# An industrial analytics methodology and fog computing cyber-physical system for Industry 4.0 embedded machine learning applications


Peter O'Donovan Dip MSc

113223892

SCHOOL OF ENGINEERING
NATIONAL UNIVERSITY OF IRELAND, CORK


Thesis submitted for the degree of Doctor of Philosophy

3rd April 2018

**Supervisor:** Dr. Dominic O'Sullivan

**Head of School:** Professor Liam Marnane

# Declaration

This is to certify that the work I am submitting is my own and has not been submitted for another degree, either at University College Cork or elsewhere. All external references and sources are clearly acknowledged and identified within the contents. I have read and understood the regulations of University College Cork concerning plagiarism.

Peter O'Donovan

# Acknowledgements

I would like to thank my supervisor Dr. Dominic O'Sullivan for the opportunity to undertake this research, and providing the guidance, support and expertise needed to complete the PhD journey. In particular, his open door policy produced many engaging discussions, which greatly influenced the direction and contributions of this thesis.

*"The superior man is modest in his speech, but exceeds in his actions."*
**Confucius**

In addition, I wish to acknowledge the efforts of Prof. Brian O Gallachoir, Prof. Jerry Murphy, and Prof. Liam Marnane, which have focused on producing world-class teaching and research environments for postgraduate researchers in the School of Engineering at University College Cork.

*"If everyone is moving forward together, then success takes care of itself. "*
**Henry Ford**

On a personal note, I would like to thank my mother (Breda O'Donovan) and father (John J.D. O'Donovan) for instilling me with resilience and determination, which have proved my most valuable tools when traversing life's challenges. Indeed, I have learned to accept that overcoming life's greatest challenges depends largely on one's ability to continually move forward despite the final destination being unclear.

*"You can't connect the dots looking forward, you can only connect them looking backwards. So you have to trust that the dots will somehow connect in your future. You have to trust in something - your gut, destiny, life, karma, whatever."*
**Steve Jobs**

Last but not least, I must acknowledge my wife (Laura O'Donovan) and three beautiful daughters – Sophie, Kate and Ciara O'Donovan – for their continued love and support during the PhD journey. Although much satisfaction can be derived from academic and professional accomplishments, my greatest achievement shall always be being a father.

*"Not every successful man is a good father. But every good father is a successful man."*
**Robert Duvall**

# Tables

# Figures

# Chapter 1
## Thesis Introduction

## 1.1 Thesis contributions

The contributions of this thesis inform the *design and implementation of industrial analytics solutions* based on Industry 4.0 principles and heuristics. These intertwined research contributions consist of a (1) *unified design methodology* to describe the socio-technical roles, responsibilities, and processes relating to the development of industrial analytics capabilities within a large-scale manufacturing facility, (2) *industrial data pipeline* to streamline and automate the process of data integration, exploration and model building using traditional industrial information systems and protocols that provide access to time-series operating data, and (3) *industrial cyber-physical system based on fog computing* to enable production-ready machine learning models to be deployed and operationalised to inform real-time decision-making in the factory. The following sections describe the characteristics, usage and benefits of each contribution.

### 1.1.1 Unified design methodology for industrial analytics

**The unified design methodology provides a systematic and structured approach for multidisciplinary teams tasked with delivering industrial analytics**. The particular aspects of the methodology where novelty is claimed relates to the mapping of the entire industrial analytics lifecycle as a closed-loop process, which considers the processes needed to support the creation of high-quality models (section 1.1.2), and the processes needed to enable the deployment of production-ready models in real-time factory operations (section 1.1.3). Other unique aspects of the methodology include the integration of multidisciplinary perspectives to form acceptance and performance criteria for evaluating technical implementations based on Industry 4.0 principles (e.g. decentralised decision-making), internal stakeholder concerns (e.g. resilience, security etc.), communication latency and reliability, and maturity benchmarking.

### 1.1.2 Industrial data pipeline for automated time-series processing

**The industrial data pipeline provides factory-to-cloud integration to automate pre-processing of time-series data, and present the analytics-ready data to engineering and analytics personnel.** The particular aspect of the industrial analytics pipeline where novelty is claimed relates to the orchestration of factory and cloud compute modules, which collaborate to ingest, clean and merge several proprietary time-series data logs, before producing a single analytics-ready data set that can be accessed using an open and standard web service interface. After manual data exploration, modelling and testing activities have been undertaken by the engineer or practitioner, the final production-ready analytics model can be deployed to inform real-time decision-making in the factory (section 1.1.3).

### 1.1.3 Fog computing architecture for embedded machine learning

**The industrial cyber-physical system based on fog computing provides the mechanism for deploying and embedding production-ready machine learning models (section 1.1.2) in real-time factory operations.** The particular aspects of the industrial cyber-physical system where novelty is claimed relates to the application of fog computing as an approach for *delivering real-time machine learning within industrial environments*, and addressing prominent *Industry 4.0 design principles* and *stakeholder concerns* relating to emerging technologies and systems. The following section describes details of the implemented fog computing architecture that align with these criteria;

- **Decentralised Intelligence:** emerging systems for Industry 4.0 are expected to support decentralised decision-making, which requires intelligence (e.g. machine learning models) to be embedded and accessible throughout the factory and supply chain. This differs from traditional cloud architectures, where intelligence is persisted and executed from a central location, with results relayed to the necessary distributed components. The fog computing approach facilitates decentralised intelligence by persisting and executing shadow copies of machine learning models on compute/fog nodes, which can be located in close proximity to the relevant factory operations.
- **Near Real-time Performance:** emerging industrial cyber-physical systems are expected to extend and inform existing automation networks, which demands cyber-physical architectures that can handle time-dependent engineering scenarios. Such time-dependent scenarios may be more challenging to address using cloud computing architectures given model execution can depend on the availability and performance of external connectivity (e.g. broadband). However, the fog computing approach greatly reduces dependencies on external connectivity by persisting and executing machine learning models on compute/fog nodes located inside the factory's network.

- **Industrial Data Privacy:** emerging systems for Industry 4.0 are expected to provide appropriate levels of industrial data privacy, which has been traditionally realised using strict governance and firewall policies on industrial automation and controls networks. However, controlling data privacy can become more challenging when using standard cloud computing approaches, given real-time industrial data streams are continuously transmitted outside the factory's network boundaries. The fog computing architecture offers an alternative approach that only requires real-time industrial data to be transmitted to the local compute/fog node, and thereby mitigates the need to transmit real-time industrial data outside the factory's automation and control networks.
- **Openness and Interoperability:** emerging systems for Industry 4.0 are expected to employ open standards, with the intention of promoting system interoperability, service-orientation and reusable intelligence. This contrasts with traditional industrial information systems and technology, which are commonly based on commercial and proprietary technologies. The fog computing approach embraces open internet standards (e.g. HTTP) to ensure models can be executed by third party systems and processes using web service calls, while results (e.g. fault detected) can be propagated to other service interfaces.

## 1.2 Chapter introduction

This chapter discusses the background, motivation, contributions and structure of this thesis. These discussions provide context for future chapters, while also positioning research contributions within the broader Industry 4.0 and engineering domain.

## 1.3 Background and motivation

The Irish Research Council's Enterprise Partnership Scheme (EPS) funded this PhD research. The national research initiative is designed to support industry and academic collaboration, with the intention of addressing real-world challenges using practical and applied research. Given their global reputation as leaders in the field of biomedical device production, DePuy Ireland (Johnson & Johnson) was chosen as the industry partner. This partnership provided access to operational teams, processes and technologies to apply research efforts. These resources were engaged periodically to elicit requirements, survey technologies, and validate approaches. The primary contributors from academic and industrial organisations are summarised in Table 1.

| **Name** | **Domain** | **Contribution** |
|---|---|---|
| Peter O'Donovan<br>*IERG Research Group* | Engineering<br>Informatics | Modelled and developed all theoretical and technology aspects of the industrial analytics architecture and cyber-physical system. |
| Dr. Dominic O'Sullivan<br>*IERG Research Group* | Energy<br>Engineering | Supervised research efforts and provided guidance related to primary engineering systems and principles. |
| Dr. Ken Bruton<br>*IERG Research Group* | Energy<br>Engineering | Supported onsite technology deployment and validated the quality of engineering data needed for technical implementations. |
| Donal Og Cusack<br>*DePuy Ireland* | Operations | Facilitated requests for access to teams, data and systems, and relayed information regarding the organisation's technology roadmap for Industry 4.0. |

**Table 1 Primary research contributors**

The industry partner's motivation for participating in this research centred on developing better insights on emerging industrial analytics technologies (e.g. big data, internet-of-things, and machine learning) and their relationship to Industry 4.0. These insights are important to industry, given the poor availability of standard, formal and prescribed methods for developing industrial analytics capabilities, and nebulous and noisy nature of the commercial market. Indeed, facilities may possess different understandings of industrial analytics. While one facility may claim every data-driven insight demonstrates analytics capabilities (e.g. performance metrics, descriptive statistics etc.), another may hold the view only advanced predictive and prescriptive analytics models (e.g. machine learning) should be classified as such. Although different arguments may be made regarding formal definitions, this thesis considers industrial analytics to centre on advanced predictive analytics models because of their alignment with Industry 4.0 objectives (i.e. moving operations from reactive to predictive). This view of industrial analytics depends on multidisciplinary personnel to apply engineering knowledge and technology to solve real-time operational challenges (e.g. process automation, equipment maintenance etc.), incorporating both legacy and emerging industrial technologies.

