Harik, R. (2021a). A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces and intelligence. *Journal of Manufacturing Systems*, *58*, 210–230. https://doi.org/10.1016/j.jmsy.2020.06.012

Xia, K., Sacco, C., Kirkpatrick, M., Saidy, C., Nguyen, L., Kircaliali, A., & Harik, R. (2021b). A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces and intelligence. *Journal of Manufacturing Systems*, *58*, 210–230. https://doi.org/10.1016/J.JMSY.2020.06.012

Xia, K., Saidy, C., Kirkpatrick, M., Anumbe, N., Sheth, A., & Harik, R. (2021a). Towards Semantic Integration of Machine Vision Systems to Aid Manufacturing Event Understanding. *Sensors*, *21*(13), 4276. https://doi.org/10.3390/s21134276

Xia, K., Saidy, C., Kirkpatrick, M., Anumbe, N., Sheth, A., & Harik, R. (2021b). Towards Semantic Integration of Machine Vision Systems to Aid Manufacturing Event Understanding. *Sensors 2021, Vol. 21, Page 4276*, *21*(13), 4276. https://doi.org/10.3390/S21134276

Xu, J., Kim, S., Song, M., Jeong, M., Kim, D., Kang, J., Rousseau, J. F., Li, X., Xu, W., Torvik, V. I., Bu, Y., Chen, C., Ebeid, I. A., Li, D., & Ding, Y. (2020). Building a PubMed knowledge graph. *Scientific Data 2020 7:1*, *7*(1), 1–15. https://doi.org/10.1038/s41597-020-0543-2

Yahya, M., Breslin, J. G., & Ali, M. I. (2021). Semantic Web and Knowledge Graphs for Industry 4.0. *Applied Sciences*, *11*(11), 5110. https://doi.org/10.3390/app11115110

Yang, Z., Gay, R., Miao, C., Zhang, J. B., Shen, Z., Zhuang, L., & Lee, H. M. (2005). Automating integration of manufacturing systems and services: A Semantic Web Services approach. *IECON Proceedings (Industrial Electronics Conference)*, *2005*, 2255–2260. https://doi.org/10.1109/IECON.2005.1569254

Zhang, C., & Ji, W. (2020). Edge Computing Enabled Production Anomalies Detection and Energy-Efficient Production Decision Approach for Discrete Manufacturing Workshops. *IEEE Access*, *8*, 158197–158207. https://doi.org/10.1109/ACCESS.2020.3020136

Zhang, C., Xu, W., Liu, J., Liu, Z., Zhou, Z., & Pham, D. T. (2019). A Reconfigurable Modeling Approach for Digital Twin-based Manufacturing System. *Procedia CIRP*, *83*, 118–125. https://doi.org/10.1016/j.procir.2019.03.141