Developing A Communication Architecture for Improving Production Efficiency in Smart Manufacturing

By

GARETH ANDREW GERICKE

Thesis submitted in fulfilment of the requirements for the degree

DOCTOR OF ENGINEERING: ENGINEERING: ELECTRICAL

in the

Department of Electrical, Electronic and Computer Engineering

of the

Faculty of Engineering, Built Environment and Information Technology

at the

Central University of Technology, Free State

Promoter: Prof. RB Kuriakose (DEng)

Co-promoter: Prof. HJ Vermaak (PhD)

Bloemfontein

2025

DECLARATION OF INDEPENDENT WORK

I, GARETH ANDREW GERICKE, identity number ______________ and student number __________, do hereby declare that this research project submitted to the Central University of Technology, Free State, for the Degree DOCTOR OF ENGINEERING: ELECTRICAL ENGINEERING, is my own independent work, which complies with the Code of Academic Integrity, as well as other relevant policies, procedures, rules and regulations of the Central University of Technology, Free State, and has not been submitted before to any institution by myself of any other person in fulfilment (or partial fulfilment) of the requirements for the attainment of any qualification.

______________________________ 12-09-2025

SIGNATURE OF STUDENT DATE

Acknowledgements

This thesis would not be possible without the extraordinary support from the close people in my life. Whilst I wish to thank them all for the support and kindness they have shown me along my journey, I know it is a small flicker of light compared to the flood of sunshine they have shown me throughout. I would like them to take solace in the fact that their actions have shaped and moulded me into the person I have become. I hope to continuously return the favour, but I would also like to exemplify their example onto others, knowing they were the catalyst that started it all.

Firstly, to God, without your grace and guidance, none of this before us would be possible. There are times that I look back at the sand where you walked beside me and I see only one set of footprints. I know that you carried me during those times.

To my amazing supervisors, Professor Rangith Baby Kuriakose and Professor Herman Jacobus Vermaak, you never stopped believing me and always pushed me to achieve more than I thought was possible. There are many kind words you have told me throughout our time together. I remember all of them and reflect on them daily. I credit my ability in this achievement to what you were able to pass down and teach me. I look forward to what the future may hold for us together.

I would like to thank the Central University of Technology, Free State (CUT), which has seen me throughout my degrees and achievements. Your support and patience gave me the chance to stand on the shoulders of giants, along with international institutions, and represent all that we stand for. I hope I have shown the dedication and grace in these moments that you strive to instil in all your students that walk through your gates.

To my parents, Kevin Michael Gericke (deceased) and Stacey Gericke. For the days you spent with me, creating memories, laughter and a safe place I could always call home, I am eternally grateful. This achievement is as much yours as it is mine. I strive every day to make you proud and become the person you wished. I hope to live the life you invested in for me. I hope to repay your kindness for the rest of my life.

Finally, to my beautiful fiancé, Janique Jade Dean. Your kind and gentle actions became the bedrock I found myself in. You always lifted my spirits and motivations. Your intelligent mind always pushed me to think of things differently and added monumentally to my studies. I look forward to building the rest of our lives together on our foundations with me as your Husband and you as my Wife.

Thank you.

Abstract

Smart manufacturing units have become the latest manufacturing standard within Industry 4.0 for production floor requirements that enable functionality and data structures for the next-generation manufacturing scene. Accompanying these requirements, functionality and structures, there has been a vast sea of research to define requirements within smart manufacturing. These requirements can be satisfied with higher-order structures and lower-level implementations sharing strategies to naturally allow for the flow of requirements and data. The omission of consistency between higher-level and lower-level implementations often leads to lacklustre implementation of smart manufacturing setups, resulting in production inefficiencies and bottlenecks. These issues come as a direct contradiction of the solutions proposed for resolution in smart manufacturing. A literature review of Industry 4.0 and smart manufacturing reveals that there are similarities and complementing features for communication flows throughout all levels of a manufacturing setup. This study classifies these requirements between different levels using a communication architecture. This architecture type looks at the flow of information, organisation of data and a set of rules for responsibilities between levels. These additions allow for control and responsibility of data at each level, allowing for consistency and traceability throughout a manufacturing setup. This study outlines objectives to benchmark a current manufacturing setup in Simulink, identify its production efficiency and other metrics outlined for improvement. Additional objectives for creating a communication architecture, scoring the implementation of this new architecture, comparing the production metrics of the communication architecture against the benchmark and outlining the lifecycle of this architecture follow. Other authors approach the problem of consistency and production improvements with historically altered architecture from software and hardware domains; however, this study evaluates the use of a communication architecture to suggest the selection of this architecture in its applicable scenarios. The communication architecture implementation is tested on the benchmark Simulink model where the results discern meaningful cause-and-effect improvements with traits such as data organisation and responsibilities as opposed to coding timing improvements. The results are discussed to highlight how the communication architecture naturally allows for its requirements to leverage the possibility to include intelligence within smart manufacturing units.

Table of Contents

List of Figures

List of Tables

List of Formulas

List of Abbreviations

BASE	Biography, Attributes, Schedule and Execution
CA	Communication architecture
DDD	Domain Driven Design
DMP	Digital message platform
DT	Digital Twin
FIFO	First in first out
HTN	Hierarchical task network
I4.0	Industry 4.0
IC	Integrated Circuits
IoS	Internet of Services
IoT	Internet of Things
IoT-O	Internet of Things ontology
IP	Internet Protocol
MAPE-K	Monitor Analyze Plan Execute and Knowledge
OPC UA	Open Platform Communication unified Architecture
OSI	Open Systems Interconnection
PLC	Programming Logic Circuits
SM	Smart Manufacturing
SMP	Smart Manufacturing Protocol
SMU	Smart Manufacturing Unit
SP	Smart Product
TCP/IP	Transmission Control Protocol/Internet Protocol
WS02	Web Services oxygenated version 2

Chapter 1 : Introduction

1.1 Background

Hardware, software and system architectures [1] are used within industry to standardise and accurately recreate software, network and physical system setups. These setups aid developers to create manufacturing lines with best practices and optimal functioning of tools for varying levels of integration, monitoring, reporting and production tasks.

The rapid technological growth in Industry 4.0 (I4.0) [2] has resulted in many new strands of technology channels being developed in parallel. This multi-strand research approach has led to many improved practices, at the cost of scrutinised input requirements for best compatibility of developer choice to use on their own setups. One such requirement being the cohesion between physical machines and the flow of communication between them to improve the development and use of smart manufacturing.

With communication protocols used throughout I4.0 [3] being designed around a variety of factors for smart manufacturing, such as network size, latency, overhead and transmission rate, among others, the efficient use of these protocols can be out of reach in implementation due to inadequate setups and ineffective protocol practices in industry.

This often leads to manufacturing setups [4] recalculating unnecessary data flows, increasing overheads, latency and storage space whilst simultaneously reducing the quality of intelligence in a manufacturing system. These problems can be simultaneously eliminated from a system by aligning with a new means of standardisation of communication architectures, allowing for an overall improvement in production efficiency. This study, in the first order, looks at communication architectures which are currently used within smart manufacturing and analyses the communication architectures used and their implementation.

In doing so, the shortcomings of the existing architectures are outlined [5] with respect to their ability to be incorporated for communication between smart manufacturing units within I4.0. The study then proceeds to propose a new communication architecture base which addresses the key concerns that were identified.

This study concludes by making a comparison of the proposed communication architecture base with the ones previously examined. The experiments for the study are conducted on a smart water bottling plant with the hypothesis that the newly-developed communication architecture will improve production efficiency and allow for intelligent communication between machines to be facilitated.

1.2 Problem Statement

Researchers in this niche area have highlighted the lack of development of architecture design of smart manufacturing units which has resulted in a lack of structural support for I4.0 devices. Current literature also highlights that existing communication protocols facilitate information traffic between all points in a smart manufacturing line but fail to channel information into the higher levels of the smart manufacturing line.

This has led to vital monitoring information being filtered out from machine level to system and human verification level, resulting in poor cohesion between system levels. This ultimately results in reduced production efficiency and bottlenecks that are cumbersome to remove in other design areas, i.e. hardware and software.

1.3 Research Hypothesis and Objectives

1.3.1 Research Question

How can the flow of communication within smart manufacturing be organised and interpreted by machines to improve production efficiency as a system at large and avoid bottlenecks between machines?

1.3.2 Research Hypothesis

Facilitation of information through a smart manufacturing setup using the developed communication architecture will allow for intelligent decision-making to be inherent across all machines of the network. These intelligent decisions can be leveraged to monitor, calculate and respond to manufacturing changes and adapt operations for improvement in production efficacy.

1.3.3 Research Aim

This study aims to develop a communication architecture focused on exchange of information between smart manufacturing units to improve production efficiency and reduce bottlenecks by conducting the flow of machine communication to be in-line with I4.0 requirements.

1.3.4 Research Objectives

This study will make use of a pre-existing water bottling plant case study to develop and test a communication architecture for smart manufacturing units and achieve the following objectives:

1. Identify communication architecture development challenges by using the case study of the water bottling plant.
2. Benchmark the current manufacturing case study to evaluate its performance with different manufacturing scenarios.
3. Develop a new communication architecture using the case study of the water bottling plant.
4. Test the developed communication architecture.
5. Evaluate the developed communication architecture throughout the design, capabilities and results with an architecture scoring method applicable to communication metrics.

1.4 Research Methodology

The effective use of architectures in the manufacturing space has proven to be an efficient measure to model and design manufacturing lines. This study aims to develop a communication architecture to introduce smart manufacturing capabilities through the control of smart manufacturing characteristics throughout the manufacturing setup. The study will make use of a water bottle case study, with machines tasked for different requirements of the manufacturing line, to create a benchmark of its current performance and improved performance with the communication architecture.

In this study, traditional hardware and software architectures will be studied as a baseline to determine meaningful requirements and considerations, from which the communication architecture can be developed with similar methods and logical implementation. These requirements and consideration of the communication architecture will have a specific communication focus within the manufacturing system. However, the communication architecture will be developed from adaptive methods and reasoning which are included from a smart manufacturing perspective and highlight important considerations for future iterations and research of the communication architecture itself.

Commonalties between smart manufacturing and communication flow/organisation will be evaluated to determine core components of the architecture. These core components will then be evaluated to see how they are best modelled and related, deriving the remaining aspects of the communication architecture.

The water bottling plant will then be modelled through MATLAB Simulink, providing simulated, real-time data to test and characterise the water bottling plant in a robust manner. The communication architecture will then be incorporated into the water bottling plant and evaluated for effectiveness in production efficiency and bottleneck avoidance.

1.5 Original Contribution of this Study

The packaging, organisation and computation of information in a smart manufacturing environment culminates in a communication architecture and creates a void for much supporting information to validate the communication architecture itself. Therefore, this study recognises its various original contributions, which are:

- The communication architecture for smart manufacturing units
    - The core components of the communication architecture
    - The relationships of the communication architecture
- The responsibilities of each communication layer in a smart manufacturing environment
- A scoring method for a communication architecture
- A scalable Simulink model for smart manufacturing detailing the implementation of the communication architecture

The focus and objective of this study, however, is to develop the communication architecture. The communication architecture is designed in such a way as to allow for future iterations and understandings to be drawn from this implementation for important considerations and attributes. The original contribution is also hoped to be used as guide to develop future architectures in unique and emerging fields.

1.6 Layout Of Thesis

Chapter 1: Introduction to Industry 4.0

This chapter introduces the study, providing insights into the background of past, current and future manufacturing practices. This is followed with stating the problem statement and setting the boundaries of observations and operations for the study by proposing the research hypothesis, aim and objectives.

Chapter 2: Literature Review

This section of the study dwells upon the extensive background and applicable theory of Industry 4.0 and smart manufacturing, hardware, software and systems architectures and communication protocols. The literature review is conducted to showcase important aspects currently being used within the manufacturing space and to highlight important applications and gaps in the research field that are used to develop a communication architecture to facilitate and standardise communication flow within a smart manufacturing network.

Chapter 3: Methodology

The methodology section of the study introduces the case study used to create, validate and expand the considerations in smart manufacturing units, such as their attached sensors, smart products, IoT, digital twins, and so forth, providing longevity within the architecture. Within the methodology section, metrics of the manufacturing scenarios will also be explored to characterise the baseline improvement of the manufacturing setup and the improved performance of the communication architecture. This section will also include the foundation of a communication architecture scoring method, to help guide and critique iterations of the communication architecture, comparisons with alternative improvements, best practices and areas for improvement.

Chapter 4: Results

This chapter reviews the obtained results in both a quantitative manner to show improvements with production efficiency, and a qualitative manner to show natural co-operations between smart manufacturing and communication architectures. This is done to bring about a symbiotic relationship through which to achieve manufacturing goals and improve production goals by reviewing the entire manufacturing life-cycle chain.

Chapter 5: Discussion

This discussion chapter reviews the results gathered from the implemented communication architecture and discusses the advantages of incorporating the architecture to solve manufacturing issues and thereby highlight the original contribution of this study.

This chapter also highlights the considerations taken to create and implement the communication architecture in aspects such as the machine inputs, machine outputs and machine methods in order to highlight the methodologies used to envision the architecture and act as a blueprint for continuous innovation and further research in and around the communication architecture to bring a greater longevity to this fourth level of architectures.

A comparison between existing methods for introducing smart manufacturing systems with architectures is conducted to show the advantages of the developed communication architecture and the novel contribution from the study.

A description of the interactions between the communication architecture core components rounds of the chapter with compliments from the results gather in the experiments for clarity and ease of explanation of the data flows.

Chapter 6: Conclusion

This chapter summarises the study, highlighting the important contributions that a communication architecture can make to manufacturing scenarios and showcasing the results gathered from the implementation of the case study. This chapter also explores areas of future work within and considering communication architectures, whilst reiterating the co-operational advantages of communication architectures in smart manufacturing to allow future developments with similar methods practiced in this study.

Chapter 2 : Overview of Smart Manufacturing and Communication Architectures – Literature Review

2.1 Introduction

A literature review of the integral components relevant to this study is needed to showcase its significance. This literature review explains the technologies used in this study to portray the significance and influence of I4.0, smart manufacturing, smart manufacturing units, communication in smart manufacturing unit networks, architecture parts, architecture model types, architecture implementation areas and the limitations of current research.

The literature review also attempts to indicate relational balance between each part for future iterations of this study to consider the impact, both negative and positive on all parts to gauge effective implementation of improvements and whether certain characteristics inevitably become redundant and replaced.

2.2 Industry 4.0

Industry 4.0 [6] is the latest transformative manufacturing approach to achieve greater autonomy and flexibility in the production environment. The manufacturing process is made up of scheduling, supply chain logistics, production, verification, delivery and feedback to achieve autonomy, automation, integrated systems, feedback of big data and simulated systems, and forecast of trend requirements.

To achieve this set of requirements and integrate all connected parts, I4.0 is realised through four main pillars of technology groups [7], consisting of Internet of Things, Internet of Services, cyber-physical systems and smart manufacturing. These technology groups are seen to be distinctive, showing unique similarities between technologies under each umbrella, with enough similarity around its edges to allow for interaction between these groups. These interactions are also further refined by the input and output requirements of each unique technology type.

2.2.1 Internet of Things

Internet of Things (IoT) [8] technologies are concerned with sensors and/or devices being connected and communicating through networks, often wirelessly. IoT is responsible for the transportation of information where fully implemented systems would not suffice due to either space or cost requirements.

IoT therefore adopts the principle of a more dispersed network to collect information [9] and relay it to a central location for processing or feedback control. The addition of internet-capable devices has seen a gradual but large-scale adoption within the I3.0 space. The refinement of the practice to transition away from purely data mediation devices has seen a form of control and intelligence being integrated into the practice of IoT, firmly breaking itself off from I3.0 and creating its own unique criteria for implementation in I4.0.

2.2.2 Internet of Services

Internet of Services (IoS) [10] follows a similar pattern to that of IoT and has set itself apart in I4.0 by offering automatic communication and information retrieval from software applications through website interfaces. The unique distinction for true classification of IoS in I4.0 comes with co-operable services where, instead of users being restricted with purely data interfaces, their access to data can be used to invoke actions and procedures with automatic methods and verifications that can be duplicated at a high level across internet services.

This is achieved by opting to allow users strategic access to not only the software interface itself, but the development tools, server platforms, storage and communication channels to allow for their specific invocations. The access that is provided should be scaled with analytics on common user requests and specifications, to automatically feed into the life-cycle of the internet service.

2.2.3 Cyber-Physical Systems

Cyber-physical systems [11] are the pillar of I4.0 and include systems relying on both physical machines and remote cyber replicants to fully connect or perform a required task. The encompassing technology grouping of cyber-physical systems includes digital twins and cloud computing where, distinctly different tasks are often created, stored and scaled on networks and serves to aid, control or analyse production data within I4.0 space.

Cyber-physical systems also have the unique advantage of consolidating multiple regions, factories and machines into single setups as, often, entire systems do not need to be replicated, only their desired inputs, methods and outputs. Cyber-physical systems similarly offer some of the best intuitive inclusion points for human operators by allowing machine operators to have safe control of physical setups through remote access, allowing machine diagnostics at locations with the greatest amount of data, enabling a holistic view of all system effects.

2.2.4 Smart Manufacturing

Smart manufacturing (SM) [12] is the inclusion of technology groups mainly focused on manufacturing lines and production facilities. With manufacturing becoming an integral part of global supply chains, the high value input and modernisation of manufacturing needs are always at the forefront for innovation.

SM introduces greater manufacturing capabilities to production lines by offering greater advancements within manufacturing by introducing a level of intelligence within the manufacturing process. This intelligence is realised through the implementation of smart manufacturing units (SMUs) that are reconfigurable, flexible and intelligent in their operations to offer more robustness within manufacturing whilst also improving production efficiency with their intelligent use of resources, reduction in pollution [13], greater connection within the total logistics chain and/or improving production timing/operations.

2.2.5 Smart Manufacturing Units

SMUs [14] are defined as the collection of systems that can process decisions for robust, flexible and intelligent decisions in manufacturing. SMUs are rudimentary implemented as an increase in hardware resources from I3.0 programming logic circuits (PLCs), sensors and machines to be classified as I4.0 capable devices. However, only through effective use of resources and connected information produced from machines are said machines able to communicate effective information. This information-sharing is one enabling step to intrinsically include intelligence into machine operations and allow for smart decisions. Whereas this view can be extracted from the differences of I3.0 to I4.0 machines, distinct differences also exist with SMUs and I4.0 requirements and capabilities that are, in part, realised and enabled by hardware resources.

Although an end manufacturing goal could create obscure and manifold criteria for the classification of intelligent control and, to a larger extent, SMUs, evaluation of a machine based on its ability to enact certain levels of control and communication creates a more well-established definition. Therefore, SMUs are rather classified by their compliance to a set of characteristics.

2.2.6 Smart Manufacturing Characteristics

SMUs inherit their abilities by the unique characteristics [15] that they possess from the requirements of technologies within I4.0. These characteristics consist of Contextual Awareness, Sustainability, Cloud Manufacturing, Agility, Interoperability and Data Analytics. These characteristics are further realised by enabling technologies that, once implemented into a machine, should allow for intelligent operations to be carried out and meet the minimum requirements of an SMU in I4.0. A summary of the smart manufacturing characteristics can be seen in Figure 2-1.

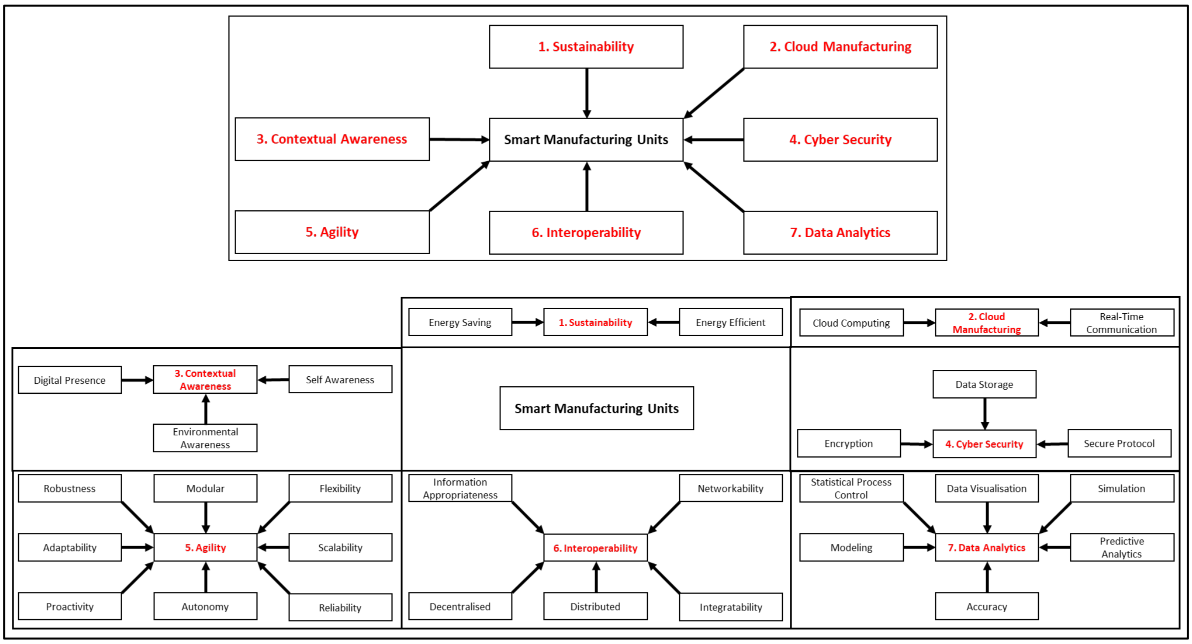


Figure 2-1: SM characteristics

These characteristics are similar to how communication protocols are realised in facets of hardware, software and communication flow implementation. They are implemented in parts with a communication architecture to bring intelligence into manufacturing and realise these characteristics partially or fully.

These characteristics are identified as contextual awareness and interoperability as they are able to define the flow of information throughout a SM setup. The below explanations of Contextual Awareness, Interoperability and their enabling technologies aim to define their inclusion in designing a communication architecture for SM or to act as a realisation from the communication architecture.

Contextual awareness [16] denotes the ability of a SMU to gather, maintain and update the attached network, machine and sensor data around itself. With this contextual awareness, comes the distinctive requirement to uniquely identify the location, type and function of the attachment to fully realise the connected network around itself. An extract of the required contextual information a SMU needs to possess can be seen in Table 2-1.

Table 2-1: Contextual awareness identification example

Item ID	Attachment	Medium	Type	Location	Function
1.1	Sensor	Electrical	Proximity	On-Machine	Product type detection
1.2	Sensor	Electrical	Distance	On-Machine	Process completion percentage
2.1	Machine	Network	Machine-Capper	Next inline	Process completion binary

Interoperability [17] refers to the ability of a SMU to communicate effectively in a network. This communication within the network is commonly referred to as networkability. Effective communication is handled by the ability of a machine to communicate with appropriate information in a decentralised and independent manner, having distributed information and tasks and further being able to integrate itself within the network and manufacturing line.

These traits, when independently instilled into an SMU, grant the machine much needed communication for the requirements of I4.0, but can elevate machine potential when paired with each other.

2.2.7 Communication in Smart Manufacturing

Traditionally, communication protocol development has been dictated using the Open Systems Interconnection (OSI) Standard [18]. Though certain communication schemes, such as RS-422 and RS-485 [17], operate within limited layers of the OSI model, the considerations for functionality have taken inspiration from the OSI standard, allowing for future additions, plug-ins and iterations to be seamless due to the detailed, specific input and output requirements defined at each layer.

For each layer of the OSI model, specific methods to achieve conformity to the standard, either through hardware or software means, have been left unspecified, yet this is not due to the lack of knowledge to specify the technology type required. Rather, it is an open-ended approach that sees longevity built into the standard to allow for evolving technologies and advancements to be integrated within the standard.

However, there are limits to holding back technological advancements with diminishing returns to abide by conformity of the standard, or to rather set the limits of outdated standards and revise their creation, whilst also offering insights from previous adaptations and iterations to refine methodologies of evolvement for communication standards.

The OSI standard incorporates a refined framework [19] used in the creation of protocols and their standardisation. This avoids protocol standards from redundant tasks whilst focusing on parsing information at intersections of the different layers of the OSI model. Similar methods should be used as guidelines for the creation of architectures in the course of refining their criteria and range in the I4.0 space.

This is seen as a requirement [20] around architecture development and integration, as architectures need to be general enough to incorporate future technological developments, but specific enough to solve issues in their use-cases yet still allowing for iterations under their requirement specifications. The OSI model consists of seven layers named in Figure 2-2, with example standards/implementations of each for added context.

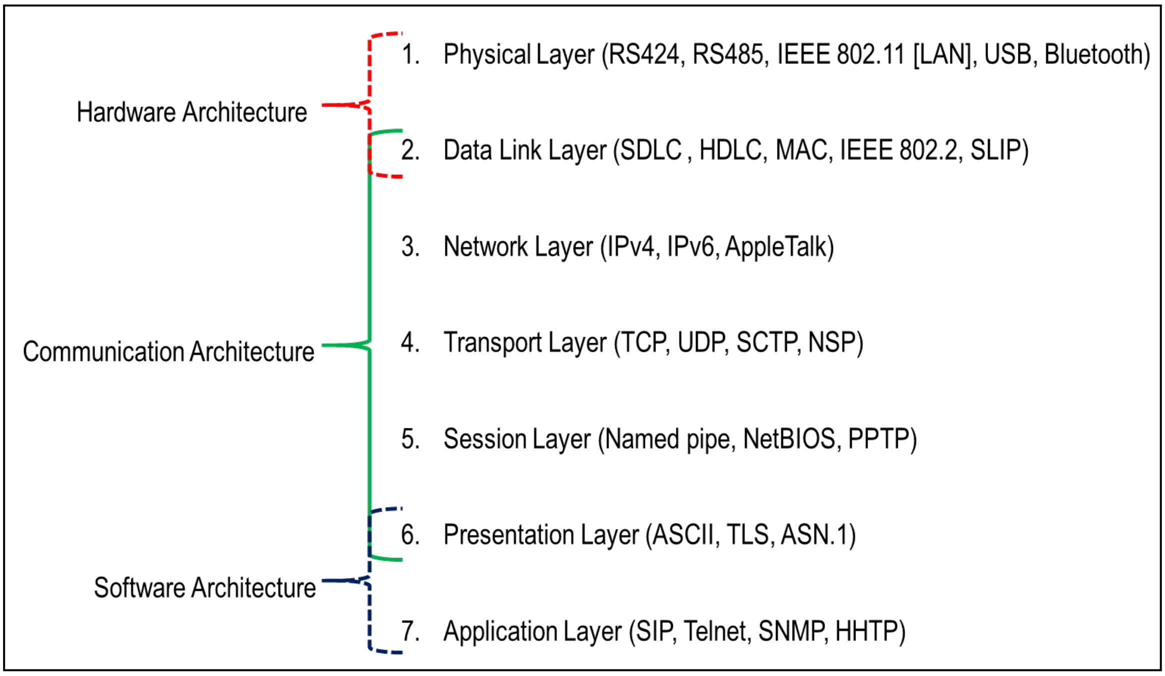


Figure 2-2: OSI model levels

From the generation of the OSI levels, clearer distinctions can be drawn on the separation of hardware, software and communication subsects. Overlaps still exist within the architectures, required to parse information from one layer to another, but modern expansion of communication layers has been necessary to increase number of nodes and data throughput rates. This has left a critical gap [21] in the parsing of information from one node to another with inherent intelligence.

2.2.8 Communication Protocol Examples

2.2.8.1 TCP/IP

The Transmission Control Protocol/Internet Protocol (TCP/IP) [22] is a combination of protocols suites (TCP and IP) to allow for data communication in a point-to-point style. This point-to-point style has been upgraded with later iterations of TCP/IP to include point-to-peer communication styles, to allow for multiple transceivers to be placed on a network.

The TCP session protocol establishes and manages communication between connected devices and enables control over data packet sizes and transmission rates but uses IP protocol to transport data packets. The IP protocol defines data destinations and routes and does not allow for handshaking, attempted reconnecting or successful data delivery. The combination of the TCP and IP protocol for the TCP/IP protocol is the basis of many first communication network styles and allows for almost any internet capable device to send and receive information in a network.

2.2.8.2 IPv4

Internet Protocol version 4 (IPv4) [23] is a fully defined communication protocol, yet it is rooted in the inception of the network layer from its predecessor, the internet protocol (IP). IPv4 saw further development from the internet protocol network layer implementation of the OSI model to allow for personal computers to access information across a network.

With the further development of IPv4 as standalone communication protocol, came the inclusion of domain network, broadcast and host addresses, time-to-live (TTL) and other technical features for improvement over the years. With IPv4's wide global adoption, many I4.0 setups still include it in their implementations where their conformity of communication is a critical metric over data latency and inherent intelligence. IPv4 has also found a growing popularity in IoT, allowing for its easy and cost-effective integration into miniaturised sensors, allowing for data capturing and remote or detached computing.

2.2.8.3 OPC UA

Open Platform Communication Unified Architecture (OPC UA) [24] has seen an increase in iterations being developed for its growing demand. With the beginning of the decade already including improvements for use in I4.0 and IoT as specifically stated by the developers, OPC UA offers one of the most distributed means to communicate information to PLCs, a core component and requirement in automation, manufacturing and SMUs.

OPC achieves this with a re-encapsulation of messages from machines of different vendors, where messages from a source PLC are read, analysed, and repackaged into the format of the destination PLC. This operation, aside from offering great versatility, sacrifices many requirements of I4.0 for machine communication, being intelligence, real-time communication (low latency) and decentralised networks. As with each machine needing to parse its information through a central OPC server for re-encapsulation, the aforementioned requirements become difficult to realise for machine communication.