### 1.3.1 Multidisciplinary engineering

The manifestation of Industry 4.0 operations depends on the emergence of multidisciplinary engineers that can computationally encode principles of mechanical, process and electrical engineering, using models and methods from computer science, software engineering and statistics, to name a few. Although the computational encoding of engineering knowledge can be observed in industry (e.g. control logic running on automation networks), the complexity associated with Industry 4.0 environments shall impose significantly higher demands in terms of scalability (e.g. larger operational data), integration (e.g. factory-to-cloud communications), latency (e.g. real-time control), and technologies (e.g. legacy and emerging technologies). Indeed, these emerging challenges have resulted in multidisciplinary expertise being highlighted as prominent impediment to Industry 4.0 adoption. Hence, this thesis possesses strong themes of multidisciplinary research, borrowing and applying concepts from the fields of industrial engineering, software engineering, computer/data science and information systems technologies.

### 1.3.2 Industrial predictive analytics

The primary goal of Industry 4.0 is to deliver self-adaptive and self-configuring manufacturing operations. Achieving this goal depends on the creation of collective intelligence, which shall be derived from many industrial analytics models (e.g. energy optimisation, prognostics, inventory etc.) distributed across the factory. However, although the manufacturing domain is engulfed with hype regarding the high-level benefits of industrial analytics and Industry 4.0, widely accepted technical roadmaps and transformations are less prominent. These artefacts are necessary to provide technical foundations (e.g. formal methodologies and architectures) for developing industrial analytics capabilities. Without such technical foundations, industrial analytics processes and pipelines may become prohibitively expense and time-consuming, stemming from inadequate technical integration, interoperability and performance.

Although the term '*industrial analytics*' may be subject to many definitions, when used in the context of Industry 4.0, industrial analytics generally refers to advanced methods supporting data-driven prediction (e.g. machine learning) and scenario simulation, which can enable smart manufacturing transformations (e.g. reactive to predictive operations). Hence, the industrial analytics aspects of this thesis focus exclusively on the development and deployment of machine learning models for Industry 4.0.

### 1.3.3 Industrial cyber-physical systems

The emergence of Industry 4.0 is entirely dependent on the design and development of cyber-physical systems. These systems enable objects and processes from the factory (i.e. physical world) to be computationally virtualised (i.e. cyber world), and subjected to numerous prediction and simulation scenarios (e.g. energy optimisation, remaining useful life) to inform operational decision-making. In many respects, cyber-physical systems extend current automation and control networks, where additional compute resources enable the delivery of real-time industrial analytics (e.g. predictive or prescriptive analytics), which may otherwise be difficult to achieve using control logic deployed on a single controller.

Importantly, one should appreciate the relationship between industrial analytics and industrial cyber-physical systems. While industrial analytics focuses on building models that encode engineering knowledge (e.g. fault prediction), industrial cyber-physical systems comprise the infrastructure, methods and technologies that enable industrial analytics models to be embedded in real-time factory operations. Thus, industrial analytics models can be developed and executed independent of industrial cyber-physical systems, but do not adhere to many Industry 4.0 design principles and guidelines, which emphasise real-time, open and interoperable decision-making.

## 1.4 Case study: Industry 4.0 AHU monitoring

The design methodology, industrial analytics architecture and cyber-physical system presented in this thesis were deployed to the industrial partner's large-scale manufacturing facility, with the intention of demonstrating and validating the proposed approaches, while also deploying an Industry 4.0 aligned technical solution that can facilitate energy engineering applications. In addition to being central to Industry 4.0 objectives, the potential benefits and awareness of industrial energy efficiency have been widely circulated. Some of these reports suggest that buildings alone account for between 20% and 40% of the world's total energy consumption, while industrial AHU's on average account for 40% of an industrial sites total energy consumption [1]. These high energy consumption levels may be attributed to quality standards and regulations associated with many industrial processes (e.g. cleanrooms) that must comply with stringent international standards [2]. The combined industrial and commercial usage of AHU's accounts for between 10% and 20% of total energy consumption in developed countries [1]. Given the significant energy consumption of AHU's, the primary energy case study in this thesis focuses on the application of an industrial analytics architecture and cyber-physical system to facilitate the identification of AHU operating faults using embedded industrial machine learning applications.

In general, the energy performance of buildings rarely meet the performance levels suggested by the design phase. This may be due to poor equipment selection, incorrect installation, inadequate commissioning, or improper maintenance for large-scale AHU's and HVAC Systems [3]. Hence, studies addressing these issues have been able to demonstrate that 20% to 30% energy savings can be achieved by recommissioning AHU operations to eradicate operating faults [4], while other studies focused on on-going commissioning of building systems for peak efficiency have reported savings of an average of 20% of total energy cost [3]. Where both recommissioning and on-going commissioning approaches have been employed, studies have reported 44% savings on electricity consumption and 78% savings on gas consumption over a 10-year period.

Figure 1 illustrates common components an AHU that are used to maintain environmental conditions and thermal comfort for the space. To begin with, air enters the unit either from outside, or recirculated from the space (i.e. return air). Once the air enters the unit the supply fan pulls the air through the unit to supply the space. As the air passes each component it is treated to meet target conditions (e.g. maintain temperature of 20 degrees), which may include filtering, heating or cooling the air before supplying air to the space etc. The components that should be active in the unit at any particular point-in-time can be inferred from the AHU operating mode.

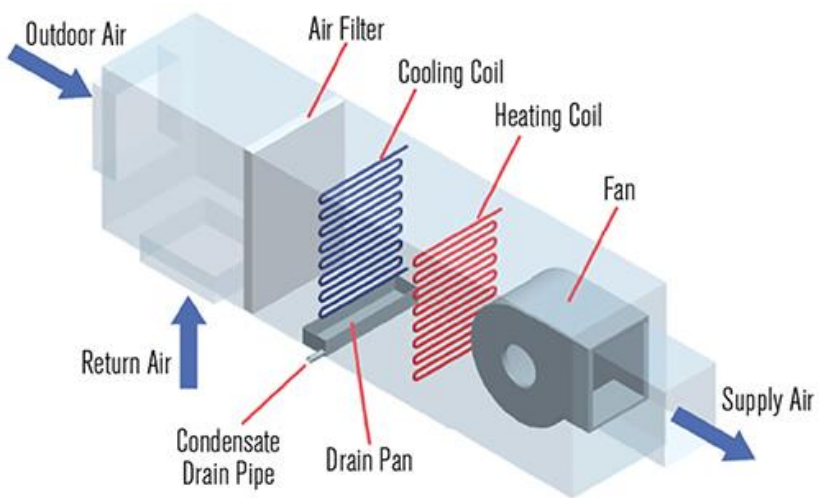



**Figure 1 AHU schematic with primary components**

Figure 2 illustrates logical control transitions from heating to cooling operations that show four discrete AHU operating modes, which change in response to an increasing outdoor temperature (from left to right). At the extreme left of the illustration the outside temperature may be low relative to the internal temperature setpoint, and therefore the AHU's heating components are engaged. As the outside temperate rises and becomes closer to the desired temperature setpoint, the AHU's heating components are deactivated as the outside and return air can be mixed to supply the space. Finally, when the outside temperature exceeds the temperature setpoint cooling components are engaged along with the maximum outside air, while further increases in the outside temperature results in the outside air being minimised, and cooling components being maximised. The four AHU operating modes are summarised below;

- **Mode 1:** engage heating components while only incorporating minimum outside air to meet circulation requirements for the space.
- **Mode 2:** combine the outside and return air without engaging heating or cooling components.
- **Mode 3:** combine the maximum outside air with some cooling operations.
- **Mode 4:** engage cooling components while only incorporating minimum outside air to meet circulation requirements for the space.

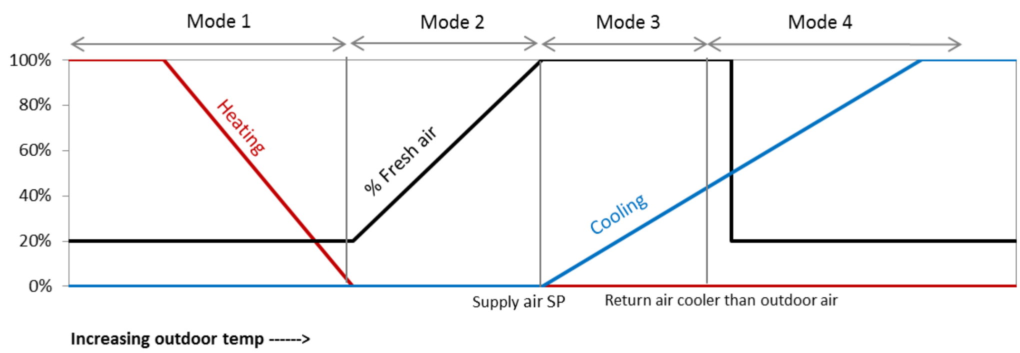



**Figure 2 AHU control sequence and operating modes**

A building energy management system commonly monitors the performance of AHU's in manufacturing facilities. These information systems receive inputs from sensors residing in the AHU, and raise alarms when upper or lower limits of operations are breached. However, these alarms are basic triggers that do not indicate faulty or improper operation. The control logic that governs the operation of AHU's enforces self-adjusting operation, whereby the improper actions of one faulty component may be compensated by other components. Figure 3 illustrates how hot water energy can be wasted due to a faulty heating coil control valve. Given the control reading indicate both heating and cooling coils are closed (i.e. 0%), the temperature of the outside air should not change after passing the coils. However, the illustration shows there is a 5 $^{O}$C rise in temperature, and therefore the heating coil control value can be assumed to be physically open despite the reading indicating otherwise. Such faults can go unnoticed for long periods of time due to the AHU being able to use the cooling coil simultaneously to counteract the heating, and maintain control of the setpoint.

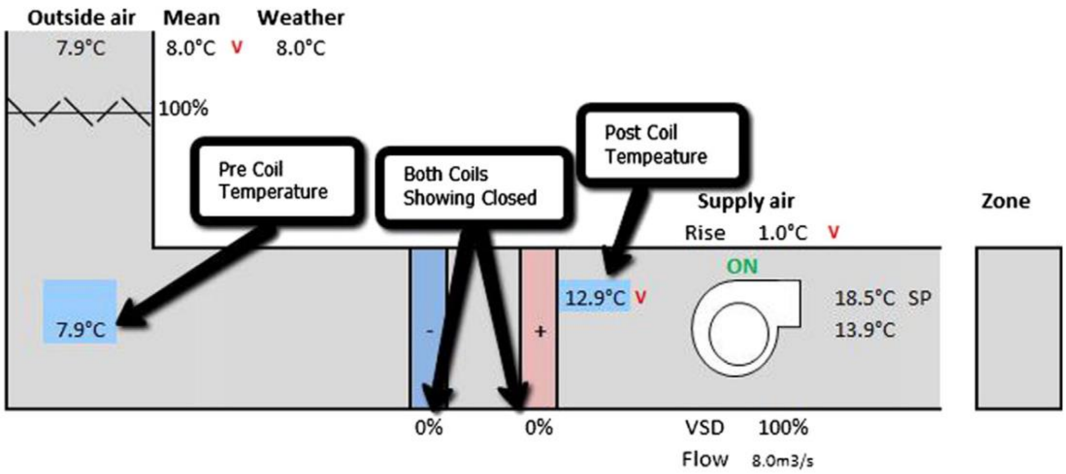



**Figure 3 Example of undetected fault in a heating coil component**

A machine learning model using the Support Vector Machine (SVM) algorithm was created to identify heating component faults for this case study. This model was trained using labelled AHU time-series data from previous research, which was previously undertaken by other IERG research engineers - Dr. Ken Bruton and Dr. Dominic O'Sullivan. Thus, this thesis does not claim any novelty or contribution of the AHU diagnostics or prognostics, but rather focuses on the methods and technical architecture to facilitate the development and deployment of these models for large-scale Industry 4.0 operations. These contributions include the development of an industrial analytics architecture comprising (a) an industrial data pipeline to automate the batch ingestion, cleaning and presentation of time-series data for building machine learning models, and (b) an industrial cyber-physical system to embed production-ready machine learning models in real-time factory operations and decision-making.

## 1.5 Research objectives

The primary objective of this research was to develop guidelines, theories and technical architectures for Industry 4.0 embedded machine learning applications. Figure 4 illustrates common design and implementation properties for industrial analytics and cyber-physical systems, which serves to highlight differences between current approaches, and those proposed by this research. The current design and implementation properties are predominantly based on observations from industry engagement, while the proposed approaches represent the primary objectives and impact of this research. Although one may argue current approaches observed through industry engagement may not be representative of other large-scale manufacturing facilities, each of the current design and implementation properties have also been identified from the literature (e.g. centralised cloud-based intelligence is much more common than decentralised intelligence).


**Figure 4 Comparison of current and proposed industrial analytics approach**

The following points elaborate on the differences between current and proposed industrial analytics approaches depicted in the illustration;

- **Perspective** refers to design concerns of those developing the industrial analytics infrastructure, with current approaches embodying design perspectives from specific disciplines (e.g. technology, process engineering etc.), while the proposed approach encourages design perspectives that incorporate concerns from multiple disciplines.

- **Methodology** relates to the underlying methods and processes informing the design of the industrial analytics infrastructure, with current approaches adopting ad hoc practices, while the proposed approach promotes the idea of formal and systematic methods.

- **Architecture** describes technical components and relationships for the industrial analytics infrastructure, with current approaches depicting hierarchal technology layers, while the proposed approach proposes a closed-loop lifecycle architecture that traces data flows throughout the factory.

- **Guidelines** inform the development of high-level design requirements and ideologies for the industrial analytics infrastructure, with current approaches influenced by internal organisation-level policies, guides and personnel, while the proposed approach adopts Industry 4.0 design principles.

- **Evaluation** relates to the procedures used to assess the industrial analytics infrastructure, with current approaches focused on commercial technology acquisition and feature availability, while the proposed approach encourages the use of performance metrics and assessments.

- **Intelligence** refers to where the primary computation and decision-making is undertaken in the industrial analytics infrastructure, with current approaches favouring central intelligence and processing (e.g. cloud server), while the proposed approach encourages decentralised intelligence and processing.

- **Workflows** depict the processes associated with the development and deployment of industrial analytics models, with current approaches depending on manual and human-assisted methods, while the proposed approach encourages automation using technology.

- **Technology** describes behaviours towards technology adoption, with current approaches utilising commercial and proprietary solutions, while the proposed approach adopts the ideology of open and standards-driven technology.

- **Integration** refers to the methods underpinning system interconnectivity and interoperability, with current approaches depending on custom integration routines, while the proposed approach considers the application of standard programmatic interfaces.

- **Analytics** relates to the delivery and usage scenarios for industrial analytics models, with current approaches applying analytics on batch (i.e. historic) operational data from standalone computers, while the proposed approach promotes the use of embedded industrial analytics in the factory to positively affect real-time decision-making and operations.

## 1.6 Research process

This thesis employed an action research process, which incorporates perspectives of participants and researchers to deliver applied contributions [5]. The action research process supports the expansion of scientific knowledge through the development of real-world and practical solutions, which can be a useful approach in contemporary or underdeveloped fields [6], [7]. Hence, action research typically begins with concepts and ideas rather than fixed hypotheses [8] – e.g. considering 'how' something may be achieved or applied, rather than 'why' a particular phenomenon occurs. However, action research can deliver highly relevant and insightful outcomes relating to practical problems [9], and can be considered an appropriate process to bridge potential gaps between academic research and industrial practices [6]. In particular, action research processes have been used extensively to facilitate technology prototyping, participant engagement, and field-based observations [9].

Of course, similar research processes can be found in the literature. Figure 5 illustrates prominent interpretative methodological processes referred to as the 'pillars of information system research' – consisting of development research, action research and grounded theory [6]. Given the similarities between these processes, choosing a particular approach does not appear to be an exact science. However, action research was chosen to guide this research given its suitability for developing new theories [8], bridging gaps between theory and practice [8], implementing innovative technologies [6], investigating complex events and processes [10], empowering the development of unique perspectives [8], and identifying new logic [11].

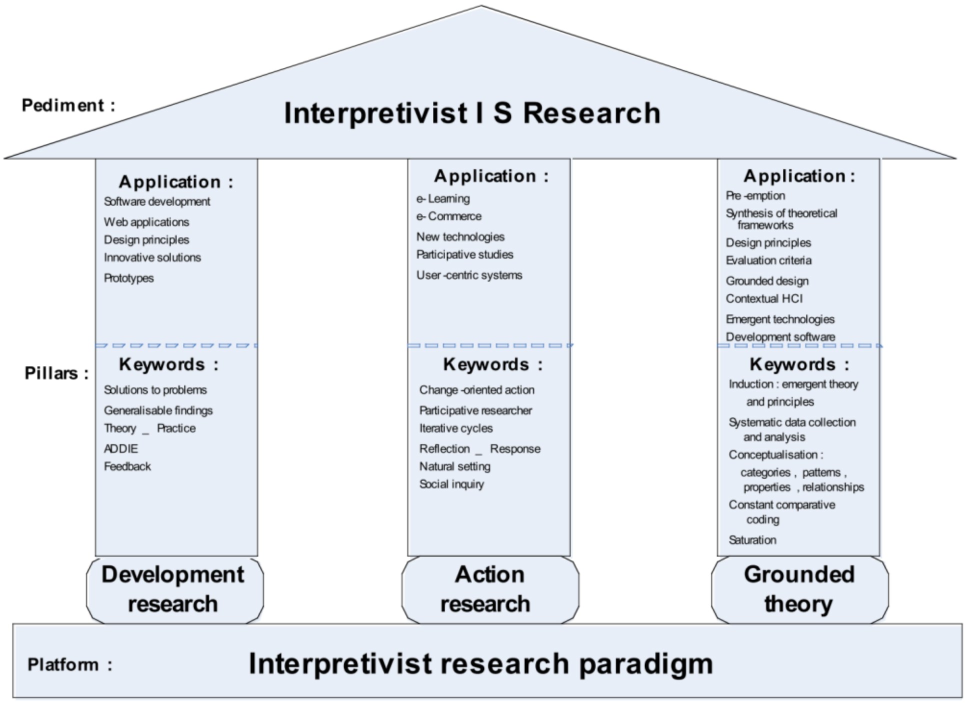



**Figure 5 Pillars for technology and information system research** [6]

Figure 6 illustrates five phases comprising the action research process, while these phases are summarised in Table 2. The process begins with the identification of a problem, objective or situation that requires action (e.g. produce new theory or process [6]). Thereafter, subsequent phases guide research efforts from problem identification (i.e. diagnosing) to definitive findings (i.e. specifying learning). These phases are undertaken iteratively, with the findings from each iteration informing the next iteration (e.g. omitting ineffective solutions), which improves and refines the technology, theory or system being developed [8]. In essence, this type of iterative research process can naturally be classified as longitudinal analysis [6].