2.2.9 Smart Manufacturing Protocol

Prior development of this study includes the creation of a newly defined communication protocol for SMUs. This communication protocol specifically allowed for decentralised communication for SMUs and allowed for real-time, low latency communication between machines. This communication protocol, named the smart manufacturing protocol (SMP) [24], also allowed for iterative development to include unique identifiers of the source machine, to update information accordingly and provide specific details of the machine, product and manufacturing status.

This information is embedded within the header of the data packet, under the *Type of Service*, *Identifier* and *Flag* sections, respectively, seen in Figure 2-3 [25]. This SMP was largely based off the IPv4 protocol in its *Network Layer*, a GET and PUT method for transmission of information between SMUs in its T*ransport Layer*, a decentralised approach in its *Physical* and *Link Layers* and an update differences method within the *Session Layer*.

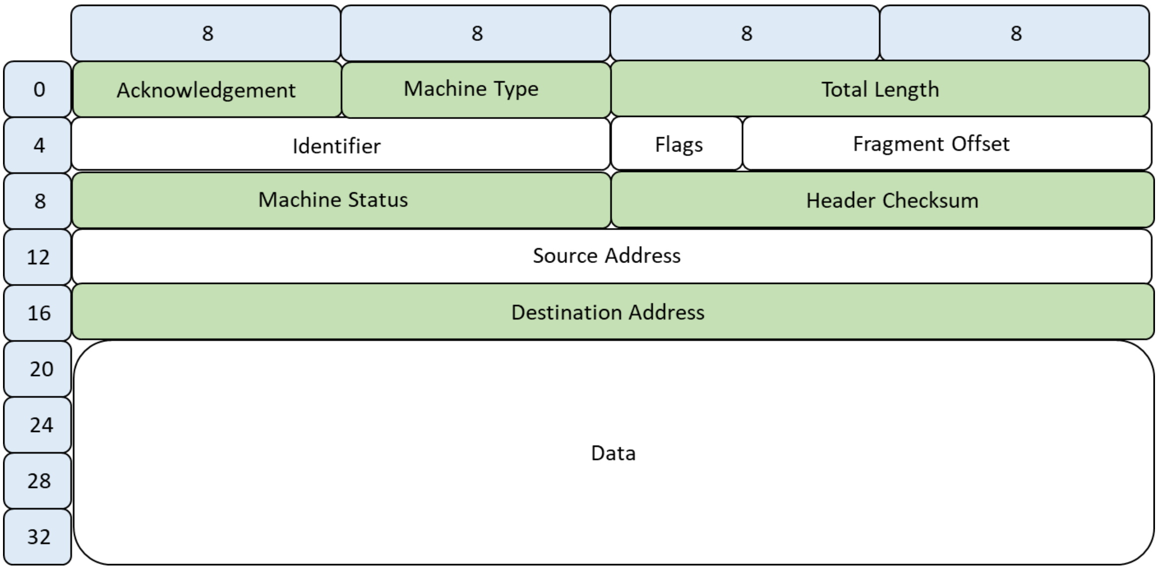


Figure 2-3: SMP data packet

Many different communication protocols can be used for machine-to-machine communication. However, these communication protocols need to be evaluated for the metrics, ability and compatibility to be used within the I4.0 and SM space. These factors include latency, communication time, payload limit and ability to handle information at different ISO levels.

As seen in Figure 2-4, a comparison of machine-to-machine communication protocols illustrates their communication time. Seeing as the SMP offers one of the better communication times and ability to process information across different ISO models, it will form the basis of communication between SMUs.

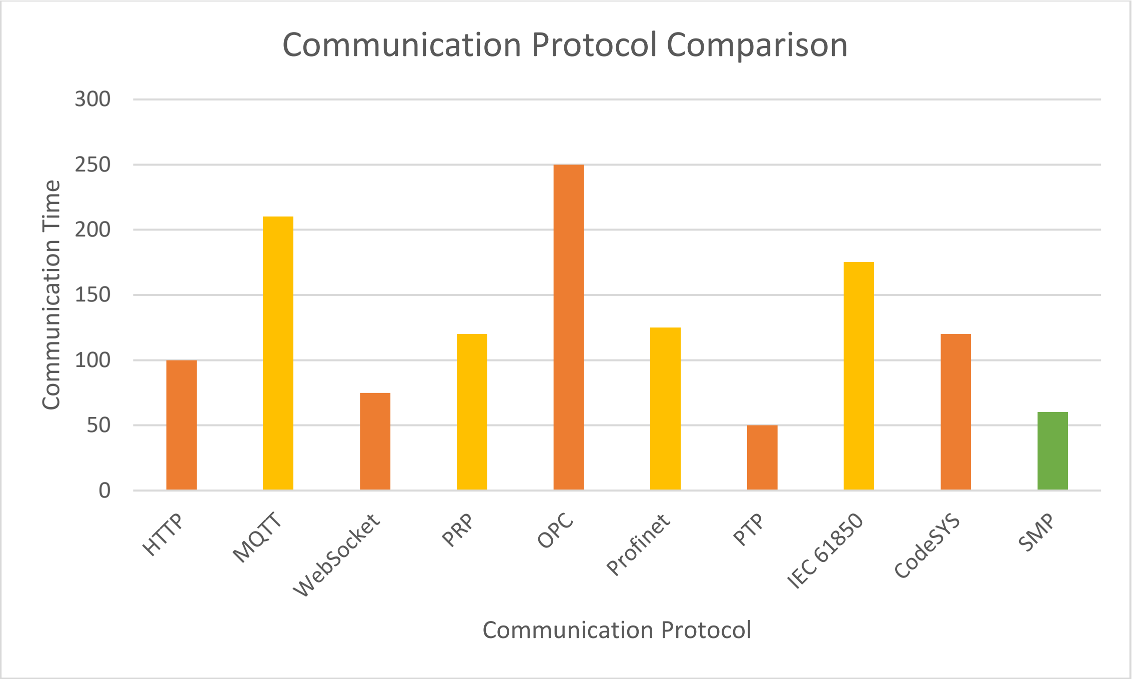


Figure 2-4: Communication protocol comparison

2.2.9.1 SMU Performance & Capability

Each SMU allows for 50 kB work memory, 1 MB load memory, 4096 bytes bit memory and an additional 256 GB memory card. Whereas the database block is limited to 50 kB of memory, each SMU is further customised with additional signal or control boards based on their required functionality. It is an added advantage of the SMU core electronics to be highly modular, yet network modularity still needs to be evaluated for best practice implementation and turnaround time.

The SMUs are also connected to each other through their network connections and communicate by means of the SMP. The SMP allows each SMU to communicate through a decentralised method that allows for packeted information to be transmitted and gathered through PUT and GET methods.

As each SMU has a direct link to each other and through the use of the PUT/GET methods, the SMU network (layer 3) is seen as a decentralised network. The average weighted packet communicated through the SMP has a latency of ~47ms, a transmission time of <10ms and a jitter response of <100us. These metrics of the SMP classifying the protocol within real-time (<100ms) are based off the requirements of I4.0.

The SMP data packet can be seen in Figure 2-3. When SMUs are first attached to a network, they broadcast a message on the 255.255.255.255 address. A network router then returns an assigned address. This destination and source addressing is similar to and inspired by IPv4. A header checksum, total length and acknowledgement field is similarly inspired by IPv4.

The header checksum is a pre-determined 16-bit numerical value, used to ensure the integrity of the transmitted data packet at the receiver device. The acknowledgement field allows the enabling and verification of received messages from the receiver to the transmitter. The total field length can be used to designate the total length of a data packet to the end of a transmission. This base of fields allows SMUs to communicate with one another in an asynchronous and bi-directional fashion, enabling the first trait of the interoperability characteristics of SMUs, being networkability.

A broadcast routine of connections on a network to collect IP addresses allows SMUs to be integrated into the network autonomously, physical and autonomous integration into the manufacturing line is outside the scope of this study.

SMUs are also routinely instructed with the use of the SMP to update each SMU on the network with appropriate information. This routine is designed outside the usual instruction set of the SMP. Also designed into the SMUs are designated memory locations for production data. These memory locations are dynamically and un-specifically allocated. An example of these memory locations is the total products produced that can be stored without the unique ID and specific production information of single production runs. This allows SMUs to track production information without specific information about that product being captured, i.e. water bottles or cars. This routine of storing and updating information at intervals allows for information to be distributed throughout the network. Since each SMU is instructed to update every other SMU, a single SMU can hold all production data at a given time. This distribution routine is seen as one component of the communication architecture at the machine-to-machine communication layer.

With the SMP routines updating information congruently across the machine network and working in a decentralised manner, a unique fetch method can be realised. As seen in Figure 2-5, SMU 1 is unable to communicate with SMU 2 directly. In this case, SMU 1 has the capability to be aware of SMU 2 by querying the network information of SMU 3.

The returned data packets will show communication information from SMU 2. SMU 1 is effectively able to GET and PUT information from SMU 2 through SMU 3 by hopping the information when needed. This total destination address can be stored in the data field of the SMP packet. This hop strategy would require a double latency and transmission delay. However, with the SMP latency being averaged at ~47ms, this double hop return can still be done within real-time. This hop strategy does necessitate an extra layer of complexity, as information that is updated from SMU 2 to SMU 3 should surely also be updated on SMU 1 in a timely fashion.

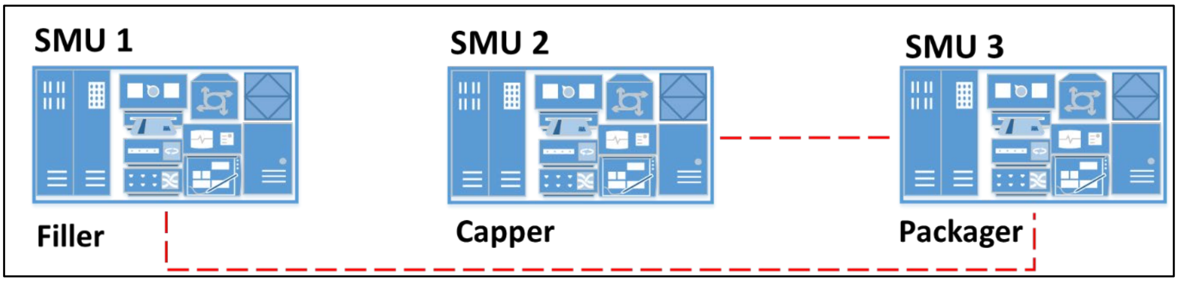


Figure 2-5: SMP hop strategy

However, a better approach can exist to this hop strategy. Since the information of the machines is congruent on each machine, SMU 1 can decide what latest information is required for updates, calculations and computations. SMU 1 can therefore obtain information about SMU 2 from SMU 3 on non-time critical requests.

This congruency of information and querying based on time sensitive data is seen as an advantageous tool in a communication network and can be added as a distributed network trait of a communication architecture. Timing requirements to control the congruency of information and acting on the cruciality of information should be stipulated on a case to case basis.

Information that is updated from one SMU to another should be designated as appropriate. For instance, if SMU 2 requires information on the bottle cap height, this information can be passed on from SMU 1. SMU 3 might not need this information. Therefore, although SMU 1 does send a message to update SMU 2, this can be stored in the work memory of SMU 2. Since this information is not relevant to the overall production information, SMU 3 is not updated with this temporary information. This allows the SMUs to send appropriate information that is divided up amongst machine specific information, production specific information, and network specific information.

The above-mentioned abilities of the SMUs to communicate in a network, allows for a collection of information sharing strategies that aid in the contextual awareness of the SMUs. With the SMUs being able to internally identify with connected sensors and machines, the SMUs are contextually aware of their own capabilities. Through the information communicated from every other SMU, the SMUs are able to become contextually aware of the surrounding network. SMUs are therefore able to be aware of their environment and current manufacturing status with the use of contextual awareness.

Alongside the highlighted qualities of the SMUs that are realised through the SMP and certain routines, it becomes imperative to derive responsibilities per communication layer of the case study. For this study, it is designated that the cloud server is responsible for high level production completion storage, human-machine interfacing for overviews of production status, and additions from IoT sensors detached from the manufacturing line.

This information allows for greater data analytics to be completed at a top level. The central server is responsible for housing and updating digital twin communication, scheduling production tasks to the SMUs, performing data analytics at a high level of each SMU, and validating production information from each SMU is within statistical process control.

The SMUs are tasked with the following:

- Communicate production information between SMUs in real-time.
- Be responsive to production interruptions and changing conditions.
- Perform accurate measurements and production, ensure data transmission between SMUs occurs periodically,
- Ensure individual production information is within process control.
- Be contextually aware of manufacturing requirements and current production and allow for edge computing services.

A summary of these responsibilities, enabling technologies and tasks can be seen in Figure 2-6.

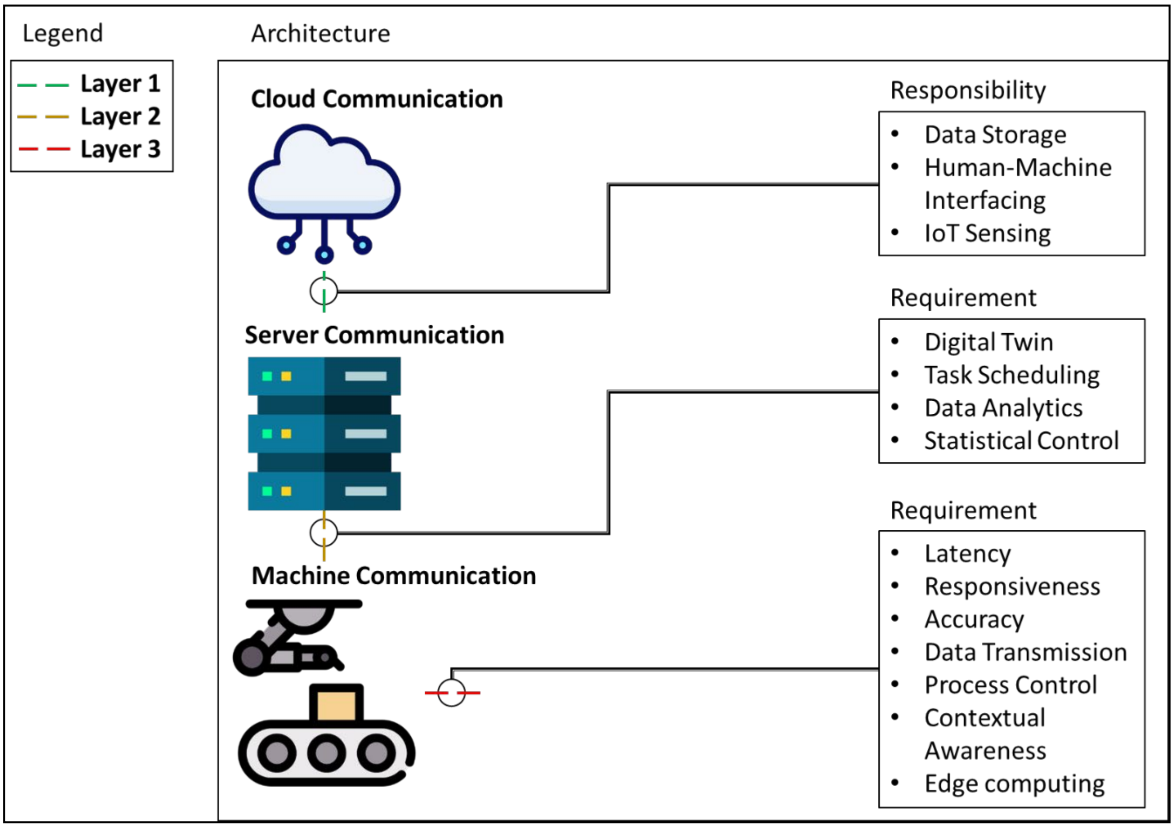


Figure 2-6: SM case study responsibilities

2.3 Baseline Manufacturing Setup Performance

This section introduces the applicable calculations and resources required for evaluation and implementation throughout the study.

2.3.1 SMU Performance

Each SMU is tasked with a specific objective within the manufacturing line. SMU 1 is tasked with filling each water bottle to the correct level. SMU 2 is required to cap the next water bottle fed in from SMU 2. SMU 3 is responsible for packaging the water bottles according to the customer orders and required order output from the central server. Each SMU is physically joined by the conveyer belt system, to allow for water bottles to be passed along the manufacturing line.

SMU 1's fill rate is detailed as:

$$Flow\ Rate = \frac{Volumetric\ Flow\ Rate}{Bottle\ Size}$$

Where:

- Volumetric Flow Rate is the Calculation of the attached Pump Cross-Sectional Area multiplied by the Output Velocity.

Formula 2-1: SMU 1 fill rate

SMU 2 and SMU 3 make use of timing limits to indicate warning and error flags for products that remain at their stations too long.

2.3.2 Central Server Performance

During the production of the water bottles, the SMUs and central server collect information and statistics for the Kadence and verification of the manufacturing line. The information that is collected from customers are ***CustomerName***, ***Customer300mlOrder,*** ***Customer500mlOrder***, and ***CustomerRequiredDate***. This information is used to correlate and tie product serial numbers to customer orders in order to track; completion progress of orders, demarcating orders for delivery, marking products for particular orders when feed in orders are altered. While most of the collected fields are simplistic to capture and assign from product serial numbers and manufacturing strategies, the ***CustomerRequiredDate*** is a unique calculation that is assigned during production scheduling to aid in future optimisations of production completion metrics. The ***CustomerRequiredDate*** is defined using a formula developed to consider the supply chain demands and is provided in Formula 2-2.

CustomerRequiredDate is defined as:

$$D = RoundDown\left(\frac{t}{10}\right) + RoundDown\left(\frac{f}{10}\right) + 2$$

Where:

- D is the CustomerRequiredDate
- t is the number of 300ml water bottles
- f is the number of 500ml water bottles

Formula 2-2: Customer required delivery date

Communication of all customer orders is sent to the central server from the cloud server as they are acquired. The central server houses a digital twin of the manufacturing setup which allows for a simulated environment [43] of the central server to determine the optimised order to complete customer commands.

These optimisations can include: a first-in, first-out approach (FIFO), sort by CustomerRequiredDate, optimise based on number of water bottles and CustomerRequiredDate, all 300ml water bottles first, and random. The central server is able to communicate these order requirements to each machine using an OPC UA connection.

Lastly, the manufacturing case study makes use of three SMUs to complete the production tasks. Each SMU is developed around Siemens S7-1200 Modules, with relay expansion modules and network expansion modules. These expansion modules allow for a direct point-to-point communication module between each SMU and the central server directly.

Whereas the production efficiency may be calculated individually by each SMU, this could at times pass over the interconnections and transmission of products between machines. A purely holistic overview of production efficiency calculations from the central server might forego details of bottlenecks between machines and greater detail may be needed to understand root causes of bottlenecks by monitoring individual machine production efficiencies alongside the central server. This study therefore makes use of a hybrid system, for each SMU to calculate their own production efficiency whilst the central server calculates the overall production efficiency per product item and for an entire production run. The production efficiency formula is provided in Formula 2-3.

$$Efficiency = \frac{Produced\ Quantity}{Produced\ target\ Quantity} * 100$$

Formula 2-3: Production efficiency

2.3.3 Cloud Server Performance

Customers are greeted with a web-based application to place their orders with the required number of 300ml and 500ml water bottles and required delivery date. The role of the cloud server could be expanded to organise, schedule and communicate with all distributed smart factories; however, within this study, the simplistic waterfall structure only necessitates the use of order collection and communication of orders to the central server. Future studies may require a holistic overview to be conducted by the cloud server, to determine current production loads of each smart factory and forecast order requirements in order to proactively prepare for future events.

2.4 Requirements of the Communication Architecture

With the requirements of SMUs explored and the baseline manufacturing scenario shared, a clearer conclusion on complimentary aspects and requirement shared between SMUs and data control can be drawn. Though other attributes of SMUs are dissimilar to the flow and structure of information, a communication architecture for SM should structure data in a way that allows for:

- information to be accessible across the network;
- information not to be reliant on singular paths;
- information that can be calculated and agreed upon throughout the network;
- information to be directly accessible where it is needed, and
- information that relates to the specific environment around it.

2.5 Architectures

The inclusion of SMU characteristics should be seen as the minimum criteria of classification for intelligent control and classification of machine intelligence. However, a holistic view alongside a non-specialisation approach of improvement needs to be considered to yield a much greater return on investment intelligence within I4.0.

For example, information that is appropriate and specific at all levels inherently reduces latency and overhead, making for speedy operation and calculations. This, coupled with a machine being contextually aware of its surroundings, allows for intelligent access and storage of pertinent information, making for a first link of relations of a machine in a communication scheme or architecture.

This is further iterated by dedicating optimally placed machine by information access for specific calculations, allowing for network hierarchies to be established, where resultants are communicated throughout a network instead of raw data points. This second iteration, though using the same relations, provides a second degree of intelligence and control within a system that can be inherited by all systems implemented alongside it, if there is commonality of approach and reference to a standard implementation around an architecture. This allows for greater advantages when implementing these characteristics in a known fashion for specific challenges whilst holding up critical computational resources.

Architectures are designed [26] and defined [27] by the makeup of core components and relationships between them. With many different architecture models existing, example definitions for creating new architectures are easy to come by. There is a greater challenge, however, in defining the requirements for the creation and important considerations of an architecture, which would ultimately underpin the longevity value thereof.

The proceeding sections of this chapter – 2.5.1, 2.5.2 and 2.5.3 – summarise previous literature of architectures to help identify critical aspects in the development of a newer branch of architectures namely, communication architectures.

2.5.1 Architecture Elements

2.5.1.1 Core Components

Architectures are classified by their core components [28] which are integral to the successful implementation and resolution of the use-case they are defined by. These core components are ultimately not always mutually exclusive and, at times, provide areas of overlap. Considerations outside of these core components still need to be defined, as the core components serve as direct inputs and outputs to handle operations within the architecture. An architecture that fails to fully consider implementable technologies outside of its scope, runs the risk of not being included into setups due to lacklustre functionality.

2.5.1.2 Relationships

Whether or not core components exhibit overlaps, requirements do exist in the form of relationships to describe the effective handover of information between core components [29]. Relationships need to be drawn up between most, if not all, core components. These relationships help not only define the requirements of the core components, and to an extent the architecture itself, but they also show the symbiotic relation between each core component. With iterations throughout the architecture's lifetime, it is only through these relationships that a discernible cause-and-effect will be effectively drawn before iteration.

2.5.2 Types of Architectures Models

Architectures can be designed with a range of models [30] connecting core components and their relationships. These models have been created as generic blueprints to allow for a multitude of models for varying fields of industry, even spanning logistics and supply chains, with no limitations to relationships or core component parts.

With these traits, the adaptability of the models to create architectures for I4.0 is surprising. However, no reference models yet exist to build from and capture the complexity of relationships and required core components for communication between machines in I4.0. This reveals more evident research gaps on the refinement of communication architecture development for SMUs in I4.0, with their effectiveness and elegance still to be scored. Some reference models that show architectural development which has been used in hardware and software architectures are expanded on in the following subsections.

2.5.2.1 BASE Model

A BASE model architecture as stated in [31] is used in the creation of software and supply chain architectures and is described by its four components consisting of the Biography, Attributes, Schedule and Execution (BASE) of a system. The four main components of the BASE model are connected through a principal relationship, named the Manager. This Manager facilitates inter-relations requests, where these requests could require inputs and parsing between multiple components of the BASE model, which can be seen in Figure 2-7.

For instance, within a supply chain architecture, the Schedule component oversees defining when new tasks are first scheduled with the architecture, and the Biography is responsible for time recording of the actual start, production and end time of each task. With tasks having a different mix of start times, required delivery times and estimated production times, it is the duty of the Manager to pass sequentially started tasks to the Biography to capture its timing requirements (the actual start, production and end times).

This parsing and reorganisation requires the input of the Manager. However, tasks can be assigned certain metadata, that can either be independently captured by the attributes of the core components or related by the Schedule during first capture. This relation, albeit not included as a named relation, can exist when needed but is open to interpretation by a system operator.

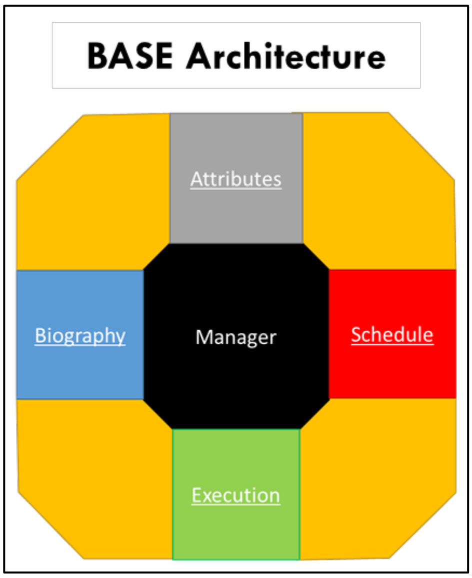


Figure 2-7: BASE architecture

2.5.2.2 *SOLID Model*

A SOLID Architecture [32] is predominantly used within software development and is used to help create object-orientated designs by defining the principles of the acronym to:

S – Single-responsibility Principle
O – Open-closed Principle
L – Liskov Substitution Principle
I – Interface Segregation Principle
D – Dependency Inversion Principle

Each of the principles are a set of guidelines and/or expressions that, once met, can aid in sharing the code with different collaborators, leading to improve, expand, test and refractor iterations with fewer issues. This does come with a higher compliance to test each change against all principles, nevertheless, it does help define iterations with intentional implementations and allows for affected components to be considered. It is also important to note that the SOLID Architecture does not mention any particular coding language in its description, allowing for universal implementation. A depiction of a SOLID Architecture can be seen in Figure 2-8.

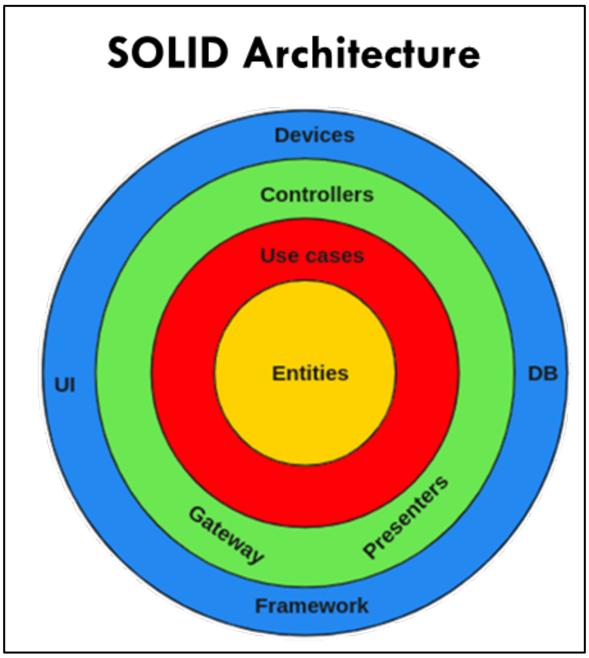


Figure 2-8: SOLID model

2.5.3 Different Architecture Implementations

2.5.3.1 Hardware Architecture

Hardware architecture [33] originated from the need to improve cycle development time and expansion of new electronic technologies by incorporating lessons learnt and best practices with the requirements from manufacturing facilities. This was largely seen as advantageous to design with an architecture, to help mitigate complex regulations and requirements and offer greater understanding between the changing physical aspects to meet physical chip size scaling and practical voltage limits.

Hardware architectures are divided into three main categories, being system design, instruction set architecture and microarchitecture with many popularised architectures consisting of Von Neumann architecture, Harvard architecture, Modified Harvard Architecture and, the most recent, Reduced Instruction Set Computers 5 (RISC-V) architecture.

2.5.3.2 Software Architecture

Software architectures [34], similar to other architectures, are used as blueprints at system level designs to describe the interactions with connected parts and the functionality input/output requirements of each part at higher level detail. Although the input/output description across architectures remains consistent, their implementations and considerations are often what change, and to that effect, so do their core components and relationships.

This leads software architectures to have their own distinct appearance, where they can describe the requirements of a software solution and therefore the requirements, input and outputs of associated methods and classes. Common examples of software architectures include event-driven, space-based, layered (n-tier), model view presenter/controller, orchestration architecture, and domain driven design (DDD) architectures.

2.5.3.3 Communication Architecture

Architectures have traditionally been used to standardise implementations and designs which allow for optimised designs and clear definition of individual component requirements. The first few notable architectures arose from the hardware design requirements of integrated circuits (IC), to standardise performance and make use of manufacturing limitations at the time. The inclusion of software architectures was then followed onto these hardware architectures as an easy way to generate and review codes, to reduce development cycle time and eliminate software bugs.

This inclusion of software architectures paved the way for standard libraries and code lines to be created with each new cycle and the translation of hardware language into application languages, such as the grouping of assembler code lines of working registers into statements. It is however clear to see with each introduction of an architecture that greater and standardised improvements can help shape an industry. As many communication protocols are developed either to contribute to single layers of implementation or are created completely informationally disconnected from the devices they are connecting to, it is evident that the development and inclusion of communication architectures into I4.0 can leverage commonalities of the industry standards and requirements.

Communication architectures are still in their infancy, yet studies within the sphere of influence have generated notable research [34, 35, 36]. These areas of research however encompass a communication protocol, but additional work is required for an encompassing communication architecture. Communication architecture should consider the particular flow of information, regardless of destination and format. Communication architectures should also describe a strategy to include considerations around entire network layers and its different attached machines, sensors and nodes.

Communication architectures can therefore make use of a culmination of different communication protocols at each layer, and can best describe considerations, inputs and outputs required at each level, as well as transitions between levels to ensure information is pertinent, efficient and appropriate. The development of a communication architecture within this study is hoped to create a viewpoint forward for consideration and classification of requirements for communication architectures, creation and implementation of communication architectures, evaluation and use-cases of communication architectures and, lastly, the further development of communication architectures.

This study thus aims to show how the effective use of the SMP at a machine communication level and considerations for data flow and operations can allow for intelligent decisions to be made from SMUs by inheriting the information exchanged by each machine to achieve the commonalities between the I4.0 requirements and SMP design.

This information flow between each layer of an implementation also needs to allow for effective designation of information and computations whilst being adaptable for other communication protocols and technology types, where needed, as to not erode crucial information and, ultimately, functionality of the smart manufacturing setup. This will ultimately allow for inclusions before and after the manufacturing line itself, to include customer order supply chains, re-configurability and delivery logistics, to name a few.

With the SMUs inheriting intelligence from communication exchange and under the introduction of a communication architecture, this study aims to evaluate the implementation of said architecture effects to improve production efficiency, identify bottlenecks and adapt to changing manufacturing conditions. These evaluations are seen as one potential classification of intelligence of a machine, where, again, the end manufacturing goal is not the specific metric but the machine's ability to identify and act towards these manufacturing goals.

This communication architecture is also noted by this study as one possible method for gathering results, and it is hoped to serve as a new perspective on how to overcome manufacturing challenges and effective solution implementations with communication architectures.

2.6 Creating the Communication Architecture

This section outlines the original considerations and requirements of a communication architecture implementation to assist with SM setups to enable improvements in production efficiency, reduce bottlenecks and introduce intelligence into the setup. This is achieved by evaluating and structuring the data flows and requirements for the different layers of a manufacturing setup. This is further detailed by the method of acting upon data, how specifically a SMU may act on this data, and what data each level of the manufacturing setup should directly be allowed to communicate with and be responsible for storing.

2.6.1 Creating the Communication Architecture Core Components

The ability of SMUs to communicate within a network is defined from their characteristics [35] and enabling technologies. It is imperative to distinguish and define the responsibilities of these characteristics on how they integrate throughout the machine-to-machine communication layer. This distinction and definition helps create the core components of the communication architecture for SM and SMUs.

This communication architecture is best implemented at the third layer of communication, as seen in Figure 2-6, as the position is best located to handle the complexity of information transfer and generation for production orientated implementations. Although the output of information expands when communicated to surrounding layers (layers 2 and 1), this study employs the use of appropriate information throughout the communication network to reduce unnecessary redundancy, allowing for a greater development of intelligence in the SMUs and to improve production efficiency with the aid of network responsibilities.

The core components of the communication architecture that are central to machine communication [36] are thus described in this section, whereas the relationships between the core components are described in Section 2.6.2. Between the core components and relationships of machine communication, the requirements, responsibilities and data conformity can be observed at other communication layers in the SM case study.

Similarly, this communication architecture will also need to consider the implications of other auxiliary technologies, of enabling attributes in I4.0. These enabling attributes, such as digital twins, IoT and collaborative decision making, will need to be discussed based on how they best fit within the communication architecture. These discussions bring about iteration to the communication architecture, for upkeep and maintenance of technological improvements and allowing longevity of the communication architecture. Other technology implementations, such as AI, that would require other dedicated architectures, i.e. hardware or software, can then also be evaluated for implementation in an I4.0 manufacturing line.

The communication architecture of machine communication is underpinned with solving a centralistic problem of data organisation allowing for greater data transfer and requiring effective data handling [37] at multiple stages of communication. This is achieved within the machine-to-machine communication level by aligning communication strategies such as protocols and route maps with the architecture. This implementation of the communication architecture is hoped to improve the specific challenge to elevate production efficiency and reduce bottlenecks during manufacturing by aligning data transfer and responsibility during multiple stages with steps outside the communication protocol.

Though some core components concern themselves with building the base of requirements for the communication architecture, some core components compliment the data organisation and flow existing between other core components. These core component distinctions are shown with the difference of responsibilities between core components of their relationships, and the parsing of information through their relationships. These responsibilities show the implementation map to design this communication architecture into SMUs and show how the natural organisation of data compliments the requirements of SMUs and I4.0.

The similarities between a communication architecture's core components and SM characteristics are identified where similar technological inclusions can be made to satisfy both requirements. These requirements are largely based on the flow and handling of information that is exchanged between two or more entities within the SM setup. The communication architecture defines the core components with characteristics and enabling technologies similar to that of SM characteristics. There are however distinctions between the structure levels of the core components and SM characteristics.

The fundamental SM characteristics identified to best model for core components in the communication architecture are interoperability and contextual awareness, where their enabling technologies are further incorporated. The core components are information appropriateness, distribution, decentralisation and contextual awareness. With additional setup of the communication architecture in a multiple layer's network (cloud server, central server and SMUs), it is prudent to also model the communication architecture with a network hierarchy in mind.

2.6.1.1 *Information Appropriateness*

The information appropriateness [38] component of the communication architecture forms the bridge within interoperability from SM. Information appropriateness handles use-cases to allow specific information to reach specific units outside of the general routine when needing to update all machines on the network. This information that is updated, is usually temporary information used to optimise and foreshadow events.

Appropriate information must classify transmitted information as production information, which is to be updated amongst all machines on the network, or machine information, which is to be updated on a specific machine on the network. Appropriate information aims to invoke a response for the receiving machine to update ongoing events or trigger actions cases in production.

Information appropriateness is considered a critical aspect within the communication architecture, as it can reduce unnecessary information shared across the network. This is possible through the use of classification of information and can reduce unnecessary redundancy, as the information is specific to machines that are updated within the network.

Appropriate information can also allow for intelligence to be introduced within SMUs. This intelligence is possible as, instead of direct messages from senders to receivers, information is first computed before being transmitted. This computation allows for invoked cases on one SMU to enact a response to send appropriate information to another SMU.

The use of appropriate information can also discern the different levels within the communication network, transmitting higher level or resultant production information to the central server, rather than raw data. This information again follows a reduction in unnecessary redundancy, as specific information is transmitted where it is used.

2.6.1.2 *Distribution*

Information and commands are required to be distributed [39] to fully realise decentralisation and autonomy. All machines across a similar network layer should be equal. This equality is only possible once machines have the same possibilities and information. This allows SMU to act upon information when it is appropriate. In an adverse case, a single machine would always need to pass tasks and information to relevant parties, centralising the network around this machine.

Distribution would require information to be updated across the network layer within timely intervals. This distribution of information allows machines to characterise information under time sensitive tasks, to either perform additional GET instructions for up-to-date information, or use the last known information where it is sufficient. Distribution of information should also consider the update interval to adjacent network layers within timely intervals.

2.6.1.3 Decentralisation

Decentralisation [40, 41] allows for machines to act independently and autonomously. By creating a decentralised network, each machine on a network is able to communicate with the other machine directly. This removes access nodes connecting machines, which centralised the network around a certain point. Instead, each machine is itself an access point. All machines can however still agree and limit themselves around a common network addressing scheme for simplicity. When machines are able to communicate in a decentralised manner, general information is shared with all machines, and specific information is shared with destination machines.

Decentralisation allows machines to be robust against single point network failures. When the machines and information are decentralised, machines are able to create decisions based on their responsibility. These decisions can be autonomous from the machine’s required responsibility for certain tasks, either as a direct requirement to their production outputs or a network hierarchy.

Decentralisation is thus classified, under the architecture, as requiring machines to be able to independently communicate to one another as its input requirement. The output requirement of decentralisation is then classified as communicating specific and general information to machines.

2.6.1.4 Contextual Awareness

Contextual awareness [42, 43], albeit similar to interoperability as a SMU characteristic, is incorporated as an entire core component in the communication architecture. This inclusion is due to the similarity and complementary nature of the enabling technologies under contextual awareness, being digital presence and machine awareness.

Although the digital presence of a machine is more concerned with a machine's ability to communicate at other network levels, their implementation in the architecture is similar. Furthermore, the digital presence of a machine acts as the gateway of communication of other network layers, bringing about machine awareness in a communication sense to the other network layers.

The enabling technologies of the contextual awareness characteristic also seem unmatched to the requirement of core components, with many relationships similarly defining the two enabling technologies. This contextual awareness characteristic is therefore included as an entire component, to unify the creation of relationships between core components.

A machine's *digital presence* is concerned with the ability of said machine to be visible on a network. With this visibility comes its input requirements, namely being accessible and the ability to be updated. The digital presence of a machine should replicate the machine's input, output and production status on a manufacturing line.

Within this study, this is successfully handled by the SMP with the use of the "Machine Status" field of the data packet. This continuous presence of a machine allows for vital information to be passed about the machine and the network. For instance, a continuous update of machines on a network to a central location, can redundantly allow for machines to identify other machines on the network without a direct connection to the machine.

This digital presence can also be used in conjunction with other core components. For instance, a split of network machines can be used to split the responsibilities and functions of a manufacturing line but still show the capability of a network. This information further enables a machine to be contextually aware of its surroundings, and in this case, specifically its network contextual awareness.

Machine awareness concerns itself with the ability of a machine to be aware of its surroundings. This technology specifically looks at a machine being physically aware of attached sensors, components and other machines to achieve manufacturing tasks. The ability to identify these sensors, components and machines is considered its input requirements. The identification of machines is already handled with the inclusion of a machine communication protocol. For this study, the SMP already allows for identification of attached machines with the "Machine Identifier" field of the communication packet.

Sensors and components can similarly be identified with the use of a unified standard for identification. This is included in the study as a pseudo-communication protocol, as sensor and component serial numbers are cross compared with the use of lookup tables defined in the memory card information of each machine. This machine awareness thus allows for an internal and external identification of a machine and production line's capabilities.

2.6.1.5 Network Hierarchy

Network hierarchies [44, 45] identify its inclusionary zone around machines and layers, to organise tasks amongst machines. This organisation is an attempt to reduce duplicated calculations and reduce network traffic by reducing network loads at singular points and avoiding failed communication attempts. The network hierarchy will therefore require agreement between SMUs to schedule tasks and responsibilities. The network hierarchy is further required to collect and understand the information within the communication network to assign responsibilities where they are best poised to be calculated or where the least instantaneous network load exists. This implementation is possible through the collection of information between machines and designing dedicated inflection points for machines to evaluate and enact upon.

The complete communication architecture for SMUs can be seen below in Figure 2-9. This communication architecture is majorly implemented in the third layer of communication, as a specific focus on its communication protocol plays a paramount task in its decisions. However, to fully realise the architecture in implementation, input considerations must be taken for first- and second-layer communication. This is more adequately seen in the network hierarchy core component. As with information and tasks being able to be scheduled decoupled from specific machines, responsibilities of each network layer need to be stipulated on adjacent communication layers. As a result of this requirement, additional components should be discussed for the purpose of implementation of the communication architecture in the SM case study.

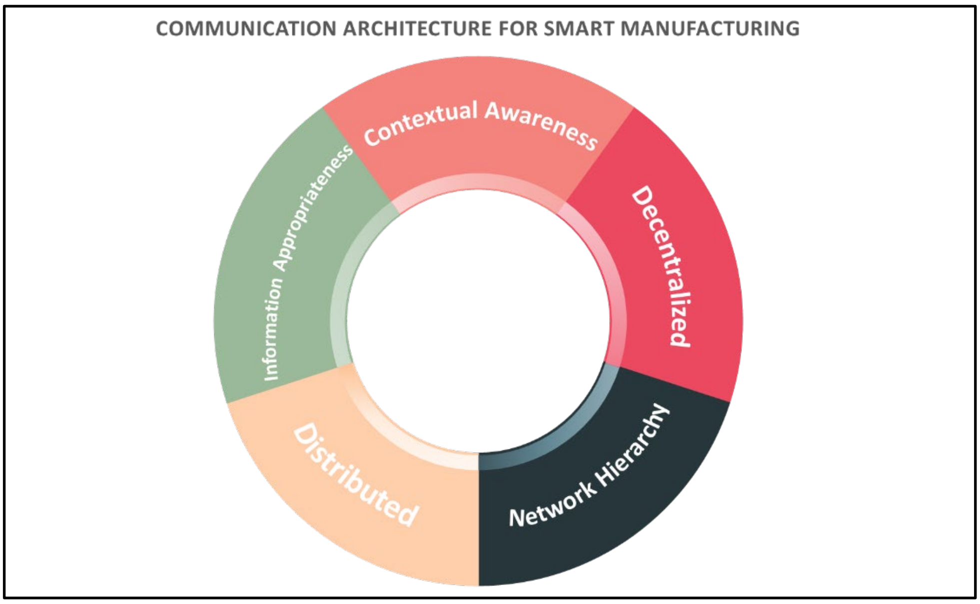


Figure 2-9: Communication architecture for SMUs

2.6.2 Communication Architecture Relationships

The relationships between core components exist not only with the neighbours of other core components, but also both opposite core components. This leads each core component to be related to each core component in a holistic connection. For these relations, each core component has its two immediate neighbour relations and their opposite component relation detailed by the gradient colour lines on the edges of the core components. These gradient lines are illustrative of the start colour core component, end colour core component and their exchange with the gradual gradient. For convenience, most architectures omit relation lines due to larger individual core components, reducing clutter.

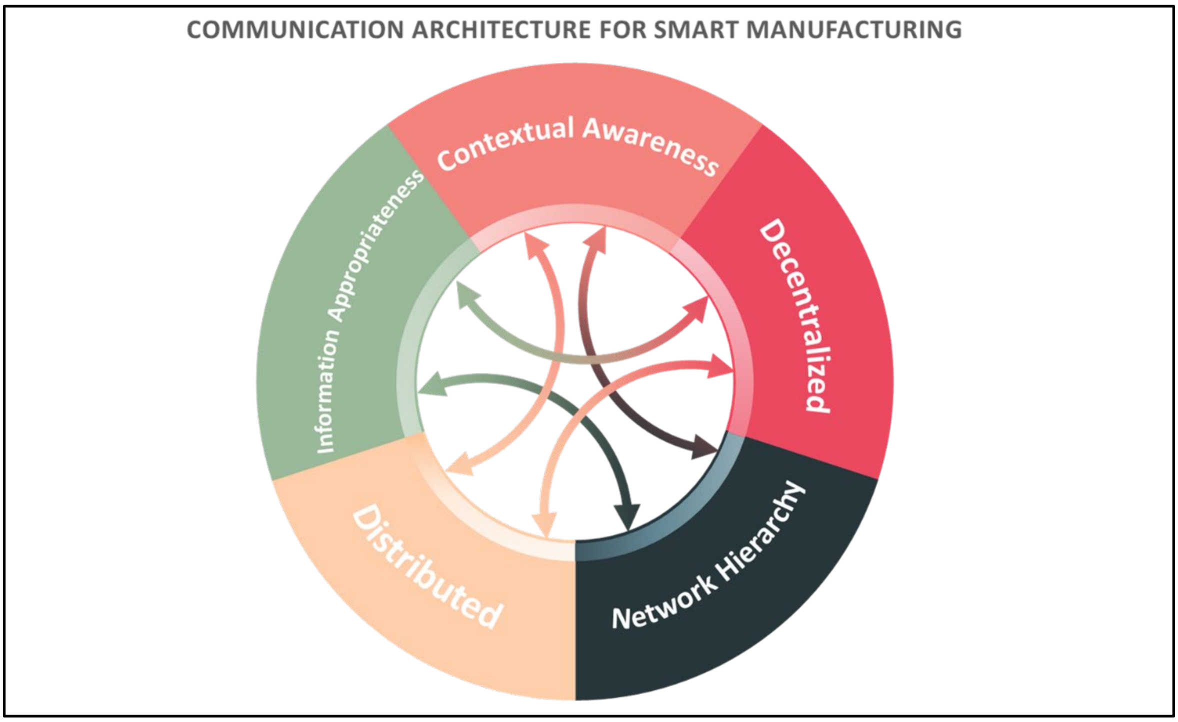


Figure 2-10: Communication architecture with relationships

The summary of the relations between the core components are thus summarised in Table 2-2. Note that identical relationships per core component are marked with a light grey tint.

Table 2-2: Communication architecture relationships

Core Component (Start)	Relation core component (End)	Exchange considerations (Relationship)
Decentralisation	Contextual Awareness	Machine Identification
	Information Appropriateness	Network Identification
	Distribution	Timely Updates
	Network Hierarchy	Classification
Contextual Awareness	Decentralisation	Machine Identification
	Information Appropriateness	Predictability
	Distribution	Network Accessibility
	Network Hierarchy	Machine Agreement
Information Appropriateness	Decentralisation	Network Identification
	Contextual Awareness	Predictability
	Distribution	Decision Making
	Network Hierarchy	Task Scheduling
Distribution	Decentralisation	Timely Updates
	Contextual Awareness	Network Accessibility
	Information Appropriateness	Decision Making
	Network Hierarchy	Reduced Redundancy
Network Hierarchy	Decentralisation	Classification
	Contextual Awareness	Machine Agreement
	Information Appropriateness	Task Scheduling
	Distribution	Reduced Redundancy

2.6.2.1 Decentralisation with Contextual Awareness – Machine Identification

The relationship of decentralisation to contextual awareness is realised by machine identification. Identifying the type of machine, along with the destination address, adds an additional layer that can be used to differentiate machines in a network. If multiple paths within a manufacturing line exist to create parallel production processes, a machine is able to self-identify this parallelism by identifying the multiple subsequent attached machines with different machine identifier tags.

Furthermore, machines would also be able to identify manufacturing line capabilities. By collecting the different machine identifier tags in a network, machines can piece together possible production capabilities and correlate this information with production requirements. This correlation of information should allow the machines themselves to confirm production readiness.

This definition is used for the exchange of information between machines, which can be complimented with each machine communicating current resource counts. Even though this does spread the information of the network across the machines on the network, it allows any one machine to communicate to the central server for production readiness and status.

2.6.2.2 Decentralisation with Information Appropriateness

The relationship of decentralisation with appropriate information considers network identification as its critical component. Only through specific information being communicated with intent to destinations, is a network able to have decentralised and appropriate information. This relationship of network identification enables machines to react to certain events by communicating and collecting information from required machines. This information can then be enacted on by setting production limits, communicating change, updating schedules or notifying about delays. This allows machines to become reactive to events. Though coding instructions would still need to be created to handle the events, this again has the information available at an application layer of communication, allowing firmware packages to handle these events with minimal implementation effort.

2.6.2.3 Decentralisation with Distribution

Decentralisation is understood with distribution of information through the use of timely updates. When messages are transmitted at regularly classified intervals, information is guaranteed to be consistent in the network. Machines can then decide on the updated information time needed to collect information and process decisions. These timely updates are realised by a dedicated watchdog process of an application in a machine's programming. These watchdog timers could be activated at the update of dedicated machine memory, load memory or dedicated SMP frame, such as machine status. The timers will then force messages to be sent as updates to all other machines with the PUT method.

2.6.2.4 *Decentralisation with Network Hierarchy*

Decentralisation is recognised with a network hierarchy using classification. Classification of a machine's duties within the network hierarchy will depend on the direct access of information the machine has access to. Whereas all machines within the network hierarchy would be able to communicate and handle general tasks, the machines themselves will be able to decide on tasks to be calculated and thus communicated. Although this means a centralised server, such as the one used within this study, would be able to solely request information from one SMU, each computational task could be divided amongst appropriately placed SMUs. This could effectively introduce parallelism into the network for computation and resource collection as the network expands.

2.6.2.5 *Contextual Awareness with Information Appropriateness*

Contextual awareness is fulfilled through appropriate information by the addition of predictability of a machine. When machines are contextually aware with the network of connected devices around them and are updated with appropriate information, machines will be able to create predictions or set the direction of manufacturing. This can be done by collecting the availability of resources across the SMUs, to identify run-out. Alternatively, SMUs could process the production output of each SMU. In both cases, SMUs would be able to calculate and set a uniformed production rate across the manufacturing line to reduce bottlenecks.

2.6.2.6 *Contextual Awareness with Distribution*

Contextual awareness is understood with the distribution of information through the ability of network accessibility. With machines naturally having the ability to communicate within a network, each machine would be required to hold and update a network map. Similar to a router's traceroute, each machine would require the ability to create a traceroute of the network to ensure information is updated and distributed across the network. This traceroute would also require allowing for multiple hop timings to compensate for non-direct connections to satisfy decentralised, redundant and single path access networks.

2.6.2.7 Contextual Awareness with Network Hierarchy

Contextual awareness and a network hierarchy are relatable through machine agreement. When machines are contextually aware of each other and need to be assigned to a hierarchy, all machines need to agree to the hierarchy. To still satisfy the other core components of the architecture, this hierarchy is both in terms of machine priority of task and availability, where a task is assigned to SMUs based on the proximity of that SMU to the information. A priority of calculation can then be assigned to the closest SMU to collect and calculate information. This can be completed as a static variable in the network. For example, calculating the number of bottles filled, and not capped, is firstly assigned to SMU 1 then SMU 2 in the case study. A dynamic variable of machine network load can then be assigned by assessing the amount of calculation and transfer of information by each SMU. If this load exceeds certain set points, SMU 2 can take over the duties to calculate and update the amount of filled bottles in the network. SMU 2 can then also update the network server and all other SMUs, with a timestamp serving as last updated time parameter for the network not to repeat calculations.

2.6.2.8 Information Appropriateness with Distribution

Appropriate information and distribution of information are gathered by decision making. If information across a network is appropriately updated and distributed in order, machines should have the information necessary to reactively respond with decision or react off triggers to production stages. With information about resource availability being distributed throughout a network, SMUs themselves can individually calculate, communicate and agree upon a set production speed. This should allow for continual uniformity within a manufacturing line, simplifying production root cause investigations, and allow for easy communication of errors.

2.6.2.9 Information Appropriateness with Network Hierarchy

Information appropriateness and network hierarchies are achieved using task scheduling. Information across a network will need to be scheduled for update. This will allow for SMUs across the network to update other SMUs with information and follow on with a task scheduled, either by the communicating SMU externally or receiving SMU internally. For updates on the production status within the case study, SMU 1 could update SMU 2 about the next bottle height to receive. SMU 1 could then instruct SMU 2 to cap the incoming bottle and prepare for the bottle height, or SMU 2 could arrive at this conclusion on its own, depending on the attached sensors and description of the network hierarchy.

2.6.2.10 Distribution with Network Hierarchy

Distributed networks are met with hierarchical networks by reduced redundancy. This reduced redundancy is stipulated through all levels of the communication layers. It is important to update SMUs on a network about ongoing production statuses, but might not be needed to store historical production information if the information is not analysed. The central server in terms of the case study, might be better equipped to store and analyse production history. Alternatively, resultants of production history that are calculated and communicated from each SMU, can be stored on the central server, instead of raw sensor data. Rather this raw sensor data can be stored daily on the SMU and transferred after production runs. With each of the relationships within the architecture seamlessly increasing the amount of communication required, this relationship aims to control this information with the removal of unnecessary overhead when not analysing and rather being replaced with intentional actions.

2.6.3 Enabling Attributes to the Communication Architecture

The aspects discussed within this section are complimentary to the communication architecture, to show input and output requirements to other communication levels or inclusion technologies of the future. Though they are not prevalent in the communication architecture itself, the discussion of these can show the realisation of the architecture in practice and considerations for iteration of the communication architecture.

2.6.3.1 Network Communication Components

2.6.3.1.1 Networkability

Regarding the aforementioned core components, all of them are an evolution of networkability [46]. Without machines able to communicate within a network, all core components would never be realised. It is however redundant for networkability to be involved in a communication architecture, as networkability is underpinned as being able to build communication architectures on top of networks and flows of information.

2.6.3.1.2 Integratability

Machines should be able to be integrated within both a network and a manufacturing line. For machines, to be placed onto a network or into a manufacturing line without the ability to be moved or replaced creates for rigid structures. These rigid structures are often hyper specialised and therefore architectures are difficult to apply within them. Rather, these hyper specialised systems will act as plugins to an architecture and never be fully incorporated. For machines to therefore be integrated brings another degree of difficulty to handle dynamic systems. Whilst this is an important characteristic for machines to possess, this study views integratability [47] as a circumstance of influence to control and not to be incorporated into a communication architecture.

2.6.3.2 Product Communication Component

2.6.3.2.1 Smart Products

Smart products (SPs) [48] can embed and communicate information. This can be achieved by using passive and detached information methods such as writable RFID tags. These products will then be able to carry information along the manufacturing line and throughout the logistic chain, to the end user.

Information captured by SPs can be of mutually exclusive events from production information captured within the production line. This information presented uniquely at the application layer is also beneficial. For instance, with production date information being able to be stored on the SPs, this information can be read by machines in the manufacturing line to aid with data analytics or destination routes. However, at a customer interface, this production date can be read in and calculated for a best before date. The inclusion of information to the manufacturing line, however, should allow for inclusion from the communication architecture.

Instead of using SPs to replace requirements and responsibilities in the communication architecture, SPs should be used to compliment the architecture alongside its core components and relationships. As SPs are integrated into the communication architecture, adherence to the architecture itself avails for the responsibilities of the SPs. Information should be distributed, allow for a network hierarchy and be related through reduced redundancy. SPs would not need to carry each production stage time completed on them, but rather, one could include information such as quality and task scheduling with appropriate information. In the case of the water bottling plant, SPs could capture the quality of water placed within the bottle, measured at different intervals of the production process, and changes in this quality can be narrowed down and still presented to the customer. SPs could also hold customer requirements directly within themselves, and when arriving to certain machines, could invoke certain processes, such as changes in colour.

2.6.3.3 IoT Component

2.6.3.3.1 IoT Sensing

The rapid expansion of IoT sensors [49] and technology has allowed for cost effective and uniquely divided resources being widely available. IoT sensors often come as single solution devices that can be expanded into micro-machines to collect production information and, at times, be involved with the manufacturing process.

IoT sensors therefore neatly include themselves with distributed and appropriate information sharing that allows SMUs to create decisions off IoT sensors or react to IoT sensor decisions. This decision making is directly tied to the IoT function. Some IoT sensors perform more of a passive role within the manufacturing line, used as data logging points, either for environmental or in-production factors. Some IoT sensors could be more directly involved, utilising their resources to analyse real-time production events, as alarm triggers or information sharing for data analytics. Regardless of the function of the IoT sensors, the classification of this technology group should adhere to a self-powered, disconnected system that is able to either communicate to a direct SMU or the network as a whole. This requirement would naturally adhere to the network accessibility and identification of the architecture, with its data packet and communication protocol set up to take full advantage of communicated messages without large software overhead requirements, as these software developments would cause their distributed systems to be isolated and not distributed.

2.6.3.4 Collaborative Decision-Making Component

Collaborative decision making [50] involves itself in the manufacturing process by including all role-players of the manufacturing line. Whereas a large majority of tasks can be automated, some tasks would require the intervention of human decisions for un-programmed and non-forecasted events. These possibilities are broken down into two sections, being agility stemming from the SMU characteristics, and human interaction, forming part of the ever-evolving I4.0 revolution.

2.6.3.4.1 Agility – Modularity

Although machines in the communication architecture should be able to identify machine neighbours and manufacturing capabilities, the physical act of creating modular manufacturing lines requires hardware intervention. Creating individual modular [51] and expansive machines would similarly require hardware standardisation. These concerns are a requirement of I4.0, yet considerations should be taken for the expansion and configurability of the communication network in the architecture.

For these factors, the use of a communication protocol to cover machine and network identification is crucial. Regarding the communication protocol and architecture, this is handled with its components of decentralisation to contextual awareness and appropriate information. Compliance to these core components and relationships could therefore allow for expansive and reconfigurable manufacturing lines and their associated networks.

2.6.3.4.2 Human interaction – Intervention

Considering the natural dynamic nature of manufacturing lines, human intervention is inevitable at times. Whether to maintain or upgrade sections of the manufacturing line, I4.0 requires humans to be present at times. Human intervention [52] on a manufacturing line is often required at less efficient times due to the nature of their intervention. Therefore, careful consideration should be taken to aid human operators.

Humans within the loop of the manufacturing line should be able to observe and diagnose manufacturing lines virtually from anywhere. Though some instances would require physical and on-site intervention, planning and preparation can always be done to reduce time and continue production. This requires human operators to have the ability to access the information of the network at any point.

Fortunately, this is possible through the network access of the communication architecture, where contextually aware machines with distributed information would be able to act as service points. These service points would be able to fetch information between machines, using the SMP GET method, from any connected SMU.

Alternatively, SMUs might not need to fetch the latest information, as their timely update relationship using decentralised and distributed core components allows machines to continually update information across the network. This update time might make it possible to more easily catch production upsets and early enough to reduce maintenance times.

These abilities to access information from any point within the server thus require a method of collecting information from any network point. This might not be actively running throughout a manufacturing line, yet the reduced redundancy task for distributed and network hierarchy components would allow for these information collection methods.

2.6.3.5 Digital Twin Component

Digital twins (DTs), when introduced to a manufacturing line [53], would need to be able to collect all vital information related to production in order to command the manufacturing line for improved performance. These commands should be decision based and enacted from the manufacturing information of the production line. Due to this, DTs are required to satisfy the complimenting SMU characteristics of cloud-manufacturing and data analytics.

2.6.3.5.1 Cloud-Manufacturing

Within the characteristic of cloud-manufacturing [54], the control around cloud computing is best satisfied with a software architecture, where event driven or micro service architectures are best poised to handle manufacturing communication. This leaves the requirement of real-time communication to be solved which is well suited to be handled within a communication architecture.

2.6.3.5.1.1 Real-Time Communication

Real-time communication [55] within the cloud manufacturing environment calls for the requirement for information to be routinely updated in methods associated within the communication network. Whereas these requirements have been framed as real-time communication, the criteria for real-time differs at network level and machine level. Although the machine level real-time communication restraint requires a transmission time of <250ms, the real-time requirement of digital twins can be more akin to live information. This live information with no strict transmission time enforced within the community of I4.0, can be classified to be between 1-5 seconds, with update intervals being between 2-10 minutes. This criterion is seen by this study to be satisfactory with the ability to collect information of DT models, act on the responsibilities and decisions of DTs and reduce unnecessary communication bandwidth through the hierarchical split. This timing requirement also compliments the memory map used by each SMU, to store temporary data of ongoing manufacturing. As the information from SMU to DT is communicated, the previous production information on the SMU can be cleared. It is advised by this study to only clear the oldest portions of the production data and not the entire memory. This could prove useful for the on-board data analytics performed by SMUs, to keep running averages in check and without introducing noise into the data.

2.6.3.5.2 Data Analytics

DTs' main role within the manufacturing scene would be to provide high-level data analytics [56] for the benefit of the manufacturing process. This can be achieved through data visualisation for characterisation, simulation of events, and provide automated responses to manufacturing events. Even as DTs evolve their own set of parameters and standardisation with addition of other technologies, such as AI, it is important to evaluate their requirements for the communication architecture to be inclusive and realise I4.0.

2.6.3.5.2.1 Data Visualisation

The bulk of data analytics can be performed at data visualisation [57]. Though most data sets can be automated for sorting, averaging and classification, these automations are the responsibility and outputs of data visualisation in I4.0. Visualised data from DTs can then be evaluated for statistical process control to update and change any parameters of the manufacturing line. This visualisation of data would be improved with the classification of information sent by the SMUs and the network, requiring the DT to access information from the decentralised network and network hierarchy.

2.6.3.5.2.2 Simulation

Simulation [58] of the manufacturing line by the DTs aims for predictable outcomes to be communicated within the DT and further on to the manufacturing line and network. This predictability requirement for simulation would require the DT to access the information of the manufacturing network that would constitute its contextual awareness and appropriate information. This access to information should allow for dedicated simulated environments and tasks to run when collected information is transferred to them. The expansion of the manufacturing line and its complexity would come with the expansion of simulations. Therefore, the use of contextual and appropriate information would allow for the reduction of overhead communication and filter messages to feed into simulations. This would satisfy not only the simulation requirement but be seen to compliment the real-time communication aspect within the cloud manufacturing component of DTs.

2.6.3.5.2.3 Automated Response

Automated responses [59] from DTs would require an output function to communicate task scheduling either before or during production. This output would therefore require the access to the appropriate information of the machine, as well as understanding the network hierarchy to best stipulate tasks. Automated responses can be taken from the effects of data visualisation and simulation to predict manufacturing events at a higher level. Although SMUs in the manufacturing line would be focused on completing and optimising daily production runs, DTs would try to forecast increases in orders around external events, to better group and distribute tasks.

The enabling attributes to the communication architecture to accommodate other I4.0 technologies can thus be summarised by the relationship to the core components as in Figure 2-11.

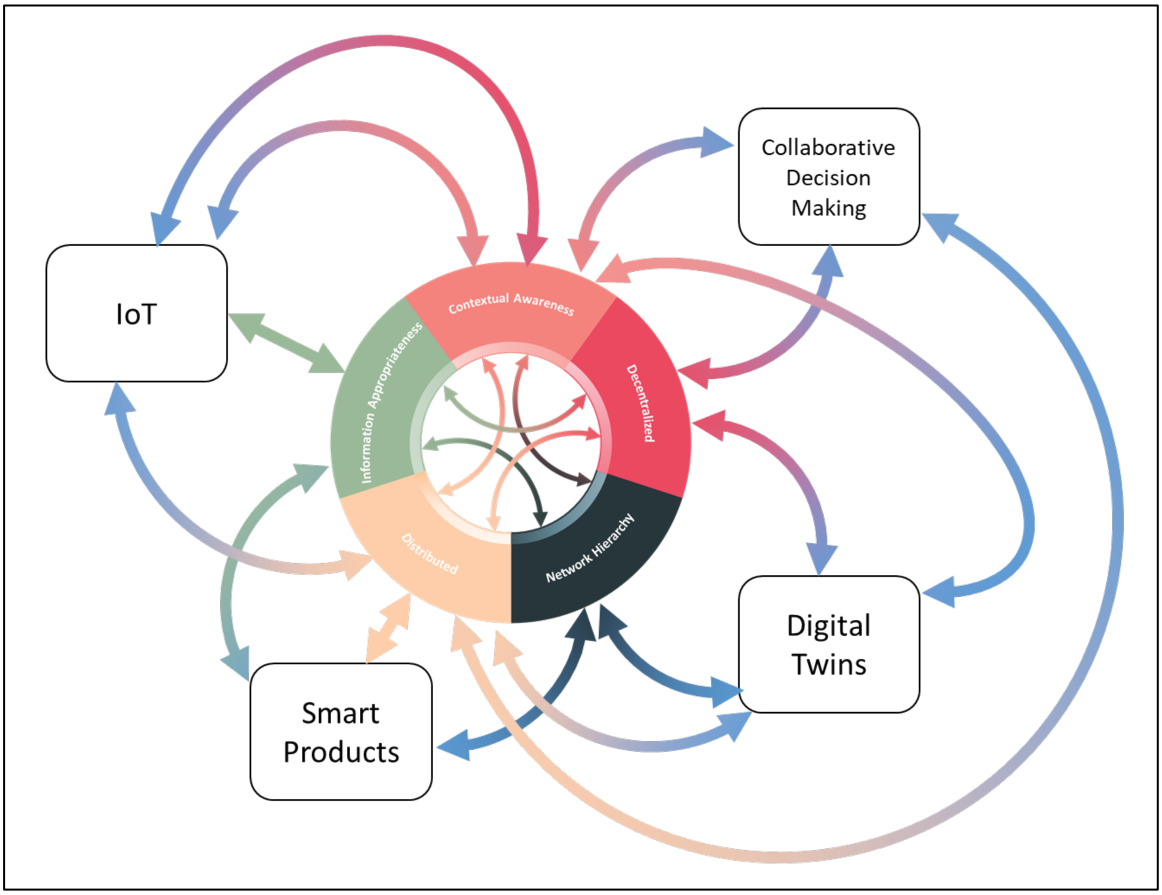


Figure 2-11: Communication architecture with enabling attributes

2.7 Limitations of Current Research

The foregoing literature review shows a vast expanse of knowledge on I4.0 and SM that has nonetheless left research gaps to connected manufacturing setups with best practices. As additional research is conducted in SM concepts and implementations, the advancements made within the research studies are often narrow in the context of the field as a whole, which makes it difficult to include best practices in other research domains.

This current state of research progress leaves SM and communication architectures without feedback loops [60] of their improvement life-cycle and offers no tangible support to incorporate architectures from these narrow breakthroughs into future implementations and develop the ongoing requirements.

2.7.1 Limitations of Smart Manufacturing

SM has seen significant advances in development for its requirements and realisation through SMUs. These SMUs have their own requirements to constitute how intelligence is enabled in the manufacturing space and separate itself from predecessors. There are, however, gaps within this research that illustrate the lack of expansion and innovation [61] on how these SMUs will communicate within their network.

Research has identified that using existing communication methods in a SM setup can cause delays in communication, which manifest as bottlenecks during production [62]. The ability of SMUs to accommodate a vast variety of communication methods does avail itself to incorporating many expanding technologies, such as edge computing; however, greater standardisation is needed for manufacturing communication standards. These standards are best maintained through an architecture.

2.7.2 Limitations of Communication Architectures

Ongoing research within the field of I4.0 and SM is expanding the knowledge and understanding of their implementations [63], nevertheless, little research is being done on the expansion of architectures in these fields [64]. Many researchers are adopting historical architectures to create and operate SM systems.

This study poses the question as to the suitability and currency of existing architectures being applicable to new manufacturing requirements. Further, this study envisages a natural similarity between SM communication with the organisation and event handling structures of a new architecture type, being a communication architecture.

Current implementations of SM systems [65] employ the use of software architectures to handle event communication of machines to analyse, act and communicate on production information. Whilst these implementations do achieve their goals and adhere to the requirements of SM, very few studies quantify the efficiency of these approaches.

In addition, even fewer studies discuss whether the implementation of this software architecture to handle data communication events is the correct approach [66]. Whereas the complexity of SM is being further realised with each implementation [67], this study shows that the organisation and responsibility assignment of communication by machines should be handled individually to realise a part of intelligence for machines.

2.7.3 Requirements of a Communication Architecture

With the characteristics outlined in section 2.2.6, observations into multiple pipelines emerge around the organisational construct for data simplicity [68, 69]. This construct has been naturally attuned with hardware architectures to allow software architectures the data throughput required for efficient real-time communication in I4.0.

However, creating a greater throughput of data to add I4.0 at higher software layers, and possibly away from the machines on the manufacturing line, is a reckless approach [70, 71]. A review of the I4.0 and SM requirements shows a possible approach for data organisation and structure to be accessed at machine level, and selectively chosen to meet the characteristic of demand at machine level to enable SM. This approach identifies gaps with current research [72, 73, 74] where communication architectures are undeveloped.

This study proposes for intelligent control to be incorporated into SMUs, as communication of information between machines and between other network nodes can allow for decisions to be enacted without sizable hardware and software considerations. Therefore, a focus on non-hardware and non-software related characteristics can be explored for creation of a communication architecture.

This study thus sets forward to create a communication architecture that naturally attunes to the SM characteristics, to allow for more efficient production in a manufacturing line by enabling said line to create intelligent decisions on demand. This communication architecture will require an evaluation for the creation of its aspects, where these aspects can be similar, inherent and/or assumed to be incorporated.

These aspects will be broken down into their core components and relationships of the communication architecture, and will further need to be evaluated on their own metrics to quantify the improvement of each aspect on a case study basis and characterise the best-fit requirement to the system. The architecture implementation improvement onto the baseline case study will also need to be evaluated to gauge the general improvement in production efficacy. This is aided using manufacturing scenarios.

Lastly, a holistic overview of the communication architecture implementation will need to be discussed in order to allow for iteration and longevity to be involved along with the rapid life-cycle of I4.0.

Chapter 3 : Methodology

3.1 Introduction

This chapter explores the reasoning and inclusion of core components and their relationships to the communication architecture. Similar to how methods, classes and areas of optimisation/implementation are considerations for software architectures, so too must communication architectures be defined.

In section 2.3.3.3, the definition of a communication protocol remains open-ended, and this chapter evaluates a communication architecture encompassing communication protocols, communication flow and communication responsibilities for consideration and classification. This chapter then guides the implementation of the communication architecture implementation in Simulink and develops the metrics for evaluation.

Manufacturing setups are however complex with their unique distribution of resources and tasks and, as such, are often separated with interlinking communication protocols. The developed communication architecture must then consider the distribution of the network. This distribution must be interchangeable based on the communication protocol used. It is however the responsibility of a communication protocol, or additional methods to be compliant to the communication architecture.

3.2 Smart Manufacturing Case Study

In this study, a case study will be used as a benchmark to evaluate the communication architecture. The case study network layout is detailed in Figure 3-1, where a cloud server collects water bottle orders from customers. Customers can order a mix between 300ml and 500ml water bottles and set their required delivery date. Orders are limited to a maximum of 100 of each bottle. For evaluation of the Simulink model, multiple simulations are run on each manufacturing scenario totalling to 10 total evaluations of each manufacturing scenario. The average for each scenario will then be taken for review in the results section.

The cloud server communicates all customer orders to a central manufacturing server through an IPv4 connection from where the production information is communicated to each SMU through an OPC connection. The SMUs at each station are responsible for filling, capping and packaging the water bottle orders for a day. Customer orders and the order in which bottles are fed into the manufacturing line can be varied. The SMUs can communicate with each other through the developed SMP covered in section 2.2.8.

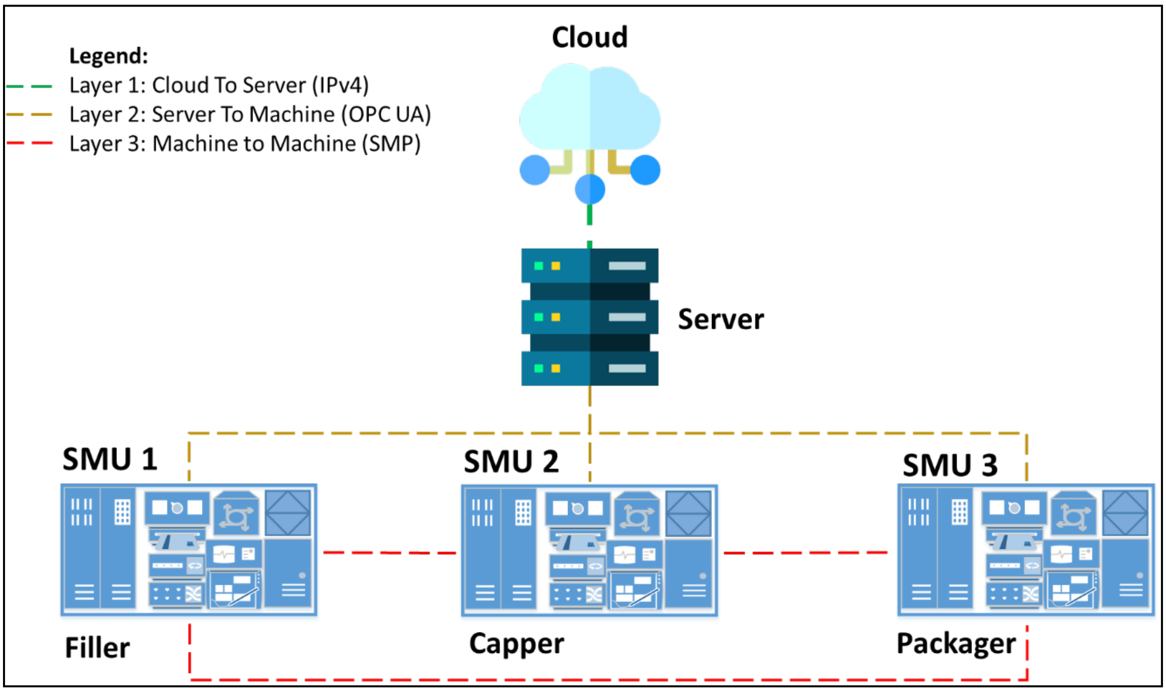


Figure 3-1: SM case study

3.3 Implementation of Case Study

3.3.1 Production Scenarios

For the study, it becomes practical to create a comparison benchmark of the performance with the communication architecture compared to the normal operation of a manufacturing line. This comparison can be evaluated within the SM case study by the inspection of improvement through manufacturing scenarios. The baseline manufacturing performance was taken from real world experiments of a laboratory setup of a replica manufacturing line. The baseline manufacturing setup was then replicated in MATLAB Simulink to allow for the creation of different manufacturing scenarios and the communication architecture.

These production scenarios are required to evaluate the production process of the manufacturing line during normal, chaotic and random operations. Important metrics to monitor as a benchmark and effect from the communication architecture should include production rates, production monitoring, resource monitoring, communication between machines about production events, communication between machines about product information and bottleneck areas.

This study therefore evaluates four production scenarios for the baseline and communication architecture manufacturing implementations. These production scenarios are summarised as:

1. No organised structure (FIFO) of bottle orders.
2. Optimised order structure by required delivery date.
3. Optimised order structure by bottle type.
4. Random bottle order feed in.

These productions scenarios can be simulated in Simulink to allow for finer overview within the production data and discussion of the information flow between machines.

3.3.2 Communication Protocol

The SMP is implemented in each SMU of the Simulink model as a core communication model. The basis of the SMP implemented in Simulink can be seen in Figure 3-2. The housing of each communication protocol block of the SMUs is complimented with their physical operation. These application blocks were referenced and measured against a physical model implementation of the SMU line based on a complimentary study [75].

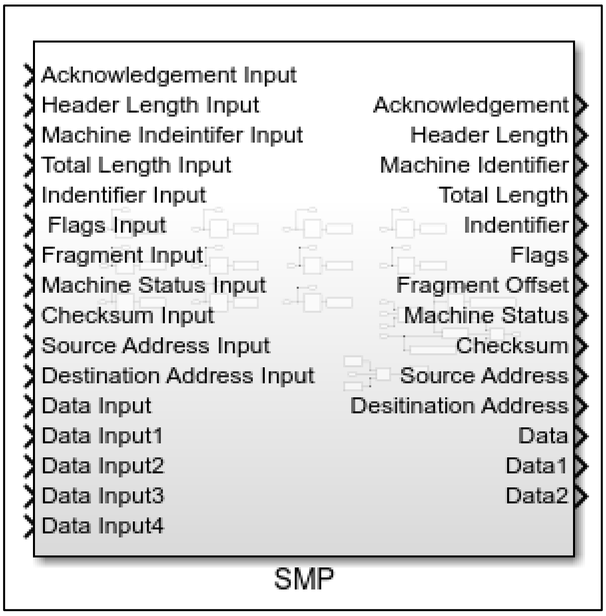


Figure 3-2: SMP Simulink code block

For the baseline manufacturing setup, the SMUs are equipped and programmed to act autonomously on their tasks, communicate with other SMUs and the central server, and optimise their performance with only their attached machinery. This is highlighted with SMU 1, where their next feed of water bottles in order can be predicted by recognising previous patterns of fed-in bottles and optimising the prepared capping height for the matched bottle with 75% certainty, or a middle location in the event of uncertainty.

Admittedly, greater optimisation between SMUs could exist with the parsing of information. This study, however, still sets out to prove whether the organisation of this flow of information can result in even better performance improvements and the availability of greater intelligence of the SMUs.

The lack of specific production information transfer between SMUs, and any machine for that matter, is a direct cause of efficiency loss even at highly optimised local stations as they are stepped back by inefficiencies during the transfer of production from one station to the next.

This step back is caused in two parts: Firstly, by creating bottlenecks of highly efficient systems being passed along by less efficient transfer systems and, secondly, by data-gathering checks not being implemented at these points of local production transfer. These implications would, in most cases, lead to the efficiency gains of highly optimised systems being plateaued at the locations that are uncontrolled.

3.3.3 SMU Simulation

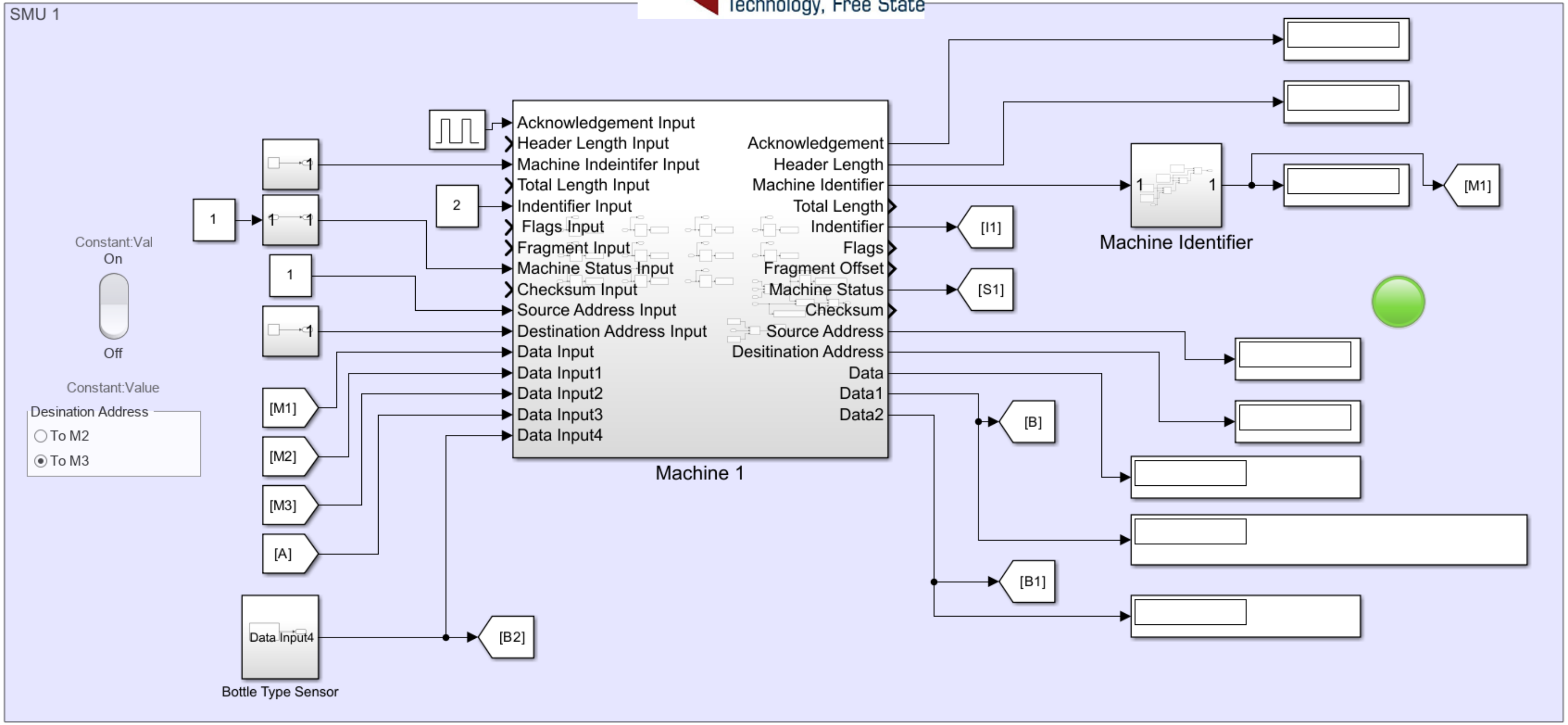
The SMUs are assigned according to their respective tasks, as described in Section 3.2. Figure 3-3 shows all communication links between the SMUs that are handled through the *Status*, *Data* and *Flag* fields. These fields may be updated to individual or all SMUs with the use of the PUSH method of the SMP or requested from any number of the SMUs on the network with a GET method.

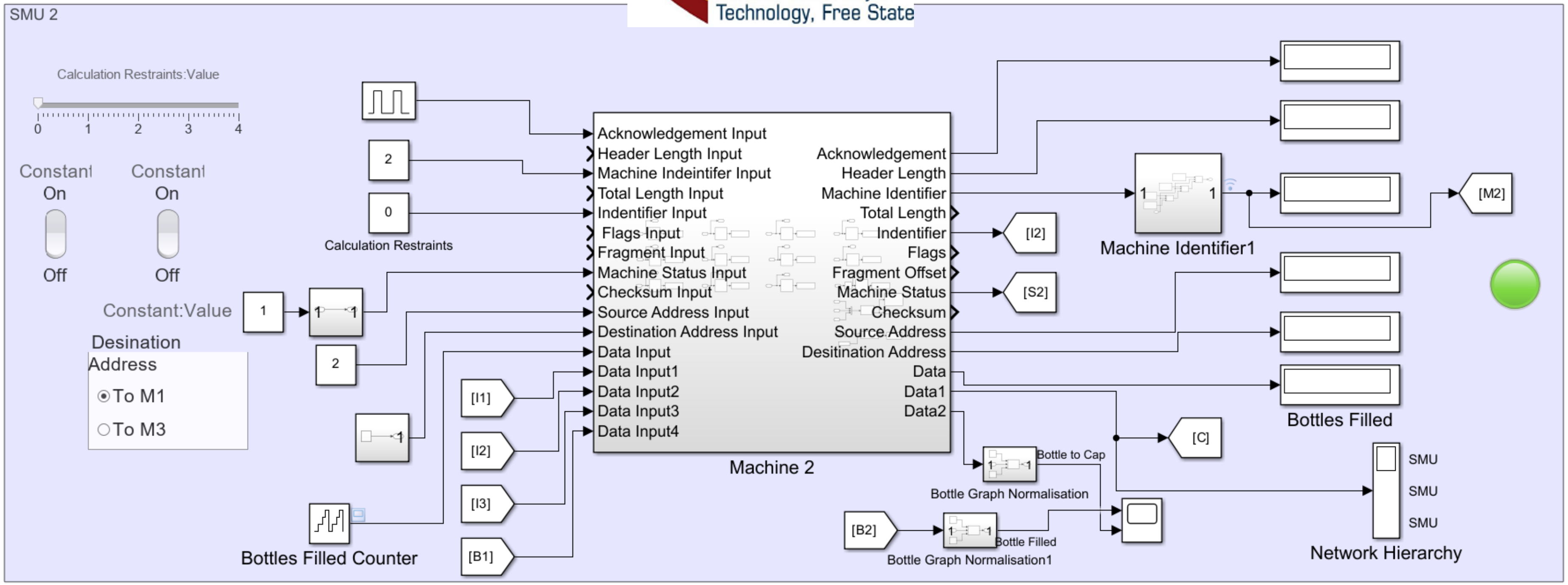
Production orders are collected from the cloud interface, diagnosed by the central server for the preferred production scenario, corroborated with a manufacturing status from the SMUs, and communicated to each machine on the network for production.

Each SMUs SMP code block has been assigned the information required for production, passing along information in the data field of the SMP. This information includes the current product and next product particulars that includes; a unique product ID, product size (300ml or 500ml) and final customer destination assignment.

The SMP data block also handles machine specific information about each SMU, such as its IP address, machine type identifier and machine status. Each SMU is the also programmed with its applicable manufacturing task.

Production assignment details of the current task are handled by the central server. This allows each SMU to receive the entire production requirement, while SMU hierarchy are agreed by the SMU and displayed by SMU 2. SMU 2 also dually handles the current computation and network load handled by each of the machines, which is then able to divide additional tasks to the network based on the least strained machine. While this information is collected by the SMP, it is only through the programmed scheme of the communication architecture that this information becomes useful and enacted upon.


SMU 1
Constant:Val
On
Off
Constant:Value
Desination Address
○ To M2
● To M3
1
1
1
1
1
2
1
1
[M1]
[M2]
[M3]
[A]
Data Input4
Bottle Type Sensor
[B2]
Acknowledgement Input
Header Length Input
Machine Indeintifer Input
Total Length Input
Indentifier Input
Flags Input
Fragment Input
Machine Status Input
Checksum Input
Source Address Input
Destination Address Input
Data Input
Data Input1
Data Input2
Data Input3
Data Input4
Acknowledgement
Header Length
Machine Identifier
Total Length
Indentifier
Flags
Fragment Offset
Machine Status
Checksum
Source Address
Desitination Address
Data
Data1
Data2
Machine 1
[I1]
[S1]
[B]
[B1]
1
1
Machine Identifier
[M1]


SMU 2
Calculation Restraints:Value
0
1
2
3
4
Constant
On
Off
Constant
On
Off
Constant:Value
Desination Address
◉To M1
○To M3
1
2
0
Calculation Restraints
2
Bottles Filled Counter
[I1]
[I2]
[I3]
[B1]
Acknowledgement Input
Header Length Input
Machine Indeintifer Input
Total Length Input
Indentifier Input
Flags Input
Fragment Input
Machine Status Input
Checksum Input
Source Address Input
Destination Address Input
Data Input
Data Input1
Data Input2
Data Input3
Data Input4
Acknowledgement
Header Length
Machine Identifier
Total Length
Indentifier
Flags
Fragment Offset
Machine Status
Checksum
Source Address
Desitination Address
Data
Data1
Data2
Machine 2
[I2]
[S2]
Machine Identifier1
[M2]
Bottles Filled
[C]
Bottle to Cap
Bottle Graph Normalisation
[B2]
Bottle Filled
Bottle Graph Normalisation1
SMU
SMU
SMU
Network Hierarchy

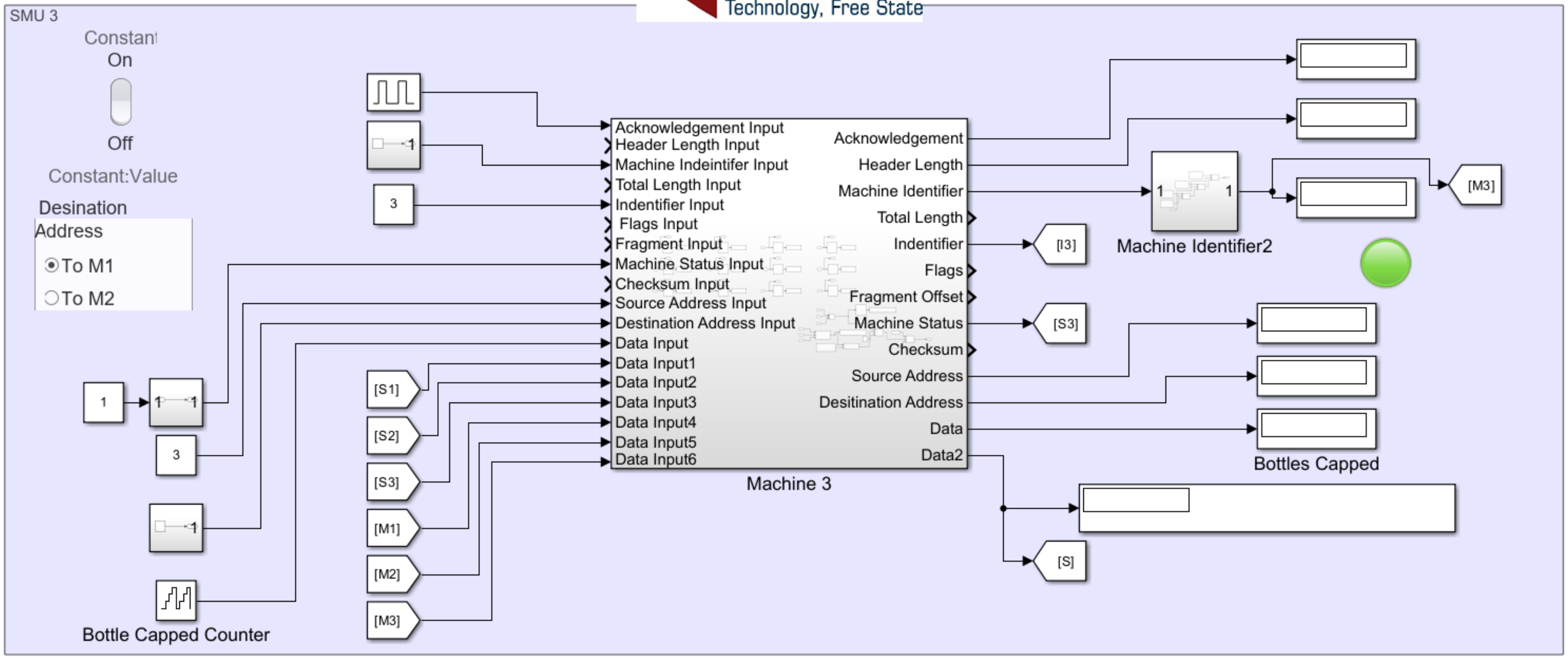


Figure 3-3: SMU communication with the SMP

3.3.4 Central Server

The central server can communicate with the SMUs using an OPC-UA connection that, along with the responsibility of organising production start information, acts as a DT of the SMUs, allowing for production information to be fed back into the system for evaluation of optimisation and use of real-time data. The central server Simulink controls can be seen in Figure 3-4.

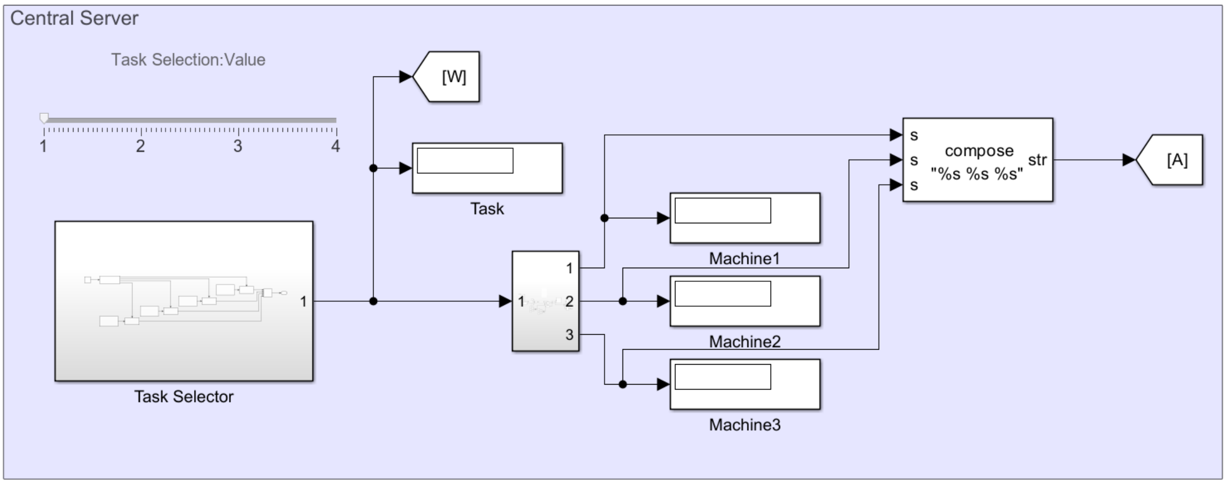


Figure 3-4: Central server oversight

The central server Simulink code involves a diagnostic of total number of unique bottle orders requested, along with their required delivery date. At its current implementation, a dedicated control loop handles the optimisation of production delivery to fulfil the production scenarios. This organisation and scheduling structure can be seen in Figure 3-5.

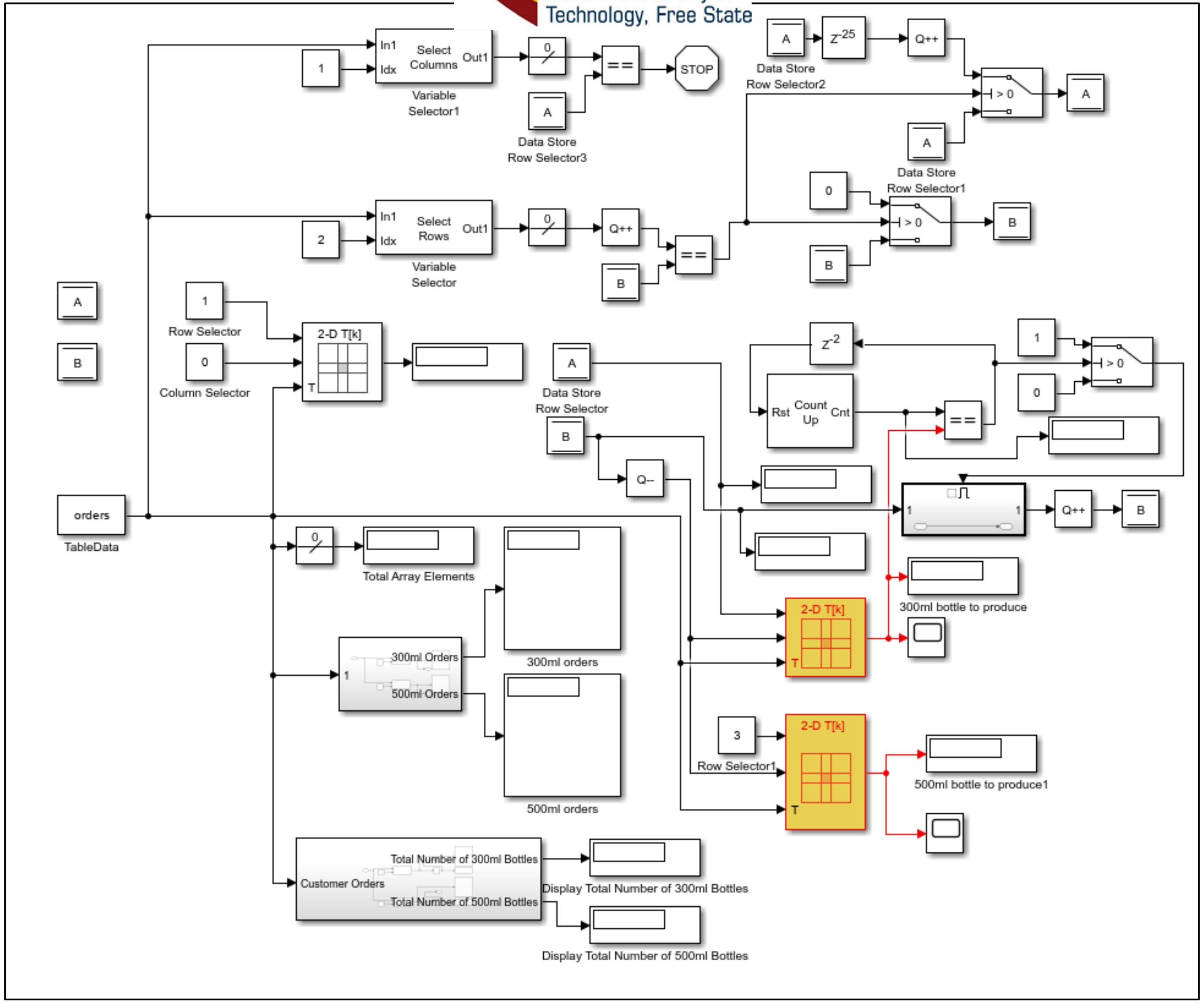


Figure 3-5: Central server Simulink code

3.3.5 Communication Architecture Traits

Within the control system of the SMUs, the SMP is implemented as a single aspect. Although the SMP does allow for general communication between SMUs, enabling specific functionality to collect, send, organise and act upon shared data enables the full use of the SMP. With the SMP aspects being included in either a hardware or software space, the full communication architecture is realised. The SMP Simulink aspect can be seen in Figure 3-6.

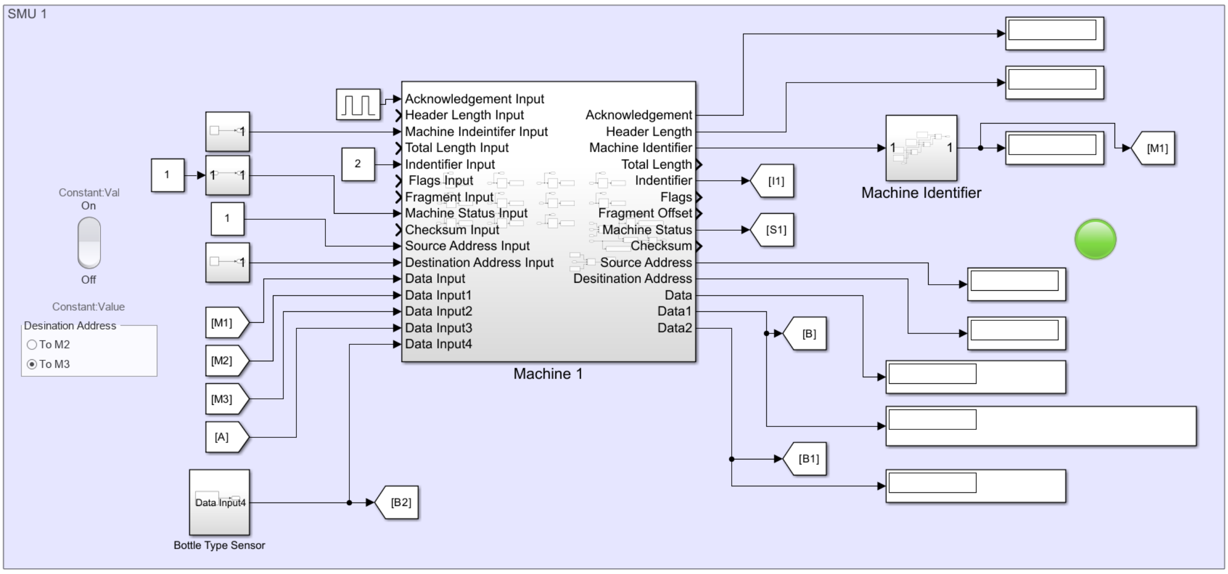


Figure 3-6: SMU 1 (filler) Simulink code

The SMP allows for current event information to be communicated between SMUs to continue the cycle of information flow and actionable decisions based on current production events. This allows the SMUs to garner a sense of intelligence, not only individually but amongst machines on the network.

Manual interventions to production and resource statuses are accommodated in the Simulink code, as seen in Figure 3-7; however, these production events also trigger automatically with predefined warnings and alarms on the SMUs. These triggers include a shortage of resources, a measured slow-down in production speed, and machine ability to continue production.

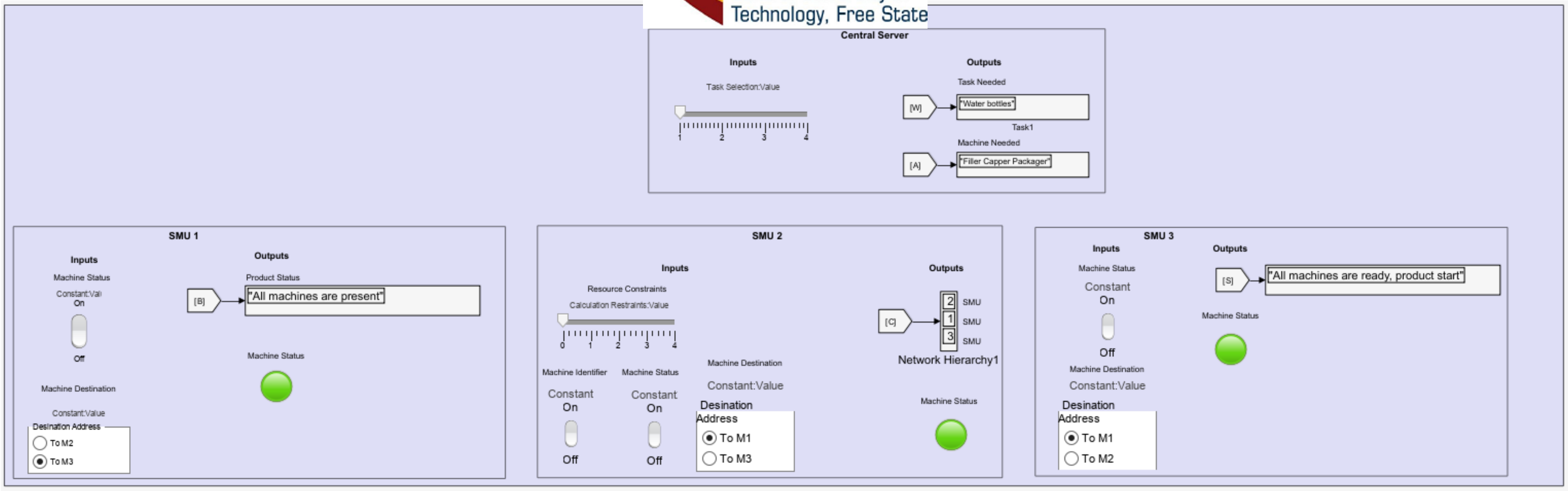
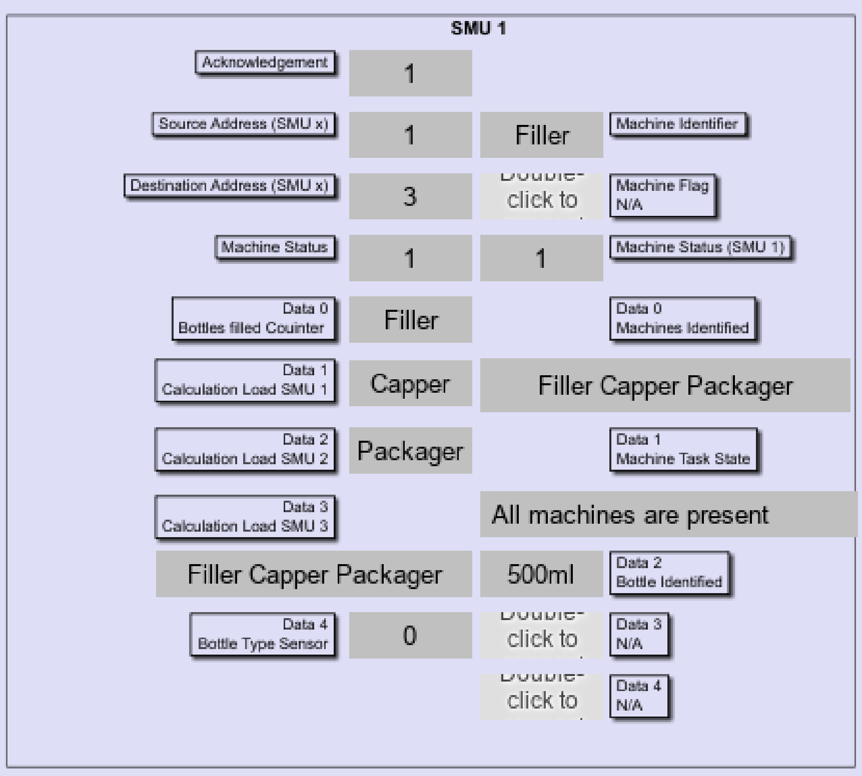


Figure 3-7: SMU manual intervention

Whereas the machines are manually encoded with trigger and alarm levels per use-case, the communication and immediate access to these triggers at a protocol level is also necessary for each machine to create decisions based off the current production environment and within a decentralised manner. This ability is granted to each machine with the control and access to the *machine status* and *flags* fields of the SMP, where these fields can be customised with decodable headers and applicable statuses.

The exact performance and optimisation these intelligent decisions have on the manufacturing line depend on the production scenario. This availability shows how the communication architecture can be one method to enable this functionality. The flow of this information can be seen in Figure 3-8 to Figure 3-10.



SMU 1

Acknowledgement: 1
Source Address (SMU x): 1
Destination Address (SMU x): 3
Machine Status: 1
Data 0 Bottles filled Counter: Filler
Data 1 Calculation Load SMU 1: Capper
Data 2 Calculation Load SMU 2: Packager
Data 3 Calculation Load SMU 3: Filler Capper Packager
Data 4 Bottle Type Sensor: 0

Machine Identifier: Filler
Machine Flag N/A: Double-click to
Machine Status (SMU 1): 1
Data 0 Machines Identified: Filler Capper Packager
Data 1 Machine Task State: All machines are present
Data 2 Bottle Identified: 500ml
Data 3 N/A: Double-click to
Data 4 N/A: Double-click to

Figure 3-8: SMU 1 production monitoring events

SMU 1 is designated by the central server for collection and identification of the manufacturing line machines, which can be seen in Figure 3-8, by the identification of the third data load reading "Filler, Capper and Packager". This information is then fed back to the central server for validation of currently attached machines on the production line. This allows both the central server and SMU 1 to confirm if the attached manufacturing line is capable of completing the requested order.

SMU 2 is tasked with calculating any additional network requests that can be sent through from the central server. These network requests could include calculation of current production efficiency, identification of bottlenecks in the manufacturing line, and a prediction of production progress competition. These requests are then scheduled to the least network-taxed SMU at the current point in time. This network load can be seen in the first data field of SMU 2 in Figure 3-9.

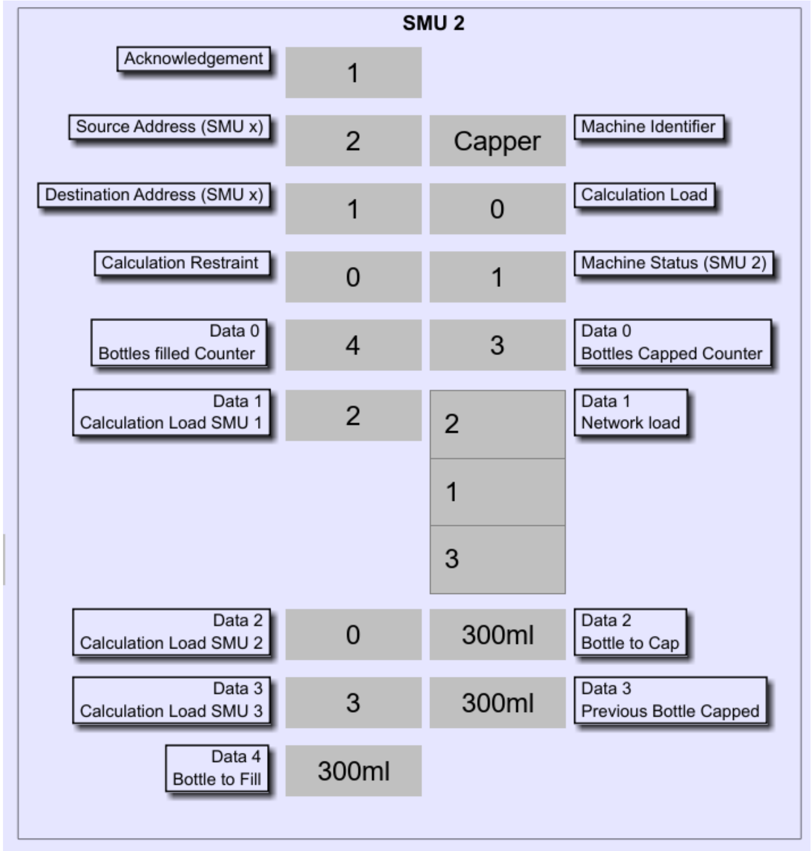


Figure 3-9: SMU 2 production monitoring events

SMU 3 is tasked within its network requirements to determine the real-time confirmation of the production running status seen in its second production status flag of Figure 3-10. This confirmation collects the production readiness capability per machine along with the requested order that is cross-checked against availability of resources and ability to complete the production run by the required deadline.

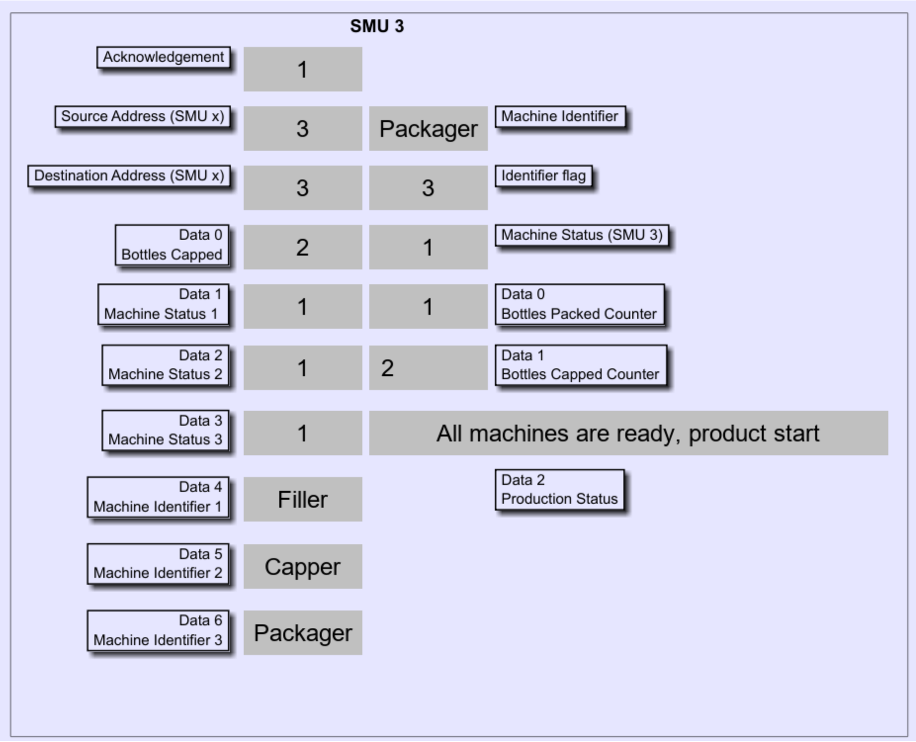


Figure 3-10: SMU 3 production monitoring events

3.4 Architecture Scoring Method

With the inclusion of a communication architecture, an evaluation method for the implementation and improvement is needed. This evaluation method should be relevant to the problems to be solved by including the communication architecture. Further, this evaluation method should serve as a baseline for future improvements and iterations, whilst the scoring method is to be refined itself through data of practical applications and industry needs. Therefore, it is prudent that this study proposes the requirements for evaluation of the communication architecture.

Although many aspects from scoring mechanisms of other architecture types, i.e. hardware and software, were taken as inspiration for this evaluation, the metrics under scrutiny were adopted from the case study and industry understanding. The metrics below are therefore outlined and expanded in subsections 3.4.1 to 3.4.6.

It is important to reiterate that these metrics are assessed at machine communication, or layer 3 of the communication network, as this is where the communication architecture is fully implemented, with preluding layers having requirements to be met in support of the communication architecture. Though some metrics can be evaluated at other layers of the communication network, their standardisation and evaluation are outside the scope of this study.

3.4.1 Responsiveness

A communication architecture should be responsive to communication requests. The responsiveness of the network describe the time required to access data (latency) and the transmission speed of the network, where both metrics would be imperative to meet the real-time communication requirement of machine communication and the ability to communicate swaths of data packets. The responsiveness properties of the communication architecture can be seen in

Table 3-1.

Table 3-1: Responsiveness of the communication architecture

Criteria	Property	Unit
Responsiveness	Time to access data	milliseconds
	Transmission bandwidth	Mb/s

3.4.2 Accuracy

The accuracy of communication is ultimately present in all communication networks. Transmission accuracy has largely been solved and forms part of the communication protocol in use, yet data accuracy within a network can be expanded for better evaluation, especially in a communication architecture. The accuracy with a communication network concerns itself with data degradation, compression loss and data recency. Data recentness refers to the ability of the architecture device storing, access and transmitting information from other devices on the network. The metrics may depend on the communication protocols used, required storage/transmit strategy, communication level implementation and manufacturing scenario. For this study, it is expected to store the last five minutes of production data with updates occurring every ten seconds from the other SMUs. Data degradation refers to the percentage of lost data from the production run/day. Individual devices or machines do not necessarily need to store their own data, but can securely create backups somewhere on the network, such as a central or cloud server while still incurring no degradation to the metric. The metrics of each aspect can be seen in Table 3-2.

Table 3-2: Accuracy of the communication architecture

Criteria	Property	Unit
Accuracy	Data degradation	%/hr
	Compression loss	%
	Data recency	X/3 0/3 – Not reliable at all 1/3 – Updated irregularly with new data 2/3 – Synchronisation with latest data 3/3 – Updated when required

3.4.3 Transmission

The transmission of data packets between machine network nodes is considered a critical aspect. Data packet sizes allow for reduced latency and overhead, and although all communication protocols are not equal with their data size limit, it should be considered when designing how machines communicate information, and their limits. Therefore, the transmission of information from machines should evaluate their data size limits, availability of data, traceability of data, and networkability. These metrics are expressed in Table 3-3.

Table 3-3: Transmission of the communication architecture

Criteria	Property	Unit
Transmission	Data size limit	Mb
	Availability	%
	Traceability	X/3 0/3 – Data is not traceable 1/3 – Data is traced from its source 2/3 – Data timestamps and critical information is available 3/3 – Timestamps, data fields, and an ID of the action is available
	Networkability	X/3 0/3 – No networkability at all 1/3 – Communication within LAN 2/3 – Communication within LAN and with communication outside of communication level 3/3 – Decentralised communication within LAN and communication outside of communication level

3.4.4 Design Time

Design time of manufacturing lines is often strenuous and is notorious for running over monetary and timely budgets. At its core, architectures should assist with applying known metrics to guide developers, reduce design times and mitigate complexities before they arise. A communication architecture should therefore stipulate the information flow and requirements at each network layer. For this, a communication architecture should be evaluated at how it is able to handle process modelling and service composition. These metrics can be seen in Table 3-4.

Table 3-4: Design time of a communication architecture

Criteria	Property	Unit
Design Time	Process modelling	X/3 0/3 – Excessive process modelling 1/3 – Blueprint available to guide process modelling 2/3 – Blueprint and recommendations based on input/output requirements exist 3/3 – Pre-developed plugins with existing blueprint and recommendations based on input/output requirements
	Service composition	X/3 0/3 – No service composition 1/3 – Service composition from sensors and tasks 2/3 – Service composition from sensors, systems and tasks 3/3 – Service composition selection from sensors, systems and tasks

3.4.5 Upkeep

Once a manufacturing line is completed, its true value lies within the ease of continuous operation and understanding of the operation. With the aid of data analytics, a manufacturing line can continuously produce value over its lifetime. Therefore, it is seen as a requirement for a communication architecture to specifically evaluate how it handles the life-cycle of its own upkeep.

This upkeep is often accompanied by upgrades and replacements to the physical lines. The evaluation of the communication architecture to handle reconfigurability and autonomy should be considered. The metrics for these criteria can be seen in Table 3-5.

Table 3-5: Upkeep of the communication architecture

Criteria	Property	Unit
Upkeep	Reconfigurability	X/3 0/3 – No reconfigurability 1/3 – Reconfigurability of sensors 2/3 – Reconfigurability of sensors and operations 3/3 – Total reconfigurability of operation, sensors and systems
	Autonomy	X/3 0/3 – No autonomy 1/3 – Automation of task scheduling 2/3 – Automation of production bottlenecks and task scheduling 3/3 – Autonomy of data analysis, production bottlenecks and task scheduling

3.4.6 Context Awareness

Context awareness appears within multiple sections of this study, and the inclusion of an entire characteristic of SMUs in an evaluation criterion calls for explicit outline of its implementation and assessment. Therefore, context awareness within a communication architecture needs to be assessed in terms of the information exchange between machines allowing for data reasoning and data storage, with both aspects being relatable to the requirements of communication within a I4.0 manufacturing line. The requirements for context awareness can be seen in Table 3-6.

Table 3-6: Context awareness of the communication architecture

Criteria	Property	Unit
Context Awareness	Data reasoning	X/3 0/3 – No data reasoning 1/3 – Data interpretation can happen within communication level 2/3 – Data analysis, interpretation and logic can be applied within communication level 3/3 – Data analysis, interpretation and logic can be applied within communication level and next available communication level
	Data storage	X/3 0/3 – No permanent data storage 1/3 – Required permanent data storage at communication level with no permanent data storage of other communication levels 2/3 – Required permanent data storage at communication level with permanent data storage of other communication levels 3/3 – Only relevant permanent data storage per communication level is required

3.5 Conclusion

This chapter introduced a case study for evaluation of a communication architecture in SM. From the overview of the case study, synergies and similarities were identified between the requirements of SM and the flow of communication, otherwise structured in a communication architecture. The roles and responsibilities of the production were also detailed, allowing for grouping of tasks.

The tasks and requirements from said case study led to the creation of the core components of the communication architecture. Identification of these core components reveal commonalities of operation for a SOLID-type architecture model, and the adoption of this model allowed for the relationships and execution between these core components. Further analysis of the communication architecture allowed for future plugins of different enabling attributes, allowing for longevity of the communication architecture and sensible direction for iterations under the original considerations for core components, relationships and enabling attributes.

This chapter described the role of the case study along with production scenarios to be evaluated for the effective operation of the communication architecture. Along with the communication architecture, it is important to quantify the qualitative improvement to manufacturing lines using a newly described scoring method for communication architecture in SM. Further, the manufacturing line is ultimately to be evaluated for quantitative improvements to production efficiency and bottleneck avoidance, which is once again plausible for evaluation through the production scenarios.

Chapter 4 : Results

4.1 Introduction

This chapter delves into the details and results of the operation of the SMUs under the guidance of the developed communication architecture. Whereas the aim of Chapter 3 allowed for review of important considerations, creation and understanding of the communication architecture, this chapter portrays its effectiveness once implemented. It also shows the Simulink model of the SM setup to produce customer water bottle orders, to allow for ease of comparison and outline the resulting effects.

This chapter is distributed across the various sections by:

- reviewing the Simulink model of the SM setup;
- analysing the performance of the baseline SM setup without the communication architecture to establish a baseline performance;
- showcasing the implementation of the communication architecture in the SM setup;
- reviewing the production scenarios run with the baseline and communication architecture implementation;
- outlining the quantitative metrics of the communication architecture;
- reviewing the qualitative measurements of the communication architecture;
- analysing the SM setup along with the communication architecture with the developed scoring method, and
- concluding the results of the study.

4.2 Smart Manufacturing Case Study

The SM case study, as discussed in Section 3.2, is supported by a cloud server and central server as depicted in Figure 3-1. The cloud server collects customer orders, which are then parsed to the central server for organisation and scheduling. The central server can reorganise bottle orders on command to the SMUs. The SMUs receive a daily production batch order to complete from the central server. The customer orders evaluated for this study are shown in Table 4-1.

Although several iterations of the production scenarios were run, this study also focuses on certain individual cases to highlight the intricacies of the communication architecture.

Table 4-1: Customer orders

Customer	Number of 300ml bottles ordered	Number of 500ml bottles ordered	Required delivery date
1	20	30	21 June 2025
2	70	10	21 June 2025
3	20	20	21 June 2025
4	40	30	20 June 2025
5	30	30	19 June 2025

Evaluation of data flows, data structures, production efficiency, and communication of production rates are done with the identical calculations outlined in Section 0.

4.3 Production Scenarios Performance

This section illustrates the performance of the production setup and highlights the results of numerous production metrics between the baseline implementation and communication architecture production line. Alongside these results, an explanation of how this was achieved via the flow of information and organisation of information around the communication architecture is explained. Multiple iterations of the simulated environment were run, while incorporating real-world effects and statistical shifts by allowing slight variation in production and task competition times. These effects were modelled after the collected data from the laboratory environments. These multiple iterations accounted to 100 production runs of the baseline manufacturing performance and the implementation of the communication architecture performance. While the differences in the simulated runs fell within 2% of monitored tasks and total production time, deeming it within statistical control, one of the simulated runs will be analysed to dive into the details of the performance and draw a comparison of the communication architecture performance.

4.3.1 Baseline Production Performance

The first production scenario of the baseline setup, seen in Figure 4-1, makes use of the FIFO principle of the customer order time. The orange line illustrates the total time the bottle spent in the production line and the blue line indicates the type of bottle that was fed into the system. It is important to note that the bottles fed into the SMUs were not organised.

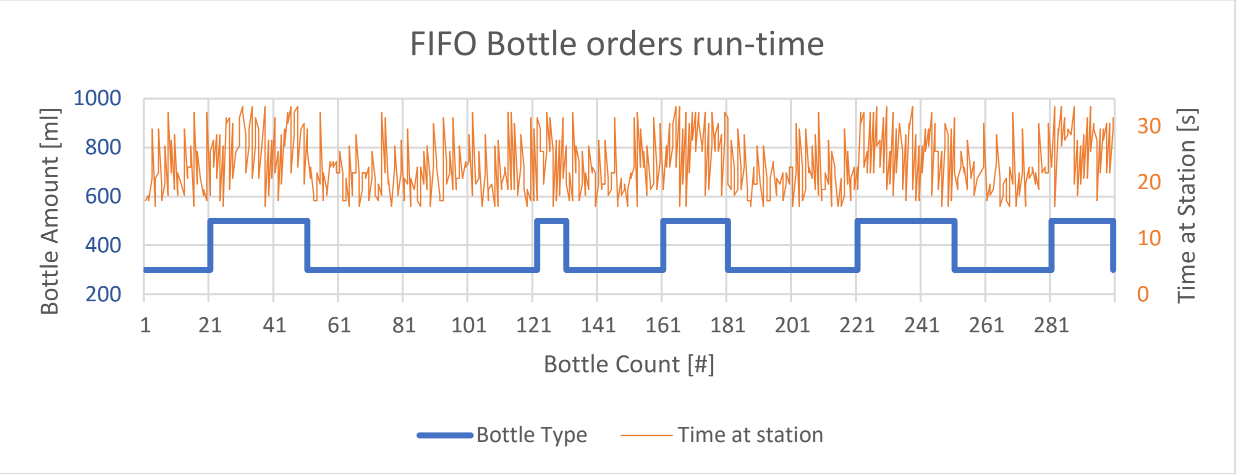


Figure 4-1: Baseline production scenario – FIFO

Figure 4-2 similarly indicates the required time to fill, cap and package a bottle for a customer order with the added complexity of customer orders being completed in sequence.

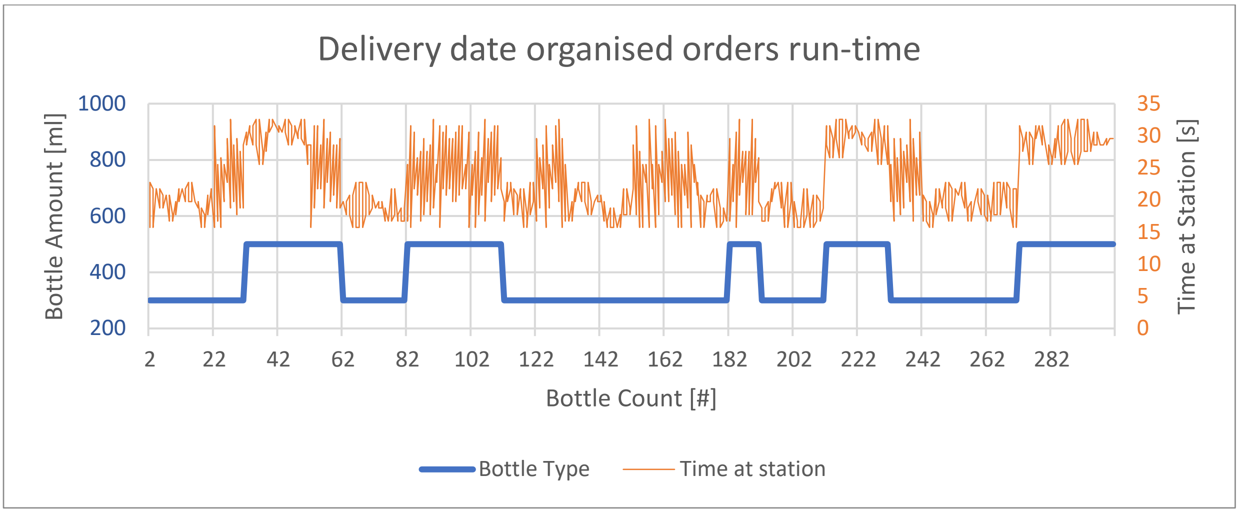


Figure 4-2: Baseline production scenario – required delivery date

Figure 4-3 depicts the baseline production performance, where all similar water bottle types are completed at once, whilst iteratively moving through the required bottle sizes.

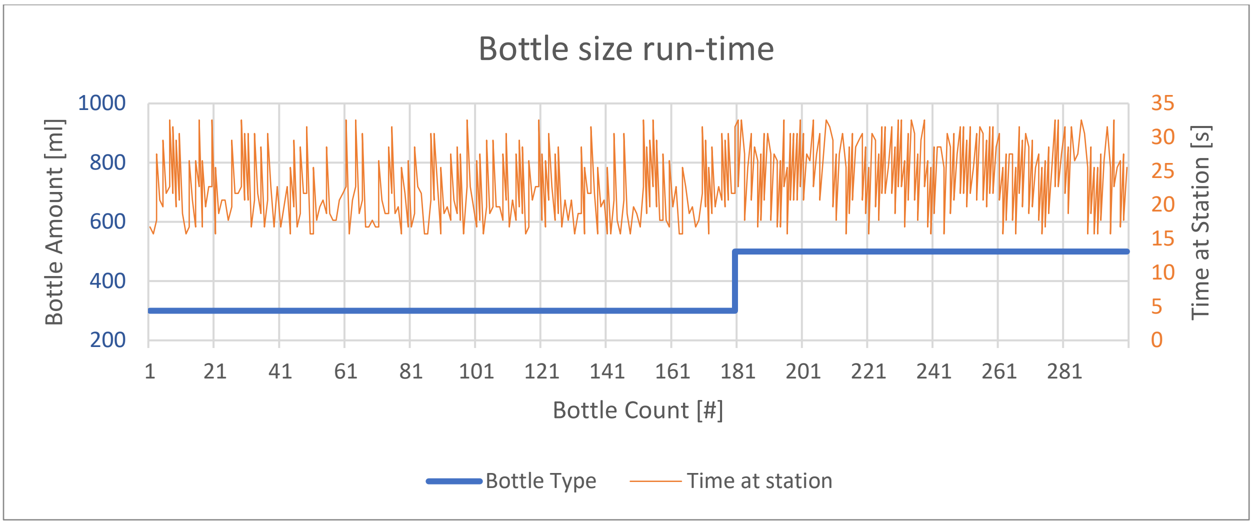


Figure 4-3: Baseline production scenario – bottle size

Figure 4-4 illustrates a random feed of bottle orders into the production line, in order to truly test the resilience of the production setup and aid in the isolation of implicating changes.

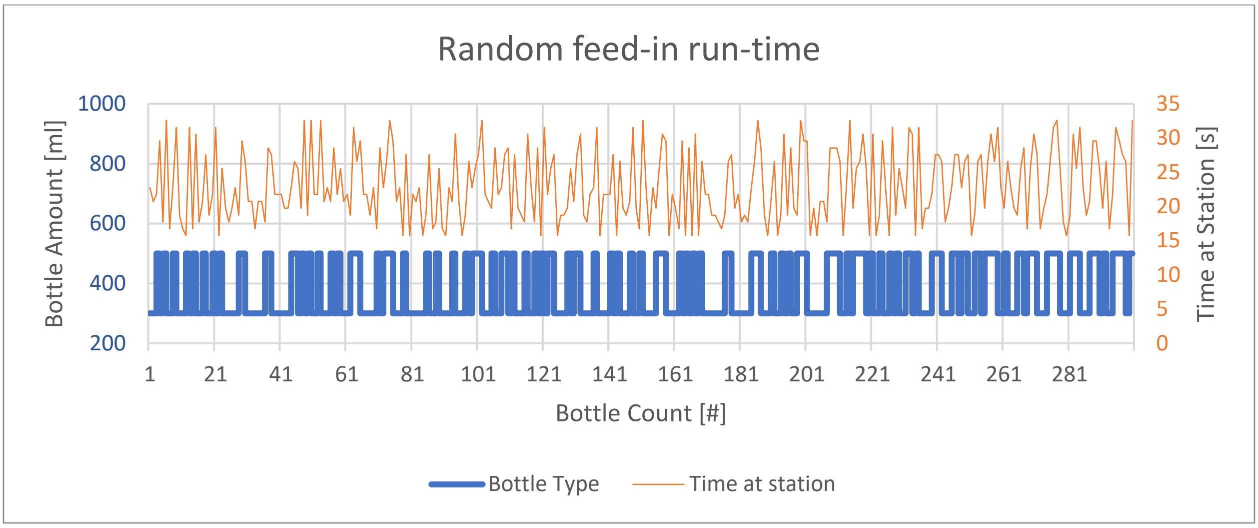


Figure 4-4: Baseline production scenario – random feed-in

The summary of all production scenarios is illustrated in Figure 4-5. An interesting aspect to point out is the apparent improvement of the optimised bottle size production scenario. This scenario seemed to have a trajectory of an improved performance in the first half of the production run; this is however quickly reversed at the knee point around 180 seconds, where the systems swops to the larger bottle-filling requirement.

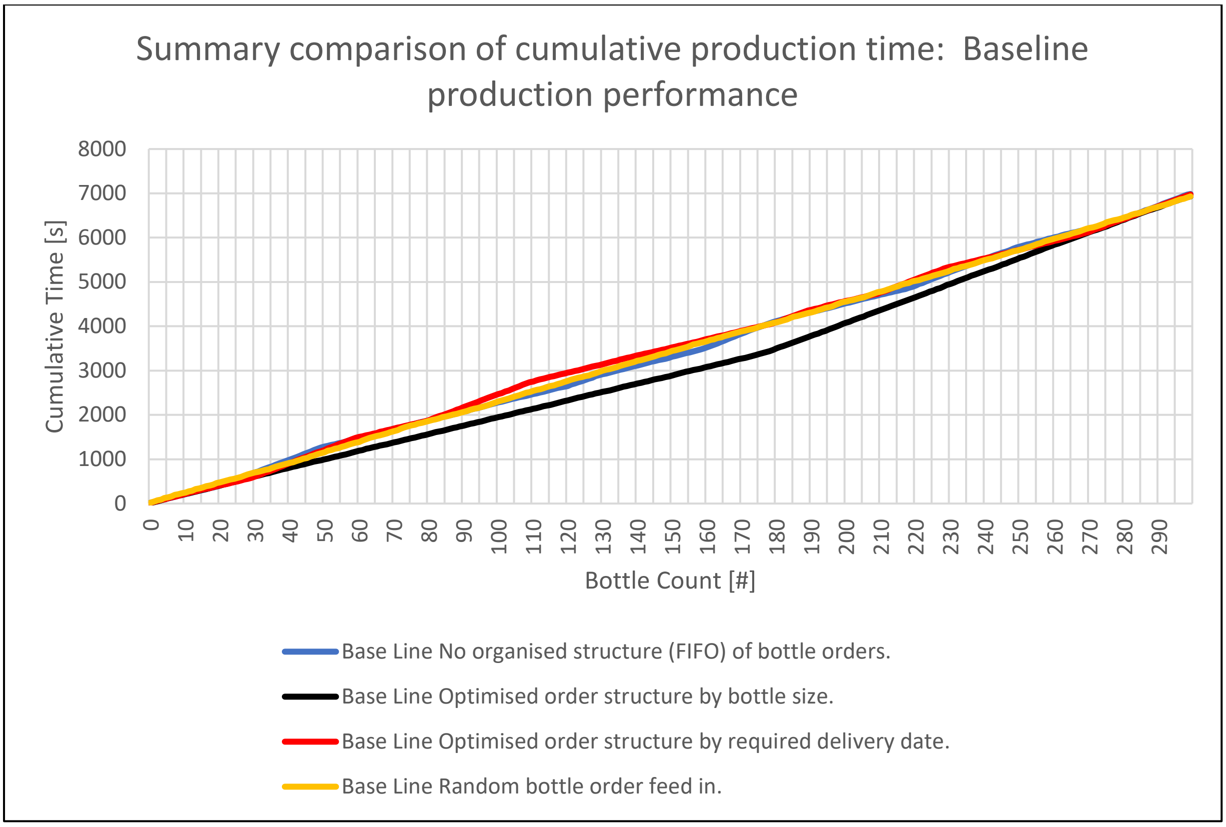


Figure 4-5: Baseline production summary

For ease, a zoomed-in performance of the last legs of the production scenarios is depicted in Figure 4-6. Within this figure it is also clearer to identify the total production time of each manufacturing scenario given the zoomed in plot. While each manufacturing scenario is closely related to each other, the total production time does deviate slightly which gives confidence to the simulation runs being within a means of statistical control.This image indicates the longest performance benchmark to be the FIFO (un-optimised) scenario. All other organisation structures seem to have negligible impact on the production run-time. This indication holds true for traditional production scenarios, as a change in order of bottle feed into the systems leads to very little optimisation. Though the change between bottle sizes (300ml to 500ml) shows very little disruption to the overall production system especially in the random scenario, this negligible improvement points to the production line at least being within process control, with environmental disruptions having insignificant effects on the production process.

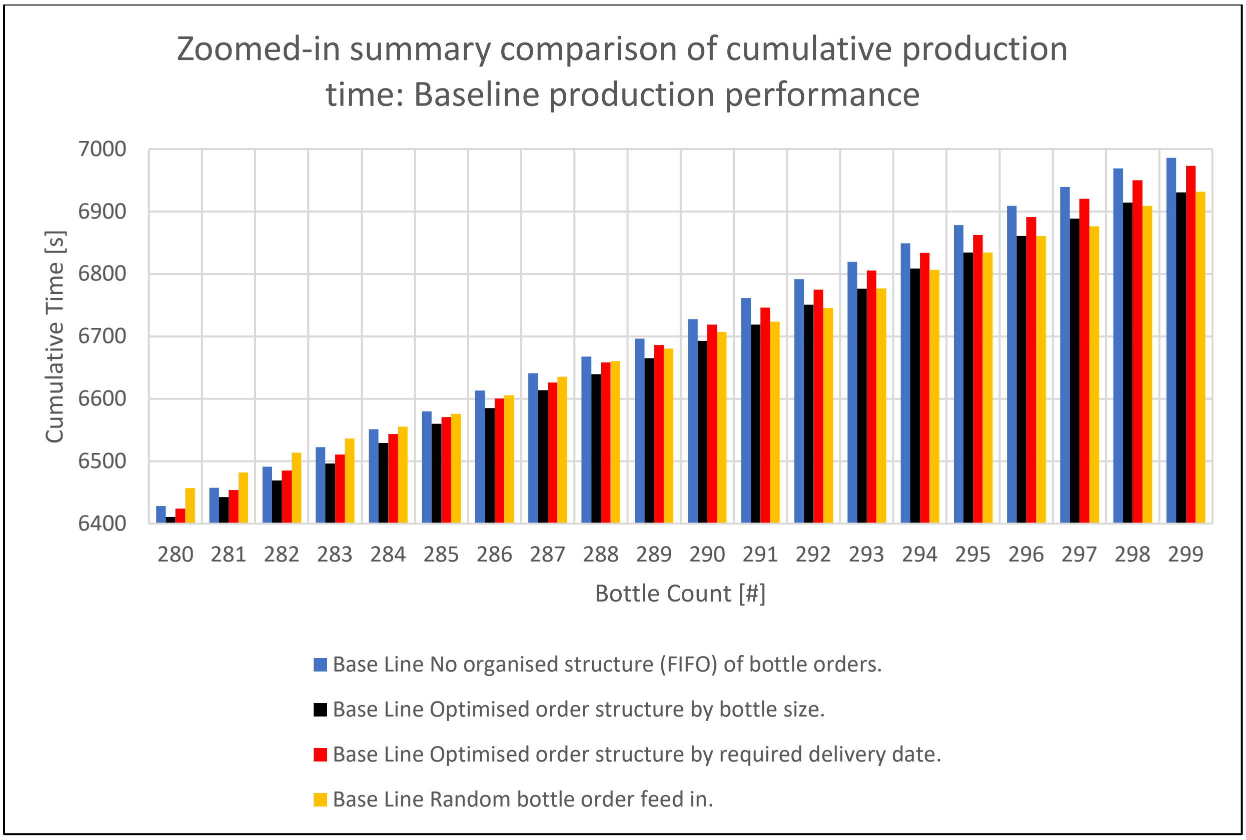


Figure 4-6: Zoomed-in baseline production summary

The setup of the production line plays a crucial role in the communication of production information and redundant monitoring of the production setup to aid in the identification of bottlenecks. Whereas the SMP isolates itself with the communication between machines, the inclusion of a communication architecture inspects all communication protocols used across all layers. This additional review allows developers to model their data for applicable use cases at responsible levels in communication layers.

An example of this is indicated in Figure 4-7 and Figure 4-8. In these figures, the central server collects production orders and optimises the order sequence from the desired production scenario. The production formation is then communicated to each SMU where the dedicated responsibilities are handled as described in Section 3.3.3.

The figures indicate the production time of the optimised bottle size approach. This case is easiest to detect production shifts and the consistency of operations from product to product. Improvements in the time to production can be seen between Customer 1 and 3 of the 300ml bottles which, at first, indicates an improved production efficiency over time. This is however unlikely and the communication logs from this production run indicate that the SMUs agreed to increase production speed when updated with available resources.

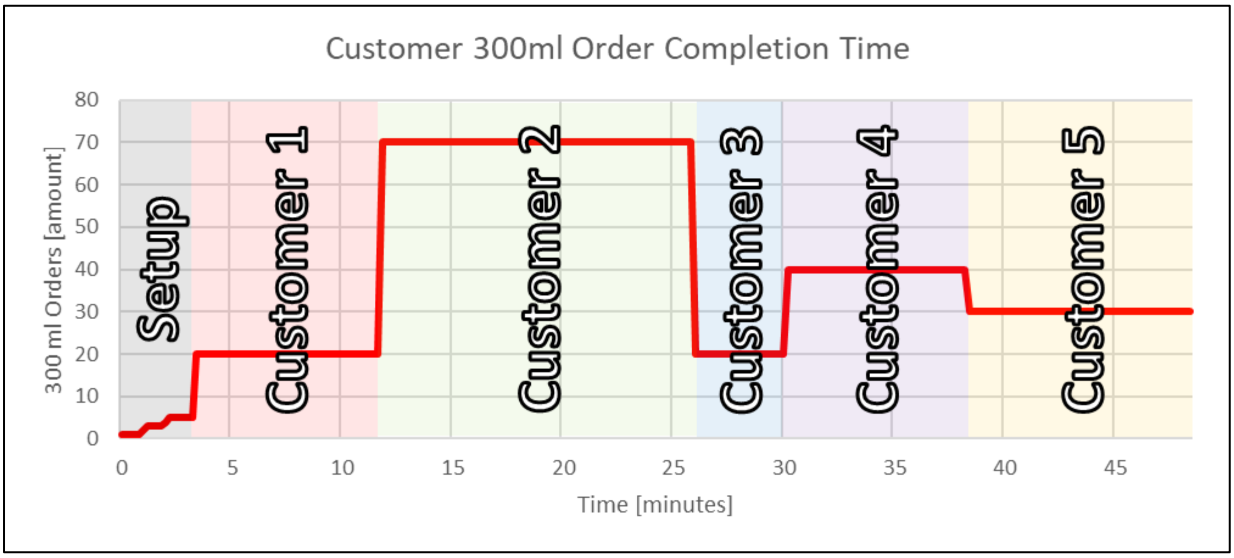


Figure 4-7: Central server customer order monitoring for 300ml

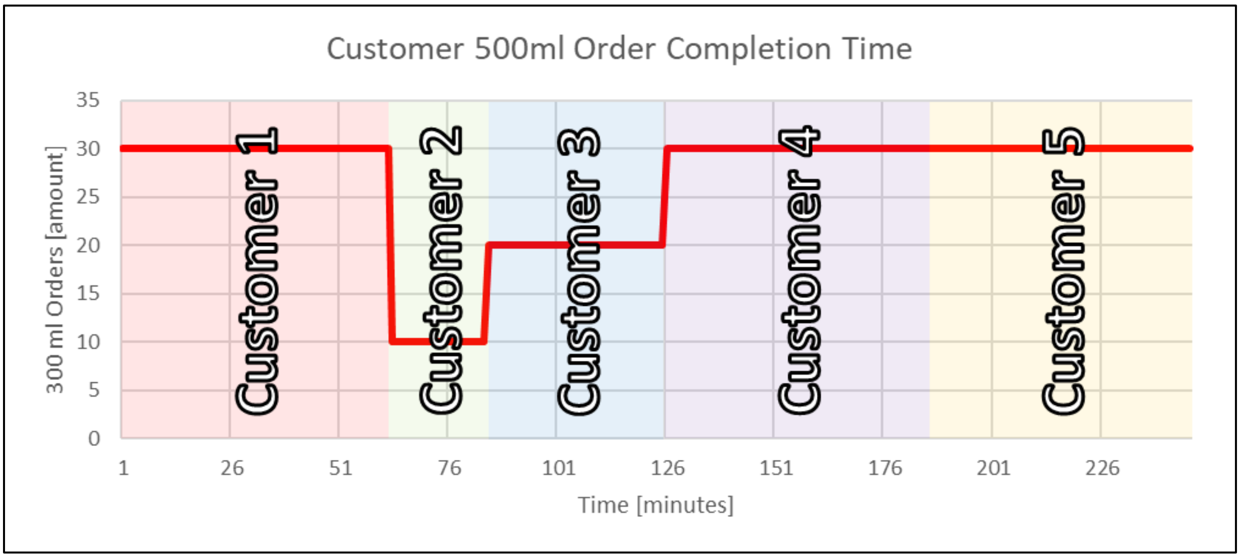


Figure 4-8: Central server customer order monitoring continued for 500ml

4.3.2 Communication Architecture Implementation

With the communication architecture allowed to communicate production specific information to nearest neighbour machines and the entire production line, the same production scenarios are set up for evaluation. Figure 4-9 illustrates the FIFO production run, with the bottle type being coloured in red, the performance of the production line to fill, cap and package one bottle in orange, and a contrasting black overlay of the baseline performance. Although the time series at first glance does indicate some noise, at a large number of cases, there are improvements in the time taken for a water bottle to travel through the production line.

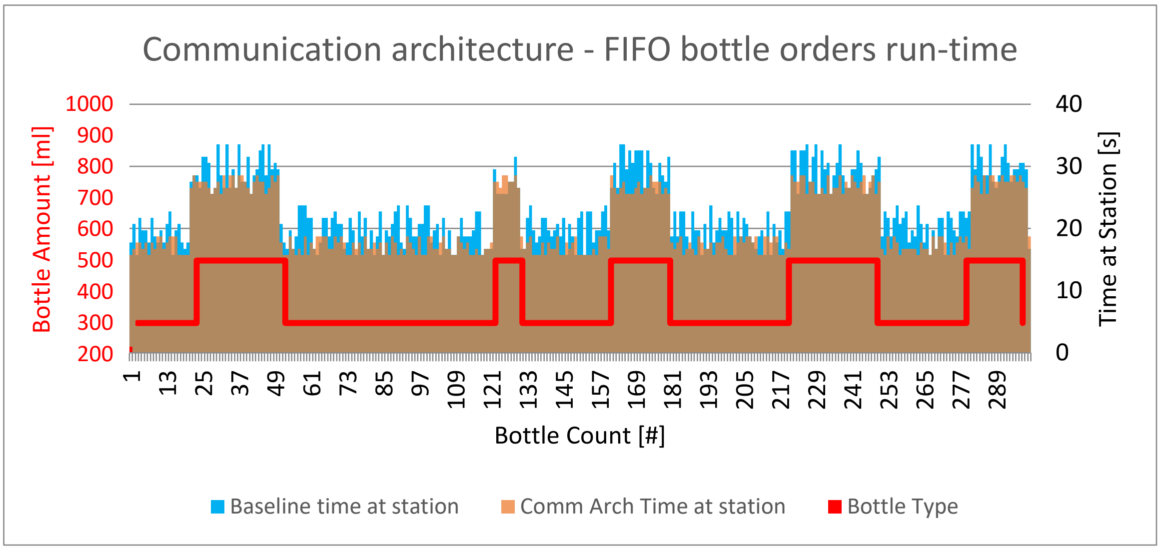


Figure 4-9: Communication architecture production scenario: FIFO

Figure 4-10 indicates the production run of the production system where the customer orders are organised by the required delivery date. With a similar dispersed clustering of the production performance, it is striking to note that the small-scale improvements below 10 seconds are enabled solely by machines having access to production information.

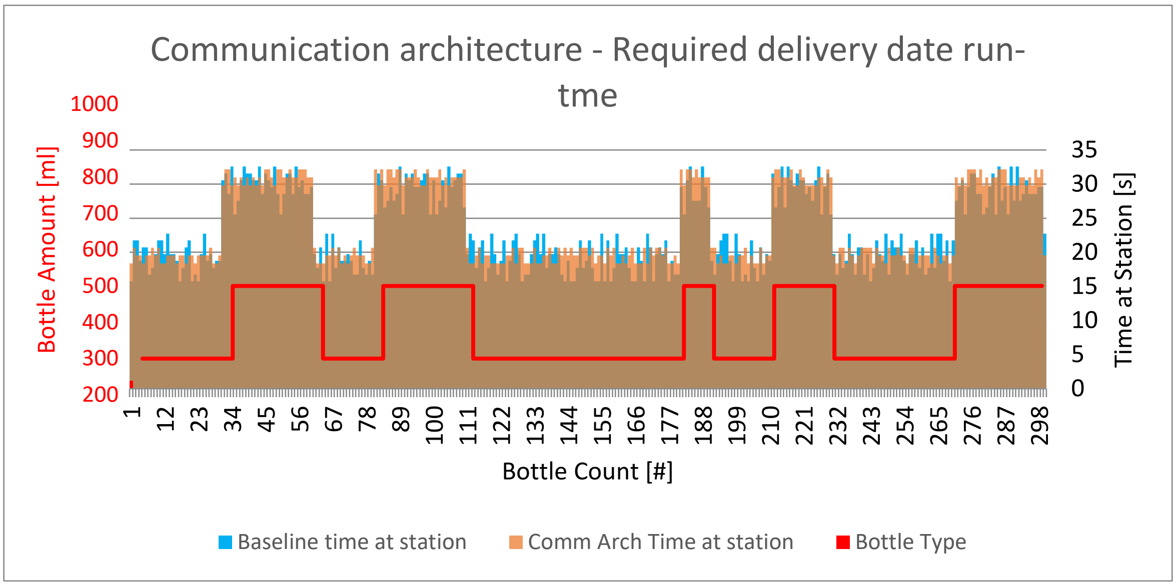


Figure 4-10: Communication architecture production scenario: Required delivery date

In reference to the SMUs' responsibilities of Section 2.2.9.1, the machines should be contextually aware of the other machines and environment around them. Whereas this can be done through attached sensors and dedicated software routines to decode message information, a communication architecture can naturally include this information. This is where the obscurity of production information can easily be transferred from machine to machine in a detached method. Figure 4-11 showcases the production scenario where identical bottles are filled to the production system in order, and here a more consistent and general improvement can be seen, where machines are commanded to return to similar positions by the use of production information.

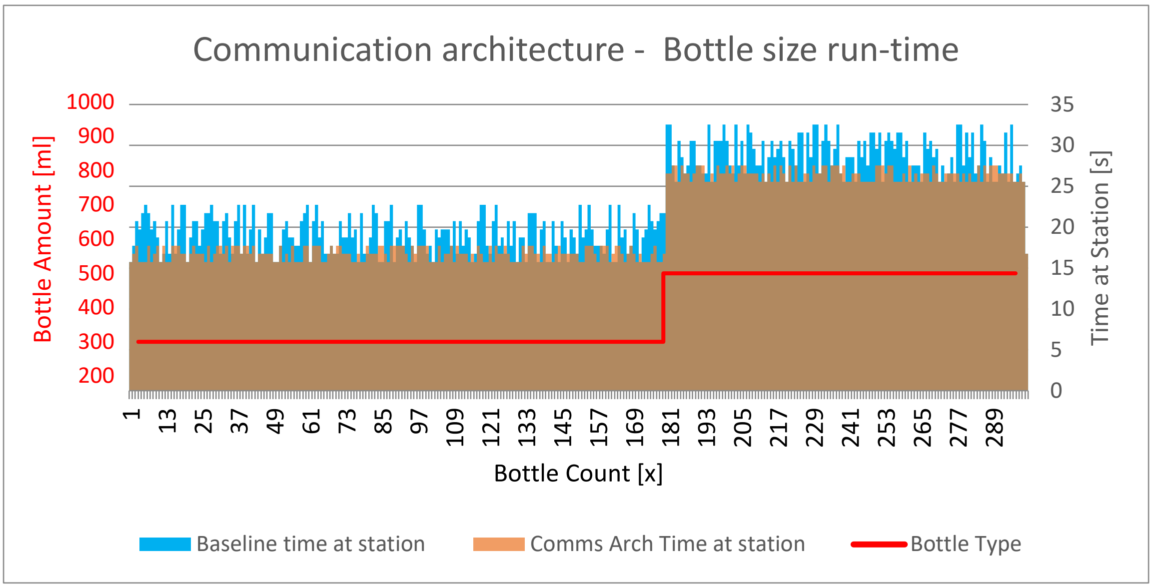


Figure 4-11: Communication architecture production scenario: Bottle size

Figure 4-12 validates the final production scenario of water bottles being fed into the production line at random. This chaotic environment still shows cases with a slight and almost general improvement to the production scenario.

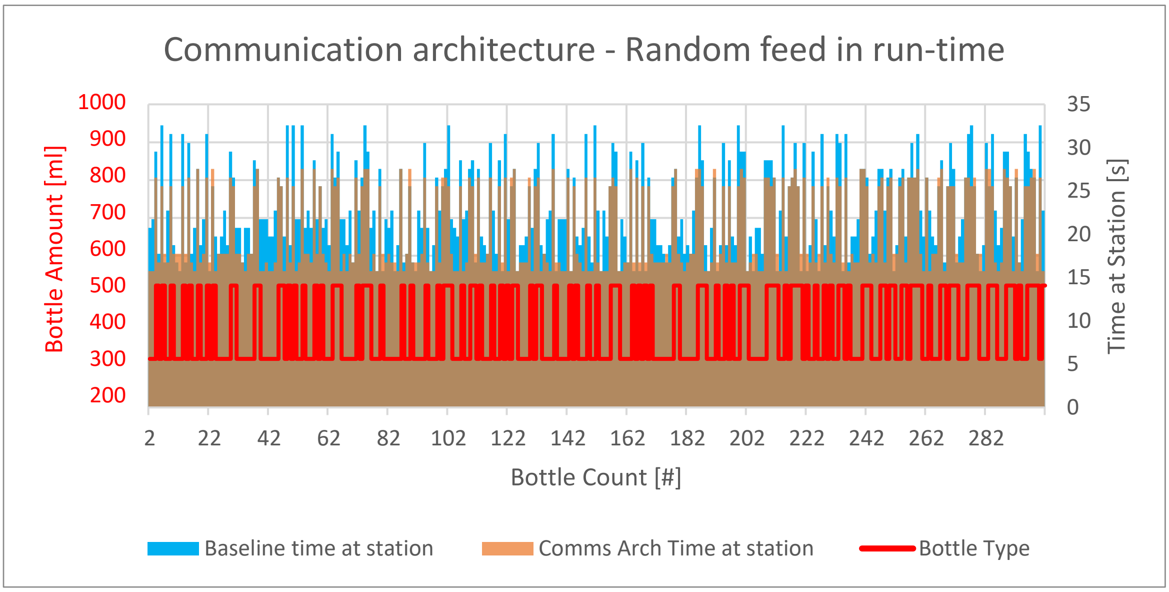


Figure 4-12: Communication architecture production scenario: Random feed-in

It is noteworthy to point out that, in all production cases, a plateau of improvement can be seen around the 180-second mark. This indicates that the machines could be optimised to a point, where the effects of communication architecture expectations allows before production limits exceed possible improvements.

Figure 4-13 summarises the production runs of the communication architecture production scenarios, where similar trends to the baseline production scenario can be seen. This offers some form of validation that production processes are consistent with moderate improvements arising from optimisation of pre-machine control and not current operation of their posed tools such as the filler pump.

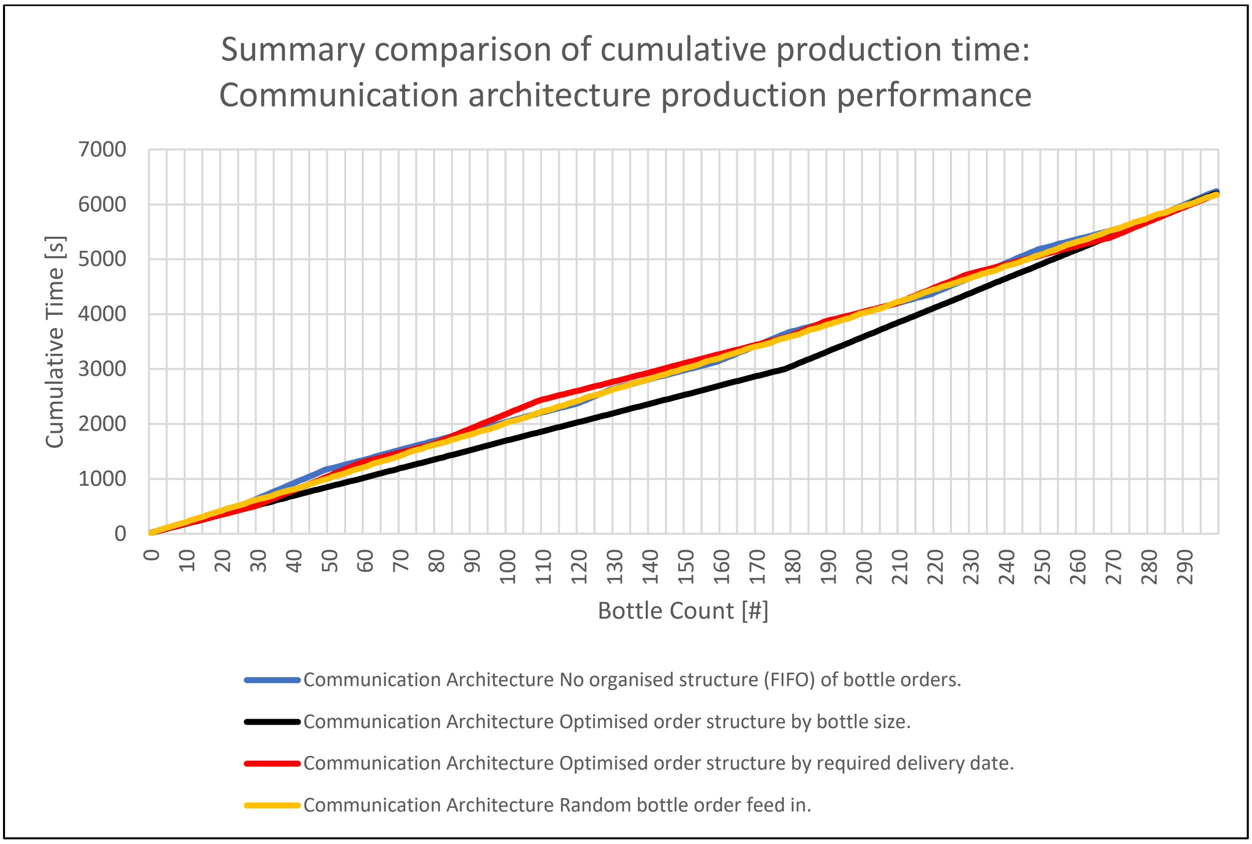


Figure 4-13: Communication architecture production summary

Figure 4-14 offers a zoomed-in summary of the latter half of the production scenarios conducted with the communication architecture. Similar patterns again seem to emerge with the FIFO structure offering the least possible optimisation, whist the plateau effect of the production scenarios for “required delivery date” and “random bottle order feed in” become more prominent.

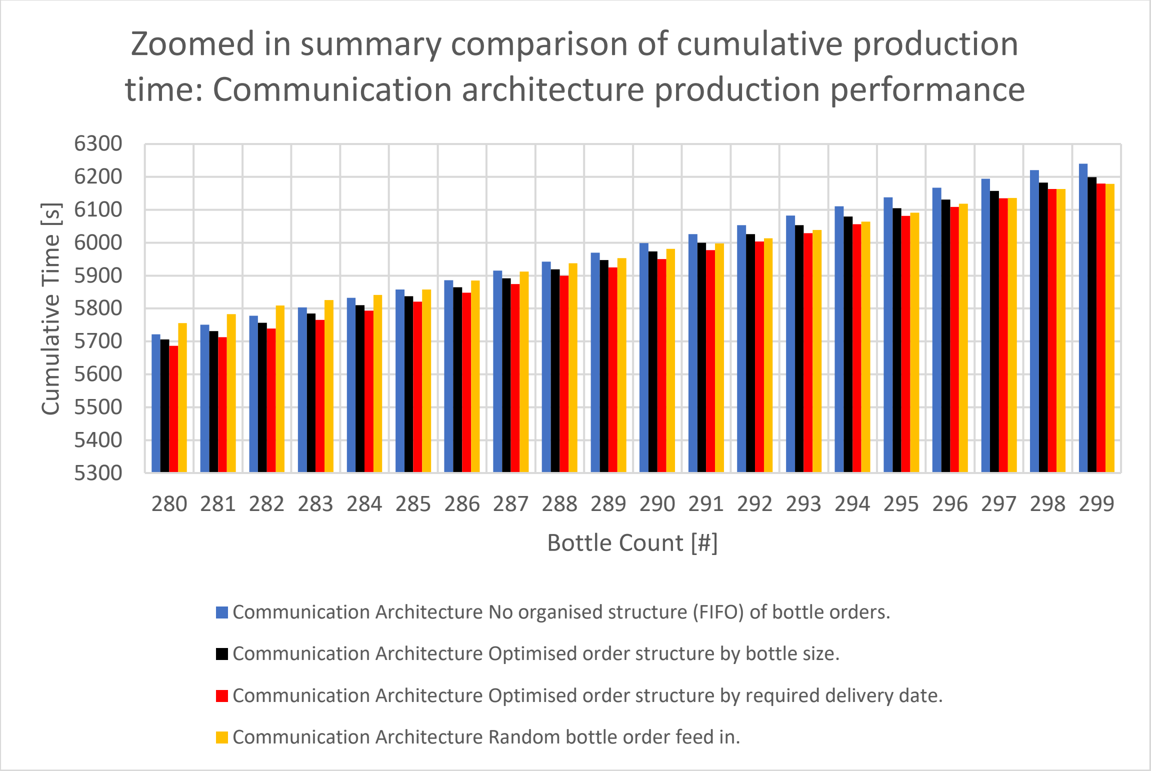


Figure 4-14: Zoomed-in communication architecture production summary

Figures Figure 4-15 andFigure 4-16 overlay both production runs across all four scenarios. Whereas there does seem to be some deconstructive interference between similar production scenarios, such as the FIFO run at 165 seconds, there are overall improvements with the communication architecture implementation.

4.3.3 Comparative Analysis

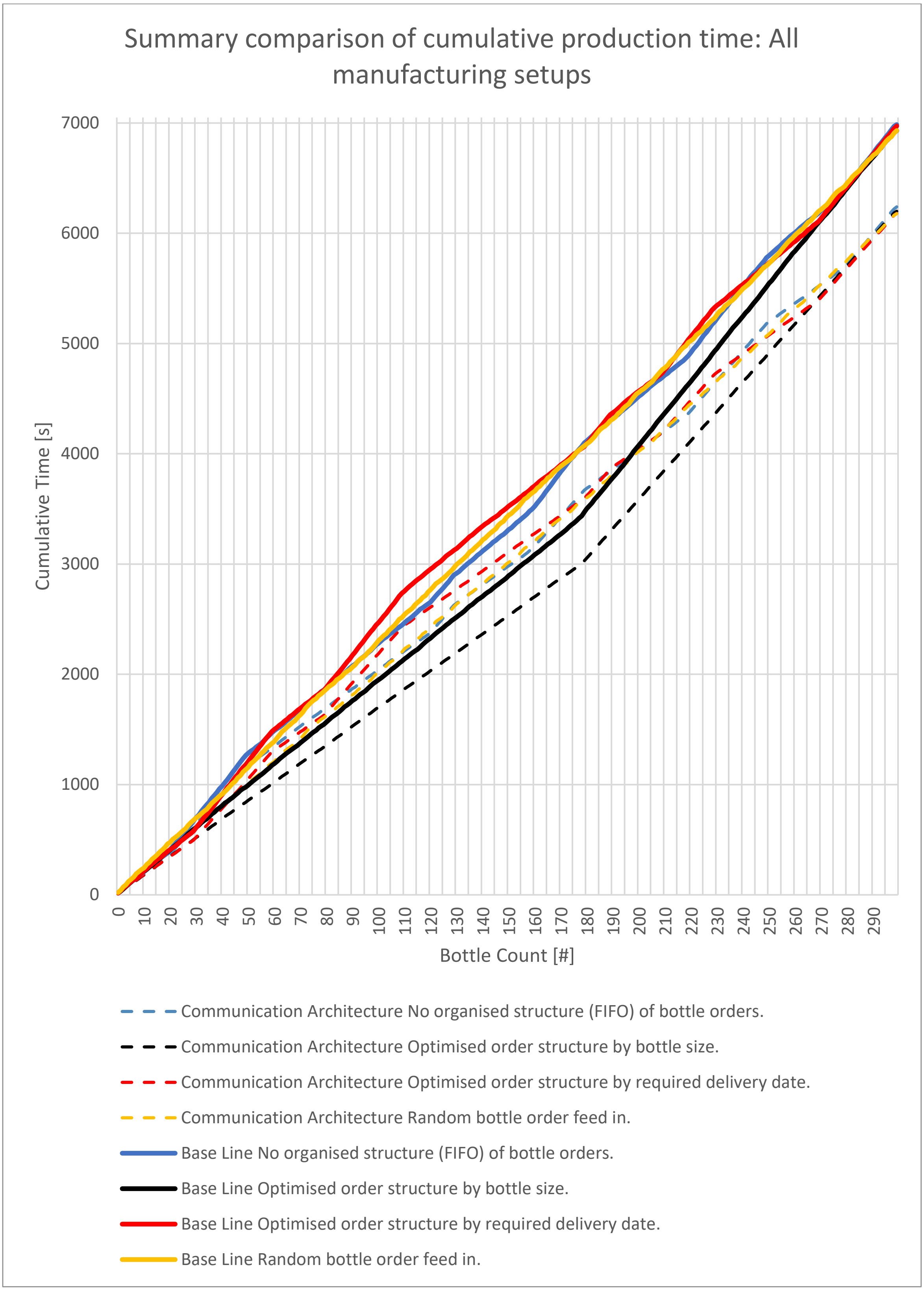
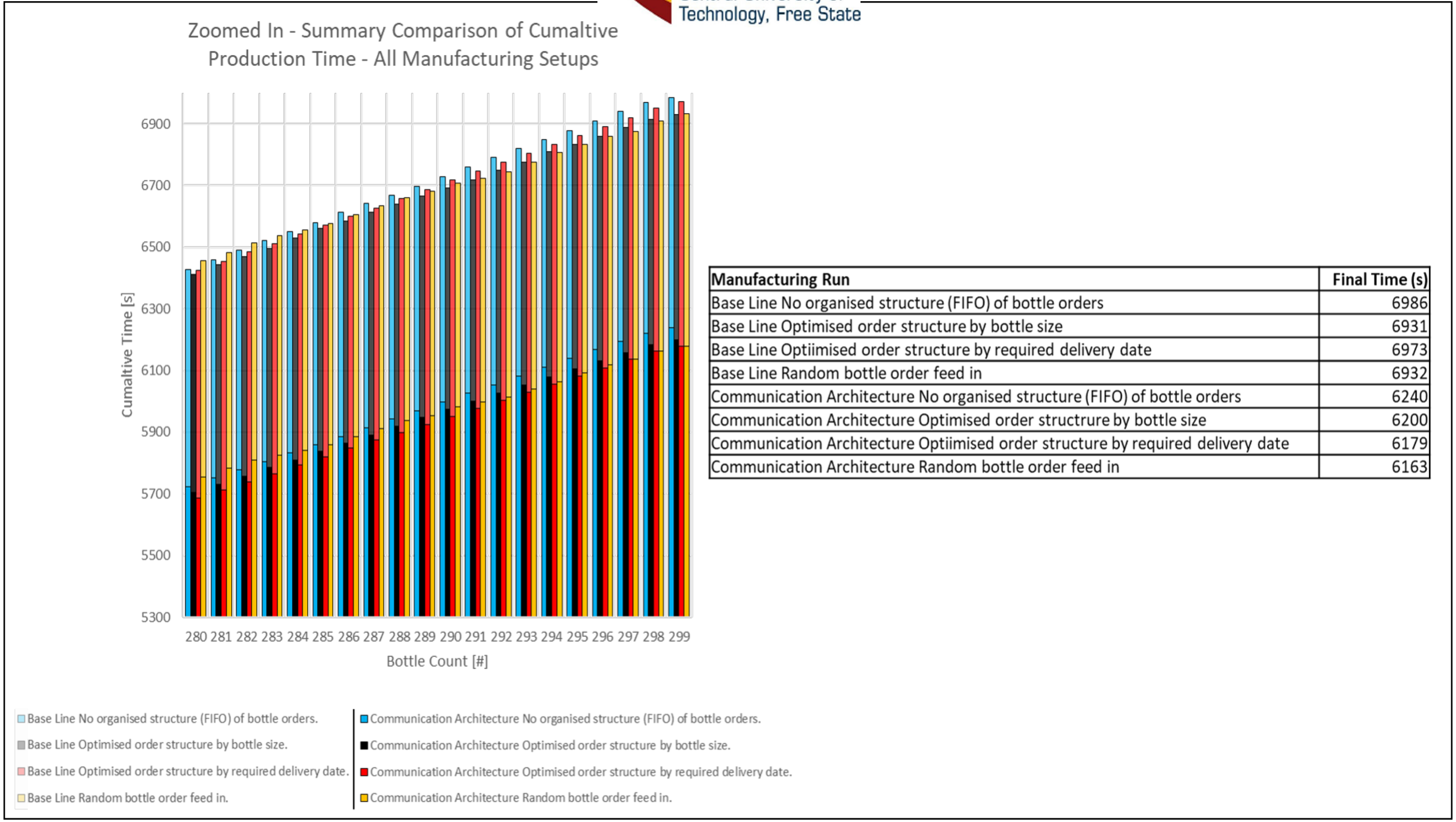


Figure 4-15: Comparative analysis of communication architecture implementation



Manufacturing Run	Final Time (s)
Base Line No organised structure (FIFO) of bottle orders	6986
Base Line Optimised order structure by bottle size	6931
Base Line Optiimised order structure by required delivery date	6973
Base Line Random bottle order feed in	6932
Communication Architecture No organised structure (FIFO) of bottle orders	6240
Communication Architecture Optimised order structrure by bottle size	6200
Communication Architecture Optiimised order structure by required delivery date	6179
Communication Architecture Random bottle order feed in	6163

Figure 4-16: Zoomed-in comparative analysis of communication architecture implementation

4.4 Quantitative Assessment of Developed Communication Architecture in Smart Manufacturing

The production scenarios outlined on the baseline and communication architecture implementations offer the best viewpoint to evaluate the improved production efficiency. Table 4-2 summarises the improved performance with the implementation of the communication architecture on the production scenarios. Whereas the improvement is not drastically significant, the ability to improve production rates with the inclusion of information access and organisation is astonishing. This correlation implies that intelligent machines having the ability to run automatously requires greater care in the structure and access to data in a manufacturing network.

While there is an apparent convergence of the performance in both the baseline manufacturing and communication architecture implementation scenarios, this seems to be explained by the unchanged production process manipulation where instead the only the apparent product order is altered. While this alteration does aid in the improvement of total production time, possibly by saving time of SMU 2 to reposition its actuators for bottling as an example, it does not aid in specific production metrics when the scope of the experiment is changed. For example, an optimised by delivery approach may reduce a percentage of late orders when incorporating logistic concerns, but this is outside the scope of the study.

The performance improvements of the baseline manufacturing scenarios are further accentuated by a further seven hundred seconds with the inclusion of the communication architecture. This is understood to reduce the reactive time of machines preparing next steps of the manufacturing process, before the product arrives at the SMU station and reducing the reactive workload by removing redundant sensing and calculations done by other SMUs in the production line. The communication architecture rather then passes collected information on about production statuses and products that SMUs further down the line may trust and use to prepare next actions.

With the total percentage improvements of Table 4-2 and zoomed in total production time of Figure 4-16, it is ostensible of a convergence of metrics.

Table 4-2: Communication architecture percentage improvement summary

Production run	Communication architecture percentage improvement
No organised structure (FIFO) of bottle orders	10.68%
Optimised order structure by required delivery date	11.37%
Optimised order structure by bottle size	10.55%
Random bottle order feed-in	10.86%

4.5 Qualitative Assessment of developed Communication Architecture in Smart Manufacturing

The SMP offers the best lens through which to inspect informed performances of the communication architecture with respect to machine intelligence. This intelligence is measured against the machine's contextual awareness and its ability to automatously resolve bottlenecks.

With the inclusion of the SMP and allowing machines to act upon the information received at a protocol level, the SMUs have immediate access to request and gather all structured parts of other SMUs' data. These parts include environmental sensors for either parts of the machine or the products currently being attended to. This information can be uniquely indexed and communicated to each SMU in the data field of the SMP, allowing for machines to gather precise information for any kind of calculation.

The SMP collection of sensor data also allows for any attached machine on the network to identify, calculate and predict production rates that would satisfy each SMU's minimum speed operation condition. This could also be complimented with logistical information for refuelling of resources in an entire I4.0 implementation. Ultimately, this culminates in SMUs being able to independently adjudicate production speeds across the network, where network hierarchies can hold specific responsibilities for these calculations.

The communication of information of neighbouring machines, also allows for SMUs to prepare for incoming products, reducing any repetitive tasks that can accumulate over large production runs. This information, coupled with the correlation of production speeds across the manufacturing line, makes up the largest contribution for avoiding and reducing bottlenecks in the manufacturing line.

The auto-identification of attached machines through the SMP "machine identifier" field further allows SMUs to specifically identify attached machines on the manufacturing line. Machines being enabled by the SMP fields allow them to calculate current production capabilities and correlate these to production requirements. This already allows for "Go" and "No-Go" production start indicators and allows for coupling to reconfigurable systems to always satisfy "Go" operations in the highly customisable factories of I4.0.

This access to the information of each machine also allows it to be scheduled and organised throughout different layers of the network where, for instance, a central server may choose to keep a collection of production information but not sensor information. The collection of this data allows for central servers to directly feed in production data, allowing for redundant monitoring through DTs.

4.6 Architecture Scoring Assessment

The communication architecture was also evaluated against the scoring method outlined in Section 3.4. This evaluation model was compared to other I4.0 architectures [75, 76]. Although these architectures were not always communication architectures, they were evaluated to be the closest comparison to I4.0 implementations.

The results of the evaluation must be taken with a bias towards communication architectures that more closely align to communication architecture scoring methods [77]. However, with the positive correlation of the production efficiency garnered in Section 0, this is accepted and should rather indicate a need for further refinement in independent scoring mechanisms for communication architectures.

Table 4-3 indicates the results of the architecture scoring with the communication architecture (CA) showing a compliance with the responsibilities and abilities required from communication architectures in general.

Table 4-3: I4.0 architecture scoring

		Architecture						
Characteristic	Property	Base[1] [78]	DMP[2] [79]	MAPE-K[3] [80]	WS02[4] [81]	IoT-O[5] [82]	HTN[6] [83]	CA[7]
Responsiveness	Time to access data	500 ms	2000 ms	500 ms	100 ms	200 ms	200 ms	60 ms
	Database bandwidth	10 Mb/s	10 Mb/s	10 Mb/s	10 Mb/s	10 Mb/s	10 Mb/s	10 Mb/s
Accuracy	Data degradation	0 %/min	0 %/min	2 %/min	0 %/min	2 %/min	0 %/min	0 %/min
	Compression loss	5%	10%	10%	5%	10%	5%	0%
	Data recency	3/3	1/3	2/3	2/3	3/3	2/3	3/3
Transmission	Data size limit	16 Mb	16 Mb	16 Mb	16 Mb	16 Mb	16 Mb	16 Mb
	Availability	99%	75%	85%	95%	75%	85%	90%
	Traceability	3/3	1/3	2/3	3/3	2/3	3/3	3/3
	Networkability	2/3	1/3	2/3	2/3	1/3	2/3	3/3
Design Time	Process modelling	1/3	1/3	2/3	2/3	1/3	2/3	2/3
	Service composition	2/3	1/3	2/3	3/3	1/3	2/3	2/3
Upkeep	Reconfigurability	3/3	1/3	3/3	2/3	2/3	3/3	2/3
	Autonomy	3/3	1/3	2/3	3/3	2/3	3/3	2/3
Contextual Awareness	Data reasoning	1/3	1/3	1/3	3/3	2/3	3/3	3/3
	Data storage	3/3	1/3	3/3	3/3	2/3	3/3	2/3
Percentage normalisation		77%	51%	68%	86%	62%	84%	91%

[1] Biography, Attributes, Schedule, and Execution (BASE)
[2] Digital message platforms (DMPs)
[3] Monitor Analyse Plan Execute and Knowledge (MAPE-K)
[4] Web Services oxygenated version 2 (WS02)
[5] Internet of Things ontology (IoT-O)
[6] Hierarchical task network (HTN) planner
[7] Smart manufacturing communication architecture (CA)

Chapter 5 : Discussion

5.1 Communication Architecture Elements and Relations

The communication architecture outlined within this study draws inspiration from its core components for the natural symbiotic relationship it can share with the SM traits. Without the need for redundancy around the enabling technologies and core components in its implementation, this study views the core component selection as a necessity, grouping for control of the flow of communication at a communication protocol and extension into architecture domain.

The relationships between all five core components allow for greater control and outline of the requirements of each core component to be met, whilst – similar to the communication network of the SMUs – this adds an extra layer of complexity. Fortunately, this complexity is elegantly handled with the use of hierarchies and responsibilities split among the relationships, like that of the network hierarchy and level responsibilities.

To describe the interactions of the communication architecture, consider the example of an SMU being powered up for the first time within the manufacturing scenario. Within the third layer of the network, an SMU is decentrally connected to each other SMU and the central server. An attempt to establish network identity is kick-off, where an IP address is requested and assigned. A query then goes out to the network to establish which machines are connected onto the network. Each machine then transmits is relevant network information (IP address), machine information (i.e. machine type collected from the SMP), attached sensor information (about the products it measures) and its physical space description (if possible). This ensure machines have a copy of the manufacturing line capabilities. Each machine distributes its information at regular and required intervals, ensuring information is adequately updated based on scenario requirements. The SMUs equipped with the relevant information, should then allocate monitoring and measuring tasks amongst the remaining SMUs and acknowledge each role. This scenario works elegantly in a counter-clockwise motion of the communication architecture, starting at the decentralized core component and moving around to each other core component as in Figure 2-9. Other scenarios, such as the updating of production competition and next products in production, can transmit information from the Information Appropriate stage to the Network Hierarchy stage, circumventing or including the Distributed core component as needed.

5.2 Impact of Architecture Traits

The use of the communication architecture also brings about new paradigm traits for information flow in the network it is implemented in. These traits are discussed below.

5.2.1 Congruent Information

The use of the communication architecture in the outlined case study is largely possible when the SMUs on the network have access to congruent information. This is solved at a machine level with the use of the SMP, where information can be actively pushed and gathered using the PUSH and GET methods.

This, in contrast with perpetual updating of information from a central location, aids in having more consistent production information flow on the network in smaller portions. Although the network strain might be more consistently elevated, the network spikes would be drastically reduced in periodically fetching large updates from all machines. However, this also flows agreeably to satisfy the requirement of real-time communication.

Through the SMP handling information congruency naturally at a machine level, it can be at developer's discretion how to have congruency flow upwards for use by the central server, in the case of redundancy or DT implementation, and to the cloud server for data analytics. The outlined layer responsibilities of the communication architecture aid in outlining how and when to implement this.

These aspects both align in allowing SMUs to create real-time decisions based on appropriate information. This allows SMUs to ultimately create faster and more precise predictions both at a machine and manufacturing line (system) level.

5.2.2 Contextual Awareness

The ability of SMUs to be contextually aware with regard to aspects of networks and environments does imply extra requirements to fully realise this ability. Whereas SMUs have access to machine-specific information using the machine identifier tag of the SMP, SMUs might also need this standardisation of possible technologies at sensor levels. This will allow SMUs not to share calculated values of its environment but instead share raw sensor information to other SMUs and allow them to discern environmental conditions. This not only allows SMUs to validate these calculations of other SMUs but can similarly be used to offload nuanced information for processing of singular SMUs on the network. Doing so might bring about a different sense of edge computing, or rather, edge communication.

5.3 Considerations for Core Component Selection

The specific selection of the core components of the communication architecture was done to align with the requirements of I4.0 and SM by identifying possible groups in processes and a theory of computational location. For instance, instead of information being collected and processed within defined production software loops, this information can be accessed directly from machines, allowing for protocol implementations to handle production information. This allows for software application loops to be built on top of the protocol stack and still gain access to the information in the traditional manner.

These core components of the communication architecture are also evaluated through the architect ring scoring method created, where improvements and balances can be made to individual core components, creating optimisation to required targets. For future creation of architectures, similar processes of requirement identification, process grouping, development, evaluation and iterations can be followed.

5.4 Communication Architecture Implementation

The use of the developed scoring method dually allows for a quantifiable approach to measure the improvement of the communication architecture, and aids in identifying areas of future improvement through iterations of the communication architecture. Though the scoring method attributes the organisation structure of the data flow in the production scenario, it also relates the responsibilities of the architecture into meaningful enabling targets for implementation.

Although the close match of the scoring method to the proposed architecture type and the closed environment of development that couples the scoring method to the communication architecture should be considered, the impact on identifications for improvement is only possible with this first iteration.

The implementation of the communication architecture, along with the evaluation of the scoring method, indicates direct improvements in the production scenario. Due to the ability of the SMUs to prepare resources, strategize production rates and offer first insights into production prediction, this is all realised using structured data. The communication architecture further shows how this data structure can be enacted upon in a different manner from other hardware and software architectures.

The quantitative benefits of data transmission and organisation, along with the qualitative improvements of prediction and intelligence embedded into SMUs are therefore seen as a synergistic improvement from the communication architecture.

5.5 Communication Architecture Contribution

Existing research outlined in Section 2.7 revealed gaps in the implementation and control around data flows at various levels of a SM implementation. This study's original contribution to identify complementary aspects between SM requirements and data structures culminated in the use of the communication architecture.

Further, this vision of a communication architecture was then implemented into a baseline case study for specific analysis around the quantitative improvements to the production efficiency and the qualitative improvements for intelligence and reactionary speed from the SMUs. Finally, the evaluation metrics of the scoring method for critiquing and guiding iterations of the communication architecture allowed for an indirect roadmap for iterations of the communication architecture. These aspects are categorised as unique contributions from this study.

5.6 Inclusions for Future Improvements

There have been proven effects of the use of the communication architecture in this case study, however, the architecture itself would be greatly improved with evaluation and iteration on multiple other case studies. The scoring method could follow a similar path for refinement, whilst a greater expansion of the production scenarios would also feed into the improvement of the communication architecture.

The communication architecture would also benefit from a circular inclusion and improvement of the ***CustomerRequiredDate*** formula to allow greater task scheduling and optimisation.

The communication architecture also makes provision for the inclusion of other technology implementations with I4.0 – namely, IoT, DT, product communication and collaborative decision making – however, additional work is needed to fully realise these aspects into the communication architecture and group their effects and requirements.

A world envisioned by this study of the communication architecture and collaborative decision making to allow users on a manufacturing line to pinpoint inefficiencies and feed into predictive maintenance would be a great stride for man and machine working side by side.

Future studies should be evaluated with the focus of back propagating the simulated implementation on the laboratory setup, in order to validate real world effects and hone the implications of the communication architecture.

Branching out into other fields of research, such as artificial intelligence, a great deal of work could be poured into self-automation of programme loops by the collection of information of the communication architecture. SMUs on the production line could evaluate trends of manufacturing scenarios and create short-term highly optimised loops to be executed until deviations are dedicated. For example, the manufacturing scenario of the bottle type scenario, could see programming instructions be implanted to machines to always prepare 300ml bottles until instructed otherwise. This could allow SMUs to further optimise production timings where the reconfiguration of preparation time is outweighed by the consistency time saving for each product that passes of a given amount.

Finally, the network aspect improvements of the implementation of a communication architecture will need to be characterised. Though there are additional qualitative improvements with the use of the communication architecture in SMUs, there is undoubtedly additional network overhead added with every communicated packet. Whereas this overhead still satisfies the real-time communication bounds of I4.0, additional scale will require optimisation and alternative strategies to be developed.

Chapter 6 : Conclusion

6.1 Introduction

This chapter concludes this study by highlighting the unique contributions within the field of I4.0 and communication architectures. A high-level overview of the research goals is followed by summarising the results and unique contributions of this study to highlight their importance and significance. Finally, the chapter concludes by discussing future work that can be explored in the research fields of this study.

6.2 Summary of Study

The opening chapter of this study introduced the field of research and the research project. This study also cast a glance on the research gaps currently prevalent in the fields of I4.0 and communication architecture. Following onto this, an overview of the research hypothesis, aims and objectives rounded off the first chapter.

The second chapter of this study dove into the connections of I4.0 and communication architectures. Although no real form of communication architectures existed before this study, a review of similar architecture types aided in revealing methodologies that can be followed and requirements to suit the communication needs of I4.0.

The third chapter focused on evaluating the requirements of the communication architecture by incorporating a SM case study and its requirements. A communication architecture was then created at a higher level, with its core components and relationships identified. The communication architecture was also expanded on for co-operation, for the inclusion of other enabling technologies within I4.0. These aspects not only expanded the explanation of the communication architecture, but aided in allowing for a drop in implementations with existing manufacturing setups. The communication architecture was then developed into the manufacturing case study, with communication flows being monitored for evaluation of production improved performance. Finally, the chapter proceeded to a suggested communication architecture scoring method, to further iterate, quantify and improve communication architectures in their own life-cycle.

The fourth chapter summarised the results of the communication architecture implementation and compared it to the baseline performance of the case study in this section for ease of readability. This chapter took a closer look at the improved production performance in terms of production efficiency and bottleneck avoidance.

The fifth chapter analysed the results of chapter four to discern applicable causes for the production performance improvement. This chapter also followed on with a discussion of the holistic performance of the communication architecture, to provide guidance on future implementation and architectures in a similar manner.

6.3 Research Objectives

The main research objective of this study was to develop a communication architecture for improvement in production efficiencies and bottleneck avoidance in SM setups. The staging point for the development of this communication architecture is sprawled with preluding gaps in research due to the rapid expansion of I4.0. This study therefore set out to lay a path for the creation of the communication architecture in a few stages.

The study evaluated similar architecture types, in the form of hardware and software architectures, to identify similarities, input considerations, outputs, responsibilities and appropriate methods for architectures and their development. From this, similarities naturally arose with extensive research conducted into SM. From this initial research, comprehensive production scenarios to evaluate the performance of the communication architecture were established.

Next, this study set out to benchmark a current manufacturing case study, to evaluate its performance against these production scenarios. An evaluation for possible improvements to production efficiencies and bottleneck avoidance was then envisioned by reviewing these production results.

Furthermore, the study then set about creating a communication architecture to allow for communication control for SMUs to allow them to collect and act upon production data and fulfil the requirements of SM. The use of this communication architecture therefore also improves the ability of the SMUs and sets a paradigm for how and where production data should flow in a manufacturing line.

The study then went about extensively testing the communication architecture against the production scenarios and analysing the data to draw impactful conclusions from the results garnered.

Then, the study discussed the performance and data gathered from the results, in order to gauge effective use of the communication architecture and allow for recommendations and improvements to be suggested.

6.4 Unique Contribution of the Study

This study developed a communication architecture that was implemented in a Simulink smart manufacturing case study, which was tasked to improve production efficiencies and avoid bottlenecks. Further to this, the following contributions of this study are identified to be novel.

6.4.1 Incorporating Intelligence into Smart Manufacturing Units

The use of a communication architecture for SMUs works alongside the requirements set out for SM. This allows I3.0 machines to be upgraded to I4.0 machines by adhering to the communication standards of the architecture and its associated protocol. For machines to incorporate the ability of contextual awareness not only within the network, but the physical environment itself, aligns machines to the requirements of SM and enables them to be classified as SMUs.

This communication architecture also allows for SMUs to communicate with interoperability at the core of their methods, allowing for decentralised communication alongside appropriate information being shared. Other aspects, such as data analytics and agility through scalability also become integratable to the SMUs.

6.4.2 Improved Production Efficiency

The use of the communication architecture allows SMUs to communicate current production needs to machines on the network in real-time. This allows for strategies and information to be shared to all other machines, naturally leveraging the ability of machines to prepare or learn and adapt for incoming production means. These abilities allow for the SMUs to prepare for production and gain access to a wider suite of information. This all culminates in SMUs leveraging shared information alongside their operational coding to improve production efficiency by cutting out idle time.

6.4.3 Bottleneck Avoidance

The communication of current productions statuses, such as availability of resources, allows SMUs to set consistent production speeds across the manufacturing line. This consistency that is agreed upon across the network and manufacturing line, ensures no SMU is left under greater strain and can reduce the backlog build-up of products at singular sources as this backlog and bottleneck can eventually slow down upstream SMUs, creating unintended chaos in the manufacturing line. However, even this chaos can be communicated with the communication architecture, allowing not only for mitigation in terms of bottlenecks, but also resolution, leading to a fulfilled meaning of bottleneck avoidance.

6.4.4 Future Architecture Creation and Evaluation

This study discussed the aspects it considers important for a communication architecture in an I4.0 environment. This study also identified the significance not only of the improved performance on a production scenario, but the qualitative improvement through which architectures can aid in organisation and structure. The study therefore self-evaluates the communication architecture due to a lack of applicable measurable criteria fit for communication architectures. This methodology and evaluation could be taken for guidance in future architecture creation and development.

6.5 Scientific Outcomes

Gericke, G.A., Kuriakose, R.B. and Vermaak, H.J. 2026. Contextual Awareness in a Communication Architecture for Smart Manufacturing. Information and Communication Technology for Intelligent Systems. Bangkok, Thailand. 9-11 April. DOI: In publication

Gericke, G.A., Kuriakose, R.B. and Vermaak, H.J., 2025, August. Quantitative Assessment of a Communication Architecture for Smart Manufacturing. In *Intelligent Systems Conference* (pp. 60-74). Cham: Springer Nature Switzerland. DOI: 10.1007/978-3-031-99965-9_4

Gericke, G.A., Kuriakose, R.B. and Vermaak, H.J. 2025. Key Considerations for Appropriate Information in A Communication Architecture For Smart Manufacturing. Information and Communication Technology for Intelligent Systems. Lusaka, Zambia. 28-29 August. DOI: 10.1007/978-981-96-6929-5_9

Gericke, G.A., Kuriakose, R.B. and Vermaak, H.J., 2023, September. A Communication Architecture Approach for Mitigating Complexities in Smart Manufacturing Units by Means of Information Appropriateness. In *World Conference on Information Systems for Business Management* (pp. 91-98). Singapore: Springer Nature Singapore. DOI: 10.1007/978-981-99-8346-9_8

Gericke, G.A., Kuriakose, R.B. and Vermaak, H.J., 2022, July. Developing an improved software architecture framework for smart manufacturing. In *Congress on Intelligent Systems: Proceedings of CIS 2021, Volume 1* (pp. 87-100). Singapore: Springer Nature Singapore. DOI: 10.1007/978-981-16-9416-5_7

6.6 The Future Use of Communication Architectures

There are improvements from SMUs on production efficiency that can be corroborated to the communication of the next immediate production, but the communication architecture can also be expanded for communication of the organised structure of production orders. This communication will allow for a hybrid optimisation, where SMUs are able to optimise the next immediate operations of the products, whilst the central server provides longer term optimisation of production organisation. This long-term and short-term corroboration would lead to a hyper optimised system, all facilitated by the communication architecture.

Additional study to analyse specific effects of the communication architecture improvement to be pinpointed and to identify noise signals in improvements, i.e., why some improvements are not consistent.

References

1. Grefen, P.W., 2025. Business information system architecture. G. DBA.
2. Amrani, A.Z., Ortega, I.U. and Vallespir, B., 2024. Industry 4.0 technologies and lean production combination: A strategic methodology based on links quantification. *International Journal of Technology Innovation and Management (IJTIM)*, 4(2), pp. 1–18.
3. Braunisch, N., Gneuß, T., Schmidt, U., Ristin, M., Van De Venn, H.W. and Wollschlaeger, M., 2024, November. Service-Oriented Architecture for I4. 0 Digital Twins. In *IECON 2024-50th Annual Conference of the IEEE Industrial Electronics Society* (pp. 1–6). IEEE.
4. Gharibvand, V., Kolamroudi, M.K., Zeeshan, Q., Çınar, Z.M., Sahmani, S., Asmael, M. and Safaei, B., 2024. Cloud based manufacturing: A review of recent developments in architectures, technologies, infrastructures, platforms and associated challenges. *The International Journal of Advanced Manufacturing Technology*, 131(1), pp. 93–123.
5. Gericke, G.A., 2021. Creating Decentralised SMART Manufacturing Units with a Newly Implemented Standard Communication Protocol.
6. Yu, Z., Khan, S.A.R. and Umar, M., 2022. Circular economy practices and industry 4.0 technologies: A strategic move of automobile industry. *Business Strategy and the Environment*, 31(3), pp. 796–809.
7. Ryalat, M., ElMoaqet, H. and AlFaouri, M., 2023. Design of a smart factory based on cyber-physical systems and Internet of Things towards Industry 4.0. *Applied Sciences*, 13(4), p. 2156.
8. Kopetz, H. and Steiner, W., 2022. Internet of things. In *Real-time systems: design principles for distributed embedded applications* (pp. 325–341). Cham: Springer International Publishing.
9. Hu, Y., Jia, Q., Yao, Y., Lee, Y., Lee, M., Wang, C., Zhou, X., Xie, R. and Yu, F.R., 2024. Industrial internet of things intelligence empowering smart manufacturing: A literature review. *IEEE Internet of Things Journal*, 11(11), pp. 19143–19167.
10. Briscoe, B., Schepper, K.D., Bagnulo, M. and White, G., 2023. Low latency, low loss, scalable throughput (L4S) internet service: Architecture. RFC 9330.
11. Zhang, K., Shi, Y., Karnouskos, S., Sauter, T., Fang, H. and Colombo, A.W., 2022. Advancements in industrial cyber-physical systems: An overview and perspectives. *IEEE Transactions on Industrial Informatics*, 19(1), pp. 716–729.

12. Singh, H. and AlMangour, B. eds., 2023. *Handbook of smart manufacturing: forecasting the future of industry 4.0*. CRC Press.
13. Khang, A. and Akhai, S., 2024. Green intelligent and sustainable manufacturing: key advancements, benefits, challenges, and applications for transforming industry. *Machine Vision and Industrial Robotics in Manufacturing*, pp. 405–417.
14. Chen, H., Jeremiah, S.R., Lee, C. and Park, J.H., 2023. A digital twin-based heuristic multi-cooperation scheduling framework for smart manufacturing in IIoT environment. *Applied Sciences*, 13(3), p. 1440.
15. Ejaz, M.R., 2023. Implementation of industry 4.0 enabling technologies from smart manufacturing perspective. *Journal of Industrial Integration and Management*, 8(02), pp. 149–173.
16. Carrera-Rivera, A., Larrinaga, F. and Lasa, G., 2022. Context-awareness for the design of Smart-product service systems: Literature review. *Computers in Industry*, 142, p. 103730.
17. Martínez-Gutiérrez, A., Díez-González, J., Verde, P., Ferrero-Guillen, R. and Perez, H., 2023. Hyperconnectivity proposal for smart manufacturing. *IEEE Access*, 11, pp. 70947–70959.
18. Halenar, I., Halenarova, L. and Tanuska, P., 2023. Communication safety of cybernetic systems in a smart factory environment. *Machines*, 11(3), p. 379.
19. Tanaya, P.I., 2024. The State of The Knowledge of Cloud Computing Technology for Manufacturing Automation Systems. *Jurnal Rekayasa Sistem Industri*, 13(1), pp. 79–90.
20. Khang, A. ed., 2024. AI-Oriented Competency Framework for Talent Management in the Digital Economy: Models, Technologies, Applications, and Implementation. CRC Press.
21. Ren, L., Dong, J., Zhang, L., Laili, Y., Wang, X., Qi, Y., Li, B.H., Wang, L., Yang, L.T. and Deen, M.J., 2024. Industrial metaverse for smart manufacturing: Model, architecture, and applications. *IEEE Transactions on Cybernetics*, 54(5), pp. 2683–2695.
22. Noor-A-Rahim, M., Firyaguna, F., John, J., Khyam, M.O., Pesch, D., Armstrong, E., Claussen, H. and Poor, H.V., 2022. Toward industry 5.0: Intelligent reflecting surface in smart manufacturing. *IEEE Communications Magazine*, 60(10), pp. 72–78.
23. Khan, S.I., Kaur, C., Al Ansari, M.S., Muda, I., Borda, R.F.C. and Bala, B.K., 2025. Implementation of cloud based IoT technology in manufacturing industry for smart control of manufacturing process. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 19(2), pp. 773–785.

24. Castro, C.M.T., Sharma, A., Kumar, D.S., Abidi, K. and Kim, N., 2022, December. The implementation of thread network for a smart factory. In *2022 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 253–260). IEEE.
25. Vasilev, P., Boneva, A., Ivanova, V. and Tsvetanov, T., 2024, October. Smart Manufacturing Integration of Heterogeneous Information Systems via OPC UA. A Case Study. In *2024 International Conference Automatics and Informatics (ICAI)* (pp. 466–471). IEEE.
26. G.A. Gericke, H. Vermaak and R.B. Kuriakose, "Communication Protocol Review for SMART Manufacturing Units within a Cloud Manufacturing Environment," *2019 International Conference on Fourth Industrial Revolution (ICFIR)*, Manama, Bahrain, 2019, pp. 1–6, doi: 10.1109/ICFIR.2019.8894788.
27. Gericke, G.A., Kuriakose, R.B., Vermaak, H.J. and Madsen, O., 2020, July. Machine to machine communication protocol for SMART manufacturing units. In *Journal of Physics: Conference Series* (Vol. 1577, No. 1, p. 012047). IOP Publishing.
28. Mejía-Moncayo, C., Kenné, J.P. and Hof, L.A., 2023. On the development of a smart architecture for a sustainable manufacturing-remanufacturing system: A literature review approach. *Computers & Industrial Engineering*, 180, p. 109282.
29. Bujari, A., Calvio, A., Garbugli, A. and Bellavista, P., 2023, June. A layered architecture enabling metaverse applications in smart manufacturing environments. In *2023 IEEE International Conference on Metaverse Computing, Networking and Applications* (MetaCom) (pp. 585–592). IEEE.
30. Mensah, F., 2024. Zero trust architecture: A comprehensive review of principles, implementation strategies, and future directions in enterprise cybersecurity. *International Journal of Academic and Industrial Research Innovations (IJAIRI)*, 10, pp. 339–346.
31. Ching, F.D., 2023. *Architecture: Form, space, and order*. John Wiley & Sons.
32. Zhang, Y., Udrea, F. and Wang, H., 2022. Multidimensional device architectures for efficient power electronics. *Nature Electronics*, 5(11), pp. 723–734.
33. Forlingieri, M., Lamm, J.G., Roth, S. and Walker, M., 2022. *Model-based system architecture*. John Wiley & Sons.
34. Fries, J., Freund, M. and Harth, A., 2023. A solid architecture for machine data exchange with access control. *Proceedings of the 1st Semantic Web on Constrained Things, Hersonissos, Greece*, 3412, pp. 74–81.
35. Garrido, M., 2022. A survey on pipelined FFT hardware architectures. *Journal of Signal Processing Systems*, 94(11), pp. 1345–1364.

36. Choudhry, M.D., Sivaraj, J., Munusamy, S., Muthusamy, P.D. and Saravanan, V., 2024. Industry 4.0 in manufacturing, communication, transportation, and health care. *Topics in Artificial Intelligence Applied to Industry 4.0*, pp. 149–165.
37. Laharia, S. and Laharia, A., Core Elements of Machine Type Communication in 6G. JOURNAL OF TECHNICAL EDUCATION, p.290.
38. Kılıç, R. and Erkayman, B., 2023. Multi-criteria analysis through determining production technology based on critical features of smart manufacturing systems. *Soft Computing – A Fusion of Foundations, Methodologies & Applications*, 27(11).
39. Hu, Y., Jia, Q., Yao, Y., Lee, Y., Lee, M., Wang, C., Zhou, X., Xie, R. and Yu, F.R., 2024. Industrial internet of things intelligence empowering smart manufacturing: A literature review. *IEEE Internet of Things Journal*, 11(11), pp. 19143–19167.
40. Walter, S. and Mikkola, M., 2023. Advancing Networked Production Through Decentralised Technical Intelligence. In *Artificial Intelligence in Manufacturing: Enabling Intelligent, Flexible and Cost-Effective Production Through AI* (pp. 281–300). Cham: Springer Nature Switzerland.
41. Antons, O. and Arlinghaus, J.C., 2024. Designing distributed decision-making authorities for smart factories–understanding the role of manufacturing network architecture. *International Journal of Production Research*, 62(1–2), pp. 204–222.
42. Terziyan, V. and Vitko, O., 2025. Context-Aware Machine Learning for Smart Manufacturing. *Procedia Computer Science*, 253, pp. 25–36.
43. Santos, A., Lima, C., Reis, A., Pinto, T., Nogueira, P. and Barroso, J., 2023, July. Design of Context-Aware Information Systems in Manufacturing Industries: Overview and Challenges. In *International Symposium on Distributed Computing and Artificial Intelligence* (pp. 322–331). Cham: Springer Nature Switzerland.
44. Estrada-Jimenez, L.A., Pulikottil, T., Nikghadam-Hojjati, S. and Barata, J., 2023. Self-organization in smart manufacturing—background, systematic review, challenges and outlook. *IEEE Access*, 11, pp. 10107–10136.
45. Cordeiro, R.F., Reis, L.P. and Fernandes, J.M., 2023. A hierarchical model for industry 4.0 concepts. RAM. *Revista de Administração Mackenzie*, 24, p.eRAMR230061.
46. Liu, X., Liu, K. and Zhang, Z.H., 2025. Complex network topology design on technology collaboration: small-world characteristics for intelligent manufacturing. *International Journal of Production Research*, pp. 1–22.

47. Banerjee, S., Jesubalan, N.G., Kulkarni, A., Agarwal, A. and Rathore, A.S., 2024. Developing cyber-physical system and digital twin for smart manufacturing: Methodology and case study of continuous clarification. *Journal of Industrial Information Integration*, 38, p. 100577.
48. Cong, J., Chen, C.H., Meng, X., Xiang, Z. and Dong, L., 2023. Conceptual design of a user-centric smart product-service system using self-organizing map. *Advanced Engineering Informatics*, 55, p. 101857.
49. Visconti, P., Rausa, G., Del-Valle-Soto, C., Velázquez, R., Cafagna, D. and De Fazio, R., 2024. Machine learning and IoT-based solutions in industrial applications for Smart Manufacturing: a critical review. *Future Internet*, 16(11), p. 394.
50. Coetzer, J., 2023. "Developing a protocol for collaborative decision-making in a smart manufacturing environment" (Doctoral dissertation, Central University of Technology).
51. Modrak, V. and Soltysova, Z., 2023. Influence of manufacturing process modularity on lead time performances and complexity. *Applied Sciences*, 13(12), p. 7196.
52. Othman, U. and Yang, E., 2023. Human–robot collaborations in smart manufacturing environments: review and outlook. *Sensors*, 23(12), p. 5663.
53. O'Connell, E., O'Brien, W., Bhattacharya, M., Moore, D. and Penica, M., 2023, May. Digital twins: Enabling interoperability in smart manufacturing networks. In *Telecom* (Vol. 4, No. 2, pp. 265–278). MDPI.
54. Caiazzo, B., Murino, T., Petrillo, A., Piccirillo, G. and Santini, S., 2023. An IoT-based and cloud-assisted AI-driven monitoring platform for smart manufacturing: design architecture and experimental validation. *Journal of Manufacturing Technology Management*, 34(4), pp. 507–534.
55. John, J., Noor-A-Rahim, M., Vijayan, A., Poor, H.V. and Pesch, D., 2024. Industry 4.0 and beyond: The role of 5G, WiFi 7, and time-sensitive networking (TSN) in enabling smart manufacturing. *Future Internet*, 16(9), p. 345.
56. Mohr, F., 2024. "Smart Data Analytics for Manufacturing Processes" (Doctoral dissertation, Massachusetts Institute of Technology).
57. Guo, J., Cheng, Y., Wang, D., Tao, F. and Pickl, S., 2023. Industrial Dataspace for smart manufacturing: connotation, key technologies, and framework. *International Journal of Production Research*, 61(12), pp. 3868–3883.
58. Fowler, D.S., Epiphaniou, G., Higgins, M.D. and Maple, C., 2023. Aspects of resilience for smart manufacturing systems. *Strategic Change*, 32(6), pp. 183–193.

59. Mahmoodi, E., Fathi, M., Tavana, M., Ghobakhloo, M. and Ng, A.H., 2024. Data-driven simulation-based decision support system for resource allocation in industry 4.0 and smart manufacturing. *Journal of Manufacturing Systems*, 72, pp. 287–307.
60. Ochoa, W., Larrinaga, F. and Pérez, A., 2023. Context-aware workflow management for smart manufacturing: A literature review of semantic web-based approaches. *Future Generation Computer Systems*, 145, pp. 38–55.
61. Gunasekaran, K., Kumar, V.V., Kaladevi, A.C., Mahesh, T.R., Bhat, C.R. and Venkatesan, K., 2023. Smart decision-making and communication strategy in industrial Internet of Things. *IEEE Access*, 11, pp. 28222–28235.
62. Cervantes, H. and Kazman, R., 2024. *Designing software architectures: A practical approach*. Addison-Wesley Professional.
63. Mejía-Moncayo, C., Kenné, J.P. and Hof, L.A., 2023. On the development of a smart architecture for a sustainable manufacturing-remanufacturing system: A literature review approach. *Computers & Industrial Engineering*, 180, p. 109282.
64. Hu, Y., Jia, Q., Yao, Y., Lee, Y., Lee, M., Wang, C., Zhou, X., Xie, R. and Yu, F.R., 2024. Industrial internet of things intelligence empowering smart manufacturing: A literature review. *IEEE Internet of Things Journal*, 11(11), pp. 19143–19167.
65. Schlemitz, A. and Mezhuyev, V., 2024. Approaches for data collection and process standardization in smart manufacturing: Systematic literature review. *Journal of Industrial Information Integration*, 38, p. 100578.
66. Wu, Y., Dai, H.N., Wang, H., Xiong, Z. and Guo, S., 2022. A survey of intelligent network slicing management for industrial IoT: Integrated approaches for smart transportation, smart energy, and smart factory. *IEEE Communications Surveys & Tutorials*, 24(2), pp. 1175–1211.
67. Saavedra Sueldo, C., Perez Colo, I., De Paula, M., Villar, S.A. and Acosta, G.G., 2023. ROS-based architecture for fast digital twin development of smart manufacturing robotized systems. *Annals of Operations Research*, 322(1), pp. 75–99.
68. Haricha, K., Khiat, A., Issaoui, Y., Bahnasse, A. and Ouajji, H., 2023. Recent technological progress to empower smart manufacturing: Review and potential guidelines. *IEEE Access*, 11, pp. 77929–77951.
69. Peixoto, T., Oliveira, B., Oliveira, Ó. and Ribeiro, F., 2025. Data Quality Assessment in Smart Manufacturing: A Review. *Systems*, 13(4), p. 243.

70. Rajković, P., Aleksić, D., Janković, D., Milenković, A. and Đorđević, A., 2025. Hybrid deployment strategy for software updates to the manufacturing execution system layer. *Computer Science and Information Systems*, pp. 27–27.
71. Roy, S., Ahmed, I., Saldanha, J., Medini, K. and Wuest, T., 2024. Bottleneck Management through Strategic Sequencing in Smart Manufacturing Systems. *Smart and Sustainable Manufacturing Systems*, 8(1), pp. 59–82.
72. Mourtzis, D., Angelopoulos, J. and Panopoulos, N., 2022. Digital Manufacturing: The evolution of traditional manufacturing toward an automated and interoperable Smart Manufacturing Ecosystem. In *The digital supply chain* (pp. 27–45). Elsevier.
73. Paul, B. and Rao, M., 2022. Zero-trust model for smart manufacturing industry. *Applied Sciences*, 13(1), p. 221.
74. Billey, A. and Wuest, T., 2024. Energy digital twins in smart manufacturing systems: A case study. *Robotics and Computer-Integrated Manufacturing*, 88, p. 102729.
75. Corallo, A., Crespino, A.M., Lazoi, M. and Lezzi, M., 2022. Model-based Big Data Analytics-as-a-Service framework in smart manufacturing: A case study. *Robotics and Computer-Integrated Manufacturing*, 76, p. 102331.
76. Gericke, G.A., Kuriakose, R.B., Vermaak, H.J. and Madsen, O., 2021, November. Creating a decentralized communication protocol for SMART manufacturing units within industry 4.0. In *Proceedings of International Conference on Data Science and Applications: ICDSA 2021*, Vol. 1 (pp. 731–741). Singapore: Springer Singapore.
77. Gericke, G.A., Kuriakose, R.B. and Vermaak, H.J., 2023, September. A Communication Architecture Approach for Mitigating Complexities in Smart Manufacturing Units by Means of Information Appropriateness. In *World Conference on Information Systems for Business Management* (pp. 91–98). Singapore: Springer Nature Singapore.
78. Sparrow, D.E., 2021. "An Architecture for the Integration of Human Workers into an Industry 4.0 Manufacturing Environment" (Doctoral dissertation, Stellenbosch: Stellenbosch University).
79. Dhanaraj, R.K., Balusamy, B., Samuel, P., Bashir, A.K. and Kadry, S. eds., 2024. Digital Twins in Industrial Production and Smart Manufacturing: An Understanding of Principles, Enhancers, and Obstacles. John Wiley & Sons.
80. Malburg, L., Hoffmann, M. and Bergmann, R., 2023. Applying MAPE-K control loops for adaptive workflow management in smart factories. *Journal of Intelligent Information Systems*, 61(1), pp. 83–111.

81. Ochoa, W., Larrinaga, F. and Pérez, A., 2023. Context-aware workflow management for smart manufacturing: A literature review of semantic web-based approaches. *Future Generation Computer Systems*, 145, pp. 38–55.
82. Gehlot, A., Mohan, L.R., Gupta, A., Anandaram, H., Alanya-Beltran, J. and Chakravarthi, M.K., 2022, July. Smart online oxygen supply management though Internet of Things (IoT). In *2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)* (pp. 1–6). IEEE.
83. Flores-Martin, D., Berrocal, J., García-Alonso, J. and Murillo, J.M., 2023. Towards dynamic and heterogeneous social IoT environments. *Computing*, 105(6), pp. 1141–1164.