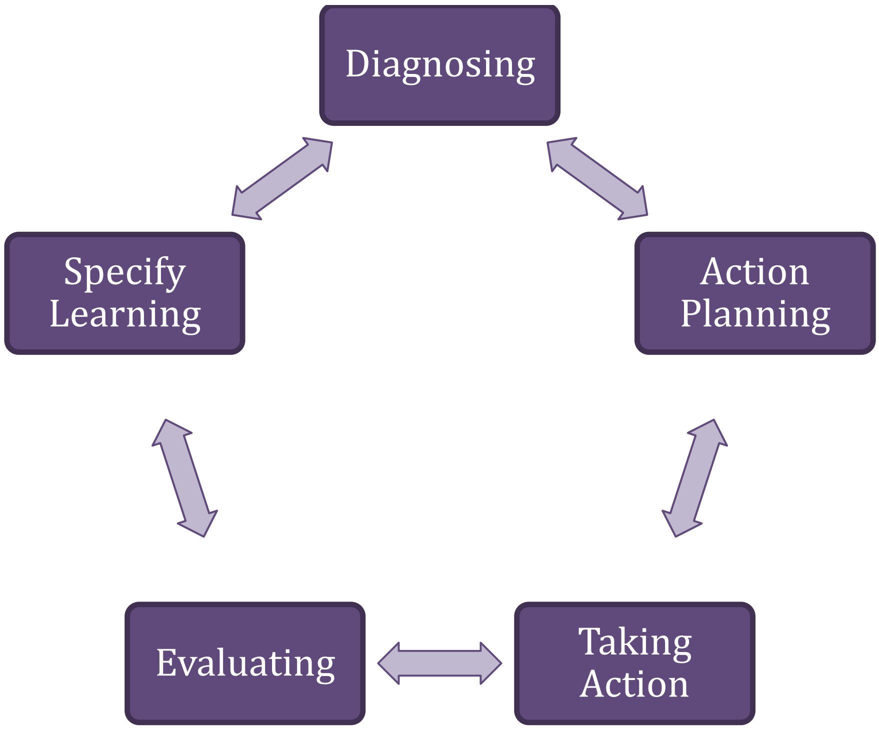



**Figure 6 Action research process model**

| Phase | Description |
|---|---|
| Diagnosing | This phase focuses on the definition or identification of the problem to be solved or investigated. |
| Action Planning | Once the problem has been identified, this phase focuses on considering possible solutions or approaches. |
| Taking Action | Given several possible solutions, this phase focuses on choosing and applying a particular solution to the problem. |
| Evaluating | After a particular solution has been applied to the problem, this phase considers the consequence s of the applied action. |
| Specify Learning | The final phase outlines general findings of the process, which may indicate further iterations of the process are needed to solve the problem being investigated. |

**Table 2 Phases of action research process**

## 1.7 Publications

The following section specifies the academic publications associated with this thesis.

### 1.7.1 Journal articles

| Year | Title | Journal | Reference |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 2015 | Big data in manufacturing: a systematic mapping study | Journal of Big Data | [12] |
| 2015 | An industrial big data pipeline architecture for smart manufacturing | Journal of Big Data | [13] |
| 2016 | Case study: the implementation of a data-driven industrial analytics methodology and platform for smart manufacturing | Journal of Prognostics and Health Management | [14] |
| 2016 | IAMM: a maturity model for measuring industrial analytics capabilities in large-scale manufacturing facilities | Journal of Prognostics and Health Management | [15] |
| 2017 | A fog computing based Industry 4.0 cyber-physical system for open and embedded analytics | Manufacturing Letters | [16] |
| 2018 | A systematic review of industrial cyber-physical system design and implementation dimensions | Journal of Manufacturing Systems | **Submitted** |
| 2018 | A performance analysis and evaluation of cloud and fog computing cyber-physical interfaces for Industry 4.0 | Journal of Manufacturing Systems | **Submitted** |

**Table 3 Lead author journal articles**

### 1.7.2 Conference proceedings

| Year | Title | Conference | Reference |
|---|---|---|---|
| 2015 | An industrial big data pipeline for Prognostics and Health Management (PHM) | PHM Society Annual Conference | [17] |
| 2017 | A systematic mapping of cyber-physical systems research for Industry 4.0 | International Manufacturing Conference | [18] |

**Table 4 Lead author conference proceedings**

### 1.7.3 Other publications

| Year | Title | Type | Title | Reference |
|---|---|---|---|---|
| 2015 | Enabling effective operational decision-making on a combined heat and power system using the 5C architecture | Journal | Procedia CIRP | [19] |
| 2016 | Design and development of a software tool to assist ISO | Journal | Journal of Engineering Manufacture | [20] |

| | | | | |
|---|---|---|---|---|
| | 50001 implementation in the manufacturing sector | | | |
| 2017 | Automatically Identifying and Predicting Unplanned Wind Turbine Stoppages Using SCADA and Alarms System Data: Case Study and Results | Journal | Journal of Physics | [21] |
| 2018 | Development and application of a machine learning supported methodology for measurement and verification (M&V) 2.0 | Journal | Energy and Buildings | [22] |

**Table 5 Contributing author publications**

## 1.8 Thesis layout

The following points summarise the remaining chapters of this thesis;

- **Chapter 2** presents a literature review comprising a thematic investigation of smart manufacturing, and a systematic analysis of the methods pertaining to the design and implementation of Industry 4.0 cyber-physical systems.
- **Chapter 3** presents a unified design methodology for developing industrial analytics architectures and infrastructures, which are aligned with common stakeholder concerns, and Industry 4.0 design principles.
- **Chapter 4** presents the application of the design methodology in a large-scale manufacturing facility, and the demonstration of fog computing as a means of embedding machine learning models with real-time automation and control networks using cyber-physical interactions.
- **Chapter 5** presents the performance results (e.g. latency) from stress testing cloud and fog computing cyber-physical interfaces, and discusses aspects of the demonstrated implementation aligned with Industry 4.0.
- **Chapter 6** presents the conclusions derived from this research, and relates these findings to the primary research objectives.

# Chapter 2
# Literature Review

## 2.1 Chapter introduction

This chapter reviews the emerging multidisciplinary engineering field of *Industrial Cyber Physical Systems*, which shares many similarities with *Industrial Internet-of-Things*. Although both terms may be used interchangeably [23], this chapter uses *Industrial Cyber Physical Systems* given its adoption in the fields of industrial systems and engineering, while many legacy sensing and automation technologies do not naturally support internet messaging and data exchanges. Therefore, this chapter takes the view that contemporary industrial cyber physical systems comprise both internet-of-things and legacy operation technology, while also possessing formal methods and approaches relevant to industrial engineering applications (e.g. equipment maintenance).

## 2.2 Cyber-physical systems

Industrial cyber-physical systems enable objects and processes residing in the physical world (e.g. manufacturing facility), to be tightly coupled with compute, communication and control systems in the cyber world [24]. Cyber-physical interfaces promote data transmissions between both worlds using numerous technologies, including wireless sensors, phones, tablets, and web services, to name a few [25]. Conceptually, these cyber-physical interfaces result in the manifestation of 'cyber twins', where each physical object in the real world, exists as a virtual entity in the cyber world. In turn, these virtual entities may be individually and/or collectively analysed, interrogated and simulated to derive operational insights and inform better decision-making.

The emerging network paradigm promising to bridge industrial physical and cyber worlds is that of the internet-of-things, which comprises internet-enabled devices and gateways to sense, collect, send and receive data [26]. In terms of manufacturing, this may include interactions with sensors, controllers, actuators, radio-frequency-identification (RFID) tags, global positioning systems (GPS), and high-definition cameras [26], to name a few. Naturally, these broad and pervasive interactions produce large data repositories (i.e. big data) describing factory operations [24]. Where sufficient high-quality data has been compiled, these datasets can be analysed using machine learning to make useful predictions (e.g. predict equipment failures). Figure 7 illustrates Google search trends between 2012 and 2016 for internet-of-things and big data, which clearly shows a convergence in Q4 2015. Theoretically, internet-of-things would be impeded where big data technologies did not exist.

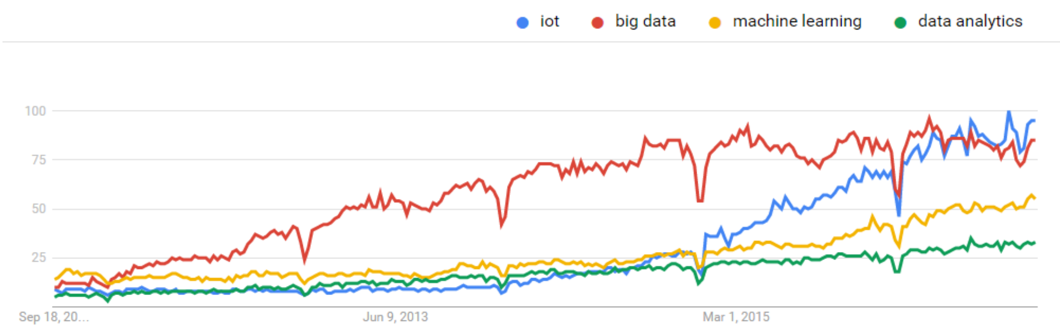



**Figure 7 Google trends for cyber-physical system technologies captured September 2016**

The remainder of this chapter explores the relationship between Industry 4.0 (i.e. smart manufacturing), industrial cyber-physical systems, and the multidisciplinary computing and engineering disciplines that inform the field. The following points summarise the three main sections of this chapter;

1. **Motivation and adoption.** Investigating the business motivation for adopting industrial cyber-physical systems (e.g. Industry 4.0).

2. **Research challenges and directions.** Assessing the prominent current and future research challenges for industrial cyber-physical systems.

3. **Multidimensional systematic analysis.** Synthesising literature pertaining to industrial cyber-physical systems to describe multidisciplinary perspectives and methods relating to design, application, implementation and standards.

## 2.3 Motivation and adoption

The adoption of cyber-physical systems is motivated by the desire to implement *smart manufacturing* operations. The term smart manufacturing refers to an emerging data-driven paradigm focused on the creation of manufacturing intelligence using real-time pervasive networks and data streams in the factory [27]–[30], with experts predicting that it may become a reality in the next 10 to 20 years. The overarching objective of smart manufacturing is similar to traditional manufacturing and business intelligence, which is responsible for transforming low-granularity raw data, to high-granularity actionable and insightful information. Such transformations are an important aspect of operational intelligence, given low-granularity operational data (e.g. time-series) is difficult to interpret, thereby making it inadequate for timely decision-making. In contrast, high-granularity information provides easily interpretable knowledge that can positively impact operations using assistive decision-making. However, smart manufacturing differs from traditional manufacturing intelligence given its extreme focus on seamless operating intelligence, where real-time collection, aggregation and sharing of knowledge across physical and computational processes to derive a self-optimising production environment [31]. Essentially, smart manufacturing ensures every aspect of the factory is monitored, optimised and visualised [27], [31], [32].

Table 6 outlines operating and technology differences between traditional and smart manufacturing facilities, with the former characterised by precise and reactive operations, and the latter embodying intelligent and predictive operations capable of self-optimisation and self-configuration. This self-based intelligence may discover and execute computations that would be too complex or obscure for personnel to model using traditional methods [24].

| | **Data Source** | **Today's Factory** | | **Industry 4.0** | |
|---|---|---|---|---|---|
| | | **Attributes** | **Technologies** | **Attributes** | **Technologies** |
| Component | Sensor | Precision | Smart sensors & fault detection | Self-aware<br>Self-predict | Degradation monitoring & remaining useful life |
| Machine | Controller | Producibility & performance | Condition-based Monitoring & Diagnostics | Self-aware<br>Self-predict<br>Self-compute | Up-time with predictive health monitoring |
| Process | Network | Productivity & OEE | Lean operations: work and waste reduction | Self-configure<br>Self-maintain<br>Self-organise | Worry-free productivity |

**Table 6 Comparison between today's factory and smart manufacturing** [24]

During the transition to smart manufacturing, some high-level business transformations can be expected. Examples of these transformations are provided below;

- **Knowledge-embedded** operations enabled by information and engineering systems possessing the knowledge and intelligence for 'smart' operations.
- **Predictive and preventive** operations replacing operations based on reactive and responsive decision-making.

- **Performance-based** operations receiving more attention, with an emphasis on minimising energy and material usage, while maximising sustainability, health and safety, and economic competitiveness.
- **Distributed intelligence** focusing on the goals and objectives of the entire organisation, rather than vertical and isolated decision-making.
- **Multidisciplinary workforces** derived from engineering, computing, and statistical disciplines, which are capable of delivering smart infrastructures and engineering applications.
- **Convergence of information and operation technology departments** to ensure legacy technology does not impede the adoption of emerging technologies for smart manufacturing.

There are many challenges that can impede these transformations, including legacy systems, proprietary technologies, quality assurance, regulatory enforcement and technical resources, to name a few. Arguably, the dependency on multidisciplinary expertise encompassing engineering, computing, analytics, design, planning, automation, and production represents the greatest challenge [33], [34]. Given these technical and personnel challenges, smart manufacturing is considered too complex for any single organisation to address [35]. Thus, several groups and initiatives were formed to support smart manufacturing adoption.

### 2.3.1 Industry initiatives and groups

Currently, there are numerous government, academic and industry groups promoting and supporting smart manufacturing. The most prominent of these include the Smart Leadership Coalition (SMLC) [31], Technology Initiative SmartFactory [36], Industry 4.0 [37], and The Industrial Internet Consortium (IIC). Given the contemporary, qualitative and multi-dimensional nature of smart manufacturing, some aspects of these initiatives may employ different terminology, but share an overarching vision of real-time, pervasive and data-driven intelligence for optimising factory operations. Arguably, SMLC and Industry 4.0 represent the most recognised smart manufacturing initiatives, with each loosely coupled to their geographical origins (i.e. US and EU). Figure 8 illustrates Google search trends for these initiatives between 2012 and 2016, with Industry 4.0 the most popular term since Q4 2014.

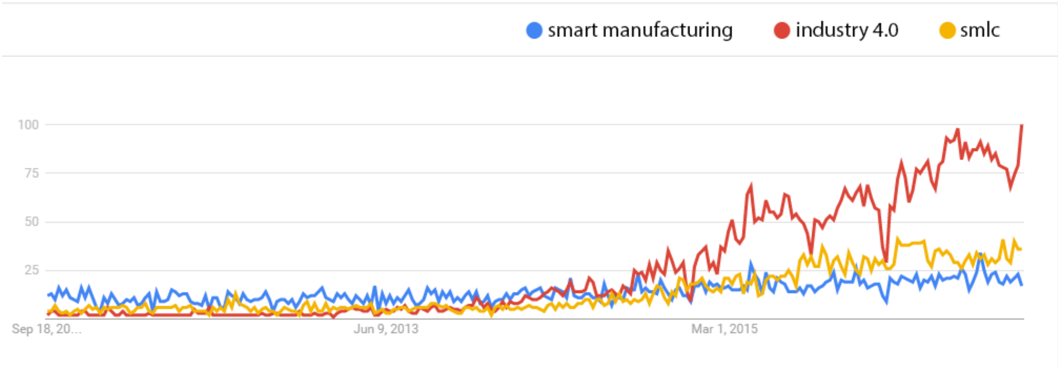



**Figure 8 Google trends for smart manufacturing captured September 2016**

The SMLC working group comprises academic institutions, government agencies and industry partners. This diversification may filter bias perspectives, while ensuring challenges relevant to the broader manufacturing community are addressed. In terms of tangible contributions, the SMLC has produced theoretical assets, such as technology roadmaps, recommendations and guidelines, as well as technology artefacts, including an open smart manufacturing platform (SM Platform) and an industrial marketplace that facilitates plug-and-play deployment of smart manufacturing applications.

Industry 4.0 refers to a high-tech strategy developed by the German government to highlight the economic benefits of smart manufacturing. The term Industry 4.0 stems from a logical naming convention referencing each industrial revolution, with 4.0 referring to an anticipated fourth revolution (i.e. smart manufacturing), which shall be enabled by cyber-physical systems, comprising ubiquitous sensing, simulation and analytics. Following the same logic, previous industrial revolutions are labelled 1.0, 2.0 and 3.0. The first industrial revolution (Industry 1.0) was brought about by the availability of water and steam power, which enabled mechanical production processes, with the first mechanical loom employed in 1784. The second industrial revolution (Industry 2.0) was brought about by the availability of electricity, which facilitated the advent of mass production processes and the division of labour. Finally, the third industrial revolution (Industry 3.0) was brought about by advanced electronics, which enabled control networks to automate production processes, with the first programmable logic controller (PLC) introduced in 1969.

### 2.3.2 Technical roadmap and adoption

There are three logical phases of smart manufacturing adoption, and cyber-physical system implementation, with each phase providing benefits exponentially more beneficial than the one previous [38]. These phases are summarised below;

- **Phase 1 - data integration and contextualisation.** Initially, facilities evaluate data availability across the factory (e.g. sensors, controllers, databases etc.) to form a global understanding of data assets. These integrations are typically complex and time-consuming, accounting for up to 90% of adoption effort where multiple legacy devices, systems and protocols exist. Once data has been consolidated during this phase, facilities may experience a positive impact on operating costs, health and safety, and environmental factors given data discovery and accessibility improvements.
- **Phase 2 - simulation, modelling, and analytics.** Given the availability of high-quality accessible operational data, facilities interrogate these datasets to build models that describe, predict or prescribe intelligent actions that can positively impact operations. Once these models have been constructed and validated, facilities may realise many operational efficiencies, such as improved production rates or product customisation.
- **Phase 3 - process and product innovation.** Where large repositories of manufacturing intelligence (i.e. models) exist, 'game changing' insights for process and product innovation may emerge. These insights differ from those of previous phases, given they are based exclusively on the discovery of new knowledge (e.g. correlation unknown to engineering first principles). Once facilities attain such insights, they may realise exponential increases in operational efficiencies capable of disrupting entire markets.

Most facilities adopting smart manufacturing must navigate these phases. Generally, the effort needed to realise each phase, and the benefits derived from each phase, are negatively correlated. The potential benefits from each phase (1 to 3) increases, while implementation effort for each phase (1 to 3) decreases. Therefore, facilities initially experience significant effort during industrial data integration, with modest gains in operational intelligence, while each subsequent phase may require less effort, but demonstrate greater improvements in operational intelligence. This easing effect could be attributed to residual and cumulative technologies, knowledge and skills transferring from previous phases.

Facilitates that realise smart manufacturing will be positioned to address many contemporary business and engineering challenges, such as increased global competition and rising energy costs, while also shortening production cycles and enhancing just-in-time product customization capabilities [31], [34]. Many of these benefits may be associated with demand-driven supply chains, which employ real-time, Internet-aware, collaborative and synchronised technologies, to optimise operational efficiencies. These potential efficiencies include (1) reducing capital intensity by 30%, (2) reducing product cycle times by up to 40%, as well as (3) overarching efficiencies across energy, emissions, throughput, yield, waste, and productivity. Where these factory-level efficiencies are considered in a broader geographical context, smart manufacturing can contribute significantly to the greater economy. Indeed, research from Fraunhofer Institute and Bitkom estimated Industry 4.0 could be worth up to 267 billion Euros to the German economy by 2025 [39].

### 2.3.3 Business and technical impediments

Facilities transitioning to smart manufacturing may encounter numerous technical challenges. These range from real-time technologies and infrastructures needed for self-optimising operations, to the acquisition of multidisciplinary personnel for Industry 4.0 engineering systems (i.e. industrial cyber-physical technology) [31]. Of course, the extent of these challenges may vary from factory-to-factory. For example, challenges facing greenfield (e.g. new facilities) and brownfield (i.e. legacy facilities) sites are quite different [27]. When compared to brownfield sites encumbered by legacy devices, systems and protocols, greenfield sites are better placed to adopt smart manufacturing technology, given they present the opportunity to design and implement industrial cyber-physical technologies from the ground-up. In the case of brownfield sites, simply replacing legacy technology can be difficult for a number of reasons:

- **Historic technology investment.** Over the last 40 years, manufacturing facilities invested in information, control and automation technology to optimise production and business processes. Given this investment, facilities may resist replacing technologies before their end-of-life.

- **Regulatory and quality constraints.** Certain manufacturing facilities (e.g. pharmaceuticals and medical devices) are subject to internal and external regulatory and quality control. These regulations and controls can limit technology choices to pre-approved and risk-assessed technologies. Although processes may exist to amend such policies, the effort and risk associated with doing so may quell initial enthusiasm for technology replacement.
- **Dependency on proprietary systems or protocols.** While industrial and automation standards exist to promote interoperability, their adoption across manufacturing environments can be sporadic. Historically, manufacturing environments have been known to utilise closed, ad-hoc and proprietary environments. These environments exemplify technology lock-in, where the adoption of smart technology largely depends on particular vendor offerings.
- **Weak vision and commitment.** Given the path to smart manufacturing may not always be clear, management and visionaries are needed to guide internal initiatives. Where such leadership does not exist, facilities may struggle to construct a business case for replacing legacy technology.
- **Quality risks and disruption.** Generally, technology projects are considered high-risk due to historic project failures and overruns. Although failed technical implementations may be considered the primary risk, operational impacts and inefficiencies during the period of user training and adoption represent potential secondary risks. These combined risk factors may postpone smart technology adoption until such time that lost opportunities affect competitiveness.
- **Emerging technologies and methods.** Transitioning to smart manufacturing depends on the integration of mainstream and emerging information technology paradigms (e.g. service-oriented architecture, internet-of-things) across industrial environments. However, modern manufacturing facilities are predominantly constructed on operation and automation technology. Therefore, replacing legacy technology with smart equivalents may depend on the attitudes and perspectives of current operation technology personnel.

### 2.3.4 Prominent standards and technologies

Manufacturing environments incorporating standards should demonstrate interoperability and openness. Such standards can be found at different levels in the factory's technology ecosystem, encompassing (a) field devices, (b) automation and control, and (c) enterprise systems [40]. Generally, these standards can decrease the time, risk and cost of system implementation and integration by promoting formal and consistent methods [41]. Therefore, environments embodying standards may experience smoother transitions to smart manufacturing, when compared to those constructed using proprietary or ad-hoc approaches.

#### FIELD DEVICES

Field device standards enable data collection and control across physical devices (e.g. controllers) in the factory. These standards expose software interfaces and architectures for programming control logic and data collection. Common field device standards include OPC, MTConnect, BACnet, Modbus, and LonWorks [42], with each differing in their level of openness, architecture and abstraction.

#### AUTOMATION AND CONTROL

Automation and control standards comprise theories, methods and architectures for designing and implementing process-driven networks. Common standards include ISA-88, ISA-95, MESA, SCOR, and DiRA [43], [44]. Of these standards, ISA-88 and ISA-95 are widely adopted for integrating sensing, control and information systems [45], with ISA-88 focusing exclusively on factory-level integrations, and ISA-95 extending these integrations to the enterprise (i.e. factory-to-enterprise integration).

#### ENTERPRISE SYSTEMS

Enterprise system standards enable data exchanges and system integrations between factory technologies. Many of these standards originate from distributed, cloud, and Internet computing, where open and distributed messaging are fundamental requirements. Common enterprise standards include HTTP, MQTT, SOAP, WSDL, XML, and B2MML, to name a few. Generally, these standards reside within open and module service-oriented architectures, which orchestrate communication and processing between distributed components and systems.

## 2.4 Research challenges and directions

Cyber-physical systems represent the primary enabling technology for smart manufacturing and Industry 4.0, with the design and development of future engineering systems largely dependent on their existence [46]. Indeed, research funding bodies in the EU and US identified the field of cyber-physical systems as a critical component in the technological evolution of many business domains [47]–[50]. These research efforts aim to amalgamate knowledge and principles from computing and engineering disciplines (e.g. networking, software, control, mechanical etc.), while incorporating industry and academic collaboration [46].

Many terms have been used to describe the application of emerging sensing and analytics to industry. These terms include industrial cyber-physical systems (ICPS), cyber-physical production systems (CPPS), and industrial internet-of-things (IIoT), to name a few. Although these emerging paradigms can be applied differently to address particular engineering scenarios, cyber-physical systems typically function in one of two ways, either as a technology (a) delivering decision-making information directly to workers, or (b) enabling automatic self-optimising operations using machine-to-machine communication [23]. Additionally, these approaches can be adopted as stepwise progressions, where initial implementation employs human-assisted decision-making, with the intention of transitioning to automated self-optimising behaviour. However, the ability to achieve self-optimisation may depend on current technology capabilities, quality and regulation constraints, and multidisciplinary expertise.

Cyber-physical systems incorporating legacy control and internet-of-things technology are needed for Industry 4.0 engineering applications, which enable decision-makers visualise real-time data streams throughout the supply chain [26]. Given appropriate insights can be derived from such streams, decision-makers can target operational improvements and efficiencies (e.g. energy consumption, maintenance scheduling, optimised control and equipment maintenance). In the context of cyber-physical systems, operational efficiencies (e.g. self-optimisation) are commonly derived using predictive analytics and computer simulations in the cyber-world (e.g. cloud), before being relayed to the physical-world [24], [26], [51]. High-quality and robust cyber-physical interactions may improve work intensity and resource usage, while providing opportunities to substitute low-skilled and manual processes [23]. However, despite these potential benefits, manufacturing facilities are behind the curve in terms of cyber-physical system adoption [26]. Indeed, studies from the German Federal Ministry of Education and Research, and the Office of Technology Assessment at the German Bundestag, both suggest the broader adoption of cyber-physical system technology remains 'relatively low' [23].

Given industrial cyber-physical systems comprise both legacy and emerging technologies, design and implementation must aim to support current devices and systems until their end-of-life [26], while augmenting automation and control networks to incorporate advanced analytics and real-time cyber-physical interactions [51]. Where these challenges are perceived as unattainable or disruptive, senior management may choose to defer technology adoption [26]. Other potential reasons for deferral include concerns regarding cyber security, commercial sensitivity, and control performance [25].

Due to the contemporary nature of industrial cyber-physical systems, the specification of competencies, architectures, and technologies remain open [23]. Although this openness presents significant research opportunities, insufficient prescription and formalisms complicate implementation efforts, requiring developers and engineers to take full ownership of system integration and technology selection scenarios, of which they may possess limited knowledge. Therefore, much of the current research centres on theoretical and conceptual methods to formalise different aspects of system design and implementation (e.g. architecture, security, control etc.) [25]. However, while such research provides an important foundation, the manufacturing domain needs more applied research demonstrating real-world implementations [26]. The high-level classification of research opportunities in the field are summarised below [46];

1. **Abstractions and architectures** to facilitate the modular design and implementation of cyber-physical systems. These representations aim to seamlessly integrate control, communication and computation, while incorporating heterogeneous systems using a ‘plug-and-play’ approach.
2. **Distributed computing and network control** to address challenges pertaining to time and event management across geographical boundaries (e.g. factory-to-cloud communication latency). These techniques, methods and technologies must address variable time delays, communication failures, decentralised real-time decision-making scenarios, and secure communications.
3. **Verification and validation** methods to ensure hardware and software components are dependable, reconfigurable, trustworthy and certifiable. These methods ensure cyber-physical components can reliably operate according to the specifications of quality, regulatory and safety policies.

These research opportunities must also be considered from multiple perspectives, such as those of process control, information technology, and management. For example, an architecture developed by a control engineer may focus extensively on algorithmic robustness, while a system engineer may choose to focus more on interoperability and data exchanges. Given the highly multidisciplinary nature of industrial cyber-physical systems, contributions from each discipline demonstrate natural bias, which indicates there are no globally accepted requirements, methods and technologies for industrial cyber-physical systems. However, design principles for Industry 4.0 provide a complete set of heuristics upon which to base future specifications. General design principles can be extracted individually from overlaps in the literature, while some publications have made efforts to collate and discuss core principles [52]–[54].

### 2.4.1 Open and consistent architectures

Where diverse manufacturing technology profiles exist, cyber-physical systems may need to accommodate multiple system architectures [25]. These architectures may incorporate components developed using different programming languages (e.g. Java, .NET, C++, Python etc.), and distributed across mobile, cloud and desktop environments, weaving numerous runtime environments in to the fabric of cyber-physical systems. Given such diverse technologies and environments, cyber-physical system implementations must leverage open and neutral technology architectures to promote flexibility, integration and interoperability. These open architectures typically include formal data exchange protocols, to provide consistent contractual guidelines and interfaces for messaging between cyber-physical components. However, current open architectures supporting enterprise systems do not consider the peculiarities and nuances of operation technology, which requires new architectures and technologies to integrate localised automation and control networks with the cyber-world [55].

### 2.4.2 Data management and processing

Fundamentally, data management and processing provides accessible, homogeneous, contextualised, and consistent data models for reporting and analysis [25], [51]. These models may include different types of data, from time-series measurements (e.g. hourly temperature), to contemporary data (e.g. social networks)[26]. However, realising such models can be difficult where participating data streams employ arbitrary naming conventions, or proprietary technologies. In these scenarios, data mapping, labelling and contextualisation must be undertaken to align data from each system [24], [51]. Once data models have been produced, data processing techniques can be used to inform decision-making, generate knowledge, or support real-time operations [56].

Table 7 and Table 8 describe data and processing requirements identified for Industry 4.0, which were extracted from 88 research publications using content analysis [56]. While data requirements demonstrate the need to model, integrate and populate (i.e. content) data to incorporate different dimensions of industrial operations, processing requirements highlight application scenarios (e.g. pattern recognition) dependent on high-quality data availability. Interestingly, these data requirements demonstrate the dimensions of Industry 4.0 integrations, including horizontal approaches for integrating systems across operating divisions (e.g. energy and manufacturing systems), vertical approaches for integrating data from different levels of hierarchical automation networks (e.g. field device, supervisory control etc.), and lifecycle approaches for tracing data flows across all possible dimensions.

| Main Category | Sub Category | Frequency | Requirements Description |
|---|---|---|---|
| Data model | Unify semantics | 15 | Unify information models and meanings |
| | Unify interfaces | 12 | Unify interfaces and communications |
| Data integration | Integrate lifecycle | 10 | Integrate data along the lifecycle of cyber-physical systems |
| | Integrate horizontally | 13 | Integrate data along the value chain and network |
| | Integrate vertically | 11 | Integrate data of the automation pyramid |
| Data content | Include produce data | 3 | Include product data and description |
| | Include process data | 7 | Include production processes data and description |
| | Include business data | 1 | Include business data and parameters |
| | Include sensor data | 12 | Include sensor and actor data from cyber-physical systems |

**Table 7 Data requirements for Industry 4.0** [56]

| Main Category | Sub Category | Frequency | Requirements Description |
|---|---|---|---|
| Decision processing | Ad-hoc networking | 21 | Build networks depending on situation |
| | Optimise network | 18 | Optimise network in local decision-making |
| | Admit autonomy | 8 | Admit autonomy in decision-making of cyber-physical systems |
| | Utilise models | 13 | Utilise comprehensive models of real production |
| | Monitor conditions | 13 | Monitor, diagnose and perform actions online |
| Knowledge representation | Detect patterns | 7 | Detect patterns and similarities in production |
| | Prepare data | 7 | Prepare, compile and filter data |
| | Transform know-how | 6 | Transform know-how and expert knowledge |
| | Predict parameters | 7 | Predict decision parameters based on past data |

| | | | |
|---|---|---|---|
| Real-time processing | Access status | 14 | Access the status of cyber-physical systems in real-time |
| | Access description | 2 | Access the description of cyber-physical systems in real-time |
| | Build networks | 4 | Build cyber-physical system networks in real-time |
| | Control production | 9 | Control operative production in real-time |

**Table 8 Processing requirements for Industry 4.0** [56]

### 2.4.3 Event management and processing

A fundamental aspect of cyber-physical systems is that of industrial measuring and monitoring, where real-time operating measurements are continuously evaluated using logic residing in the cyber world, which can trigger events that inform other systems and components of state changes or actions (e.g. fault detected) [25]. Indeed, such events are the primary source of information for describing and analysing manufacturing processes [26]. The logic used to identify and trigger events can be derived from existing control logic, engineering first principles, or new knowledge discovery. Although some events may comprise simple rule-based conditions, cyber-physical systems can support comprehensive and wide-reaching event management, using scalable and robust compute resources (e.g. cloud) to facilitate real-time advanced computation (e.g. equipment prognostics) and factory-wide monitoring (e.g. comparative analysis).

Distributed and decentralised event notification for cyber-physical systems may be realised using (1) request/response patterns, or (2) publish/subscribe patterns. In the case of request/response, clients (e.g. software agents) in the factory transmit measurements to a server (i.e. request), and are notified if these measurements triggered an event (i.e. response). To achieve continuous monitoring, this pattern requires clients to periodically poll servers (e.g. 60 seconds) for event notifications. This differs from publish/subscribe, where clients can subscribe to particular events, and receive push notifications when these events occur (i.e. without continuously polling). Given less round-trips and polling behaviours, publish/subscribe patterns can reduce bandwidth, CPU cycles and power consumption [25].

### 2.4.4 Real-time performance

The notion of real-time performance is central part to Industry 4.0 and cyber-physical systems. Essentially, real-time systems are those which are engineered to operate within particular timing constraints. These systems differ from traditional software systems given their deterministic execution, which is a prerequisite for guaranteeing execution response times (i.e. worst case execution time). Although traditional embedded control systems (i.e. automation networks) may exhibit deterministic execution, emerging distributed embedded systems (e.g. cyber-physical systems) possess additional communication latency and computation overhead (e.g. model execution), while delivering the same quality-of-service (QoS) [55]. This may involve real-time measurements from the physical world (i.e. factory) being propagated to the cyber world (i.e. cloud) for analysis, with results being used to improve decision-making or adjust operations [26]. However, such scenarios can increase in complexity where heterogeneous systems and distributed parallel processing must be supported [25].

A system's real-time operating performance may be classified as (1) hard real-time, (2) soft real-time or (3) near real-time. Of these classifications, hard real-time systems are those whose operation is incorrect (i.e. failed) where execution exceeds the timing constraint in a single instance. Typically, these are mission-critical systems (e.g. braking system in a vehicle), whereby execution outside the expected time-window serves no logical purpose (e.g. brakes engaging after crash). Greater operating leniency is afforded to soft real-time systems, whose operation may still be regarded as correct where execution occasionally exceeds the timing constraint (e.g. temporary loss of audio during video conferencing). However, continually violating timing constraints can degrade system performance, and eventually result in a complete system failure, where no further value can be derived from the system's execution. Loosening timing constraints further, near real-time systems are those whose operation are not bound to particular timing constraints, but endeavour to execute without delay. Although the type of real-time system being developed may be largely dependent on the application, given the hardware, software and expertise needed to implement tightly control deterministic systems, budgetary and resourcing constraints may also be contributory factors.

Figure 9 illustrates typical operating parameters for real-time execution. Of these parameters, worst-case execution time (WCET) indicates whether particular systems can perform to the level expected of particular scenarios (e.g. process control and automation). In contrast, best-case execution time (BCET) indicates the system's maximum performance threshold. Given maximum performance implies optimal operating conditions (i.e. low-demand placed on system), reliably maintaining this state is unrealistic for many real-world applications. However, best-case execution can provide insights on hardware configurations and code quality, insofar as optimal combinations can be identified when performance increases. Apart from hardware and software assets, other factors that may impact performance include program inputs (e.g. size of data) and execution context (e.g. cache and processor), with empirical testing of these variables being a common approach to real-time performance analysis.

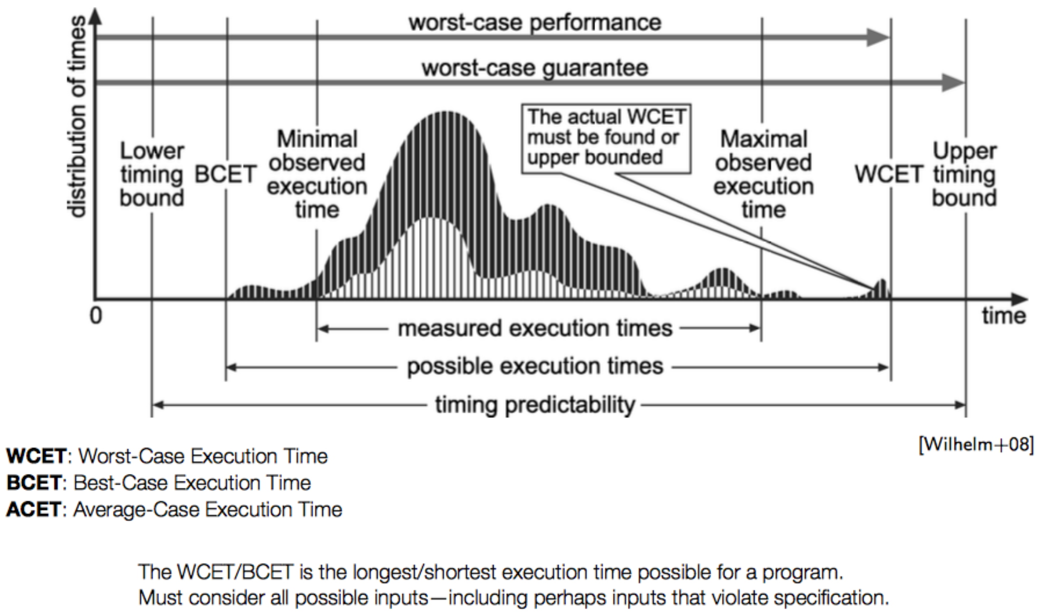



**Figure 9 Illustration of worst and base case execution** [55]

### 2.4.5 Multidisciplinary engagement

Given the engineering and technology convergences comprising industrial cyber-physical systems, the training, development and management of multidisciplinary operational teams is significantly important [23]. These teams inform the development of real-time distributed automation networks, combing expertise from information technology, electronics, engineering and mechanical systems, with in-depth knowledge of facilities, to prescribe interventions and applications [23]. Generally, technology-oriented teams excel when working with data (e.g. integration), but do not possess the domain expertise for meaningful interpretation [57]. In contrast, engineering-oriented teams possess the domain expertise to identify operation data-driven insights, but do not possess the technical skills to implement high quality, robust and scalable technology platforms and systems.

## 2.5 A systematic review and analysis of industrial cyber-physical system research

This section presents findings from a broad systematic review of industrial cyber-physical systems, extracting prominent themes across control, software and engineering disciplines, while highlighting prominent perspectives, technologies, applications and methods. A systematic review methodology was chosen due to the contemporary nature of the field, and the significant deviations between research perspectives and technical approaches. The four dimensions of the review are described below, which were designed to answer questions to inform implementation during this research;

1. **Design (section 2.5.2):** what perspectives are used to form the theoretical basis for industrial cyber-physical system implementation?
2. **Applications (section 2.5.3):** which industrial applications and factory operations demonstrate applications of industrial cyber-physical systems?
3. **Implementation (section 2.5.4):** what technology paradigms, technologies and formats are used to support industrial cyber-physical system implementation?
4. **Standards (section 2.5.5):** which technology and industrial standards are relevant to industrial cyber-physical systems?

### 2.5.1 Methodology

#### Review protocol

At the beginning of the process, several search terms were evaluated to determine the relevancy of retrieved publications. Once a primary search term was chosen, digital sources were interrogated to identify candidate publications, which were manually screened using inclusion and exclusion criteria. This criteria filtering was iterated several times before finalising publications for review. Given the completion of publication filtering, an electronic spreadsheet was created to record publication metadata (e.g. title, authors, year etc.), and dimension classifications (e.g. programming language, system modelling etc.). As publications were reviewed, researchers used the spreadsheet to classify contributions and properties of the research. After synthesising the entire publication repository, the collected spreadsheet data was visualised to highlight prominent approaches and methods.

Initial search evaluations suggested the most relevant publications were found using broad search strategies. Therefore, the search terminology simply comprised synonyms of 'Industrial Internet of Things', 'Industrial Cyber Physical Systems' and 'Manufacturing'. This search strategy was applied to prominent sources, databases and indexes, including (1) Science Direct, (2) Mendeley, (3) ACM Digital Library, (4) Engineering Village, (5) IEEE Xplore, (6) Scopus, (7) Web of Science, (8) ResearchGate, and (9) Google Scholar. After publication filtering and refinement, 93 peer-reviewed conference and journal publications remained. Figure 10 illustrates the geographical origins of these publications, with Germany and China being the most prominent contributors.

### THREATS TO VALIDITY

Although systematic reviews provide rigorous and robust methods for synthesising the literature, potential threats pertaining to the validity of the research should be considered. The primary threats considered for this review are summarised below;

- **Search strategy:** given publications were identified and filtered using a formal search strategy, biases could influence which publications were included. This threat was managed in a couple of ways. First, the search strategy was derived from group discussions to dilute biases. Second, two or more researchers filtered each dimension of the literature, with inclusion and exclusion criteria discussed and debated over much iteration.
- **Search sources:** although the digital repositories used to identify publications are prominent sources of academic literature, other relevant sources of industrial cyber-physical system research may exist. This threat was managed using different types of digital sources to provide a cross-section of research publications (e.g. indexes, databases, crawlers, and social repositories).
- **Classification accuracy:** due to the diverse and discipline-specific terminology used across industrial cyber-physical system research, some interpretation and inference is needed to synthesise the literature. Given the need for such decision-making, potential misclassifications could impact results. Although this threat cannot be completely removed, the review protocol ensured two or more researchers classified each publication, and research efforts focused more on relevance to smart manufacturing, rather than prominence alone. This meant minor misclassifications should not impact the narrative of the review.

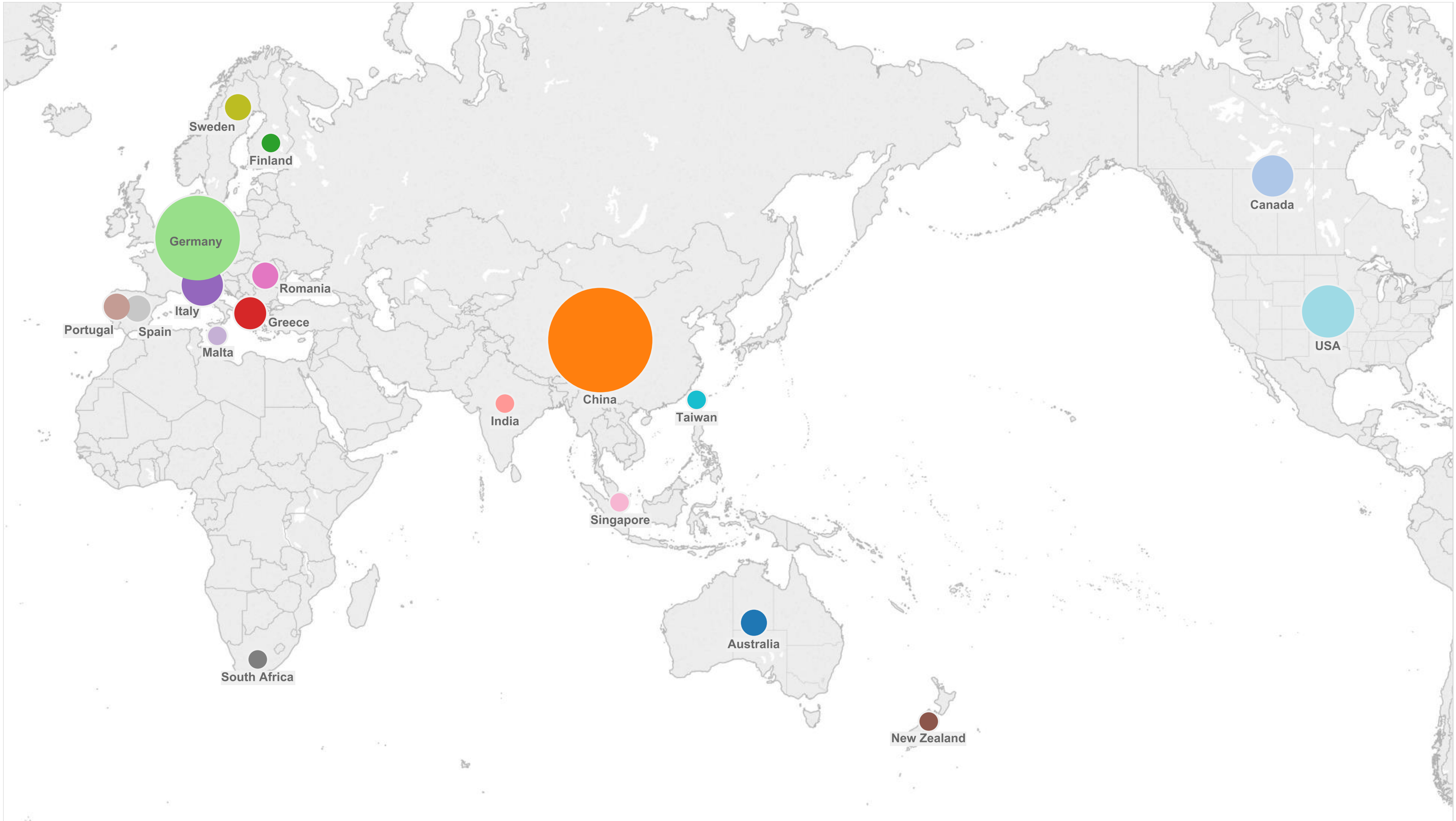



**Figure 10 Geographical distributions of industrial cyber-physical publications**

## 2.5.2 Design of industrial cyber-physical systems

### 2.5.2.1 MULTIDISCIPLINARY DESIGN PERSPECTIVES

The concept of cyber-physical systems is not new, with embedded systems from the 1970's demonstrating interactions between physical and compute components (e.g. vehicle braking systems) [58]. However, emerging industrial cyber-physical systems differ greatly in their scope, complexity, and distributed networking capabilities, while broader engineering disciplines (e.g. process engineering, computer science etc.) are needed to realise implementation. These disciplines provide knowledge and perspectives that address design challenges relevant to industrial cyber-physical systems.

Figure 11 illustrates the intertwined and overlapping nature of design perspectives relating to cyber-physical systems. In this example, design perspectives are depicted as the intersection between design concerns (i.e. x-axis), and parts comprising cyber-physical systems (i.e. y-axis). These perspectives include control robustness, control performance and software design. Although it appears some perspectives share common design concerns, each concern may differ in context. For example, both software and control perspectives are concerned with 'computing platform performance'. However, this concern may lead software engineers to optimise analytics execution in the cloud, while control engineers may aim to improve aspects of control logic on the automation network.

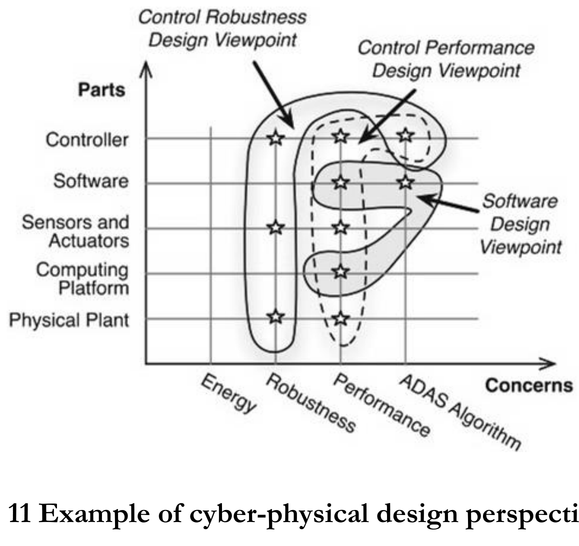



**Figure 11 Example of cyber-physical design perspectives** [58]

Figure 12 illustrates the distribution of multidisciplinary design perspectives extracted from the literature, which include (1) software, (2) control, (3) design, (4) social, and (5) robotics. Generally, software design perspectives focus on information systems and technology infrastructures for data transmission, processing and analysis, which provide the platform for cyber-physical interactions, while control design perspectives focus on distributed, robust, resilient and high-performance industrial control strategies, which ensure secure and reliable control processes. Less prominent perspectives found in the literature include those of design, social and robotics, encompassing methodologies facilitating the design process, protocols supporting human computer interaction, and integrations enabling smart robotics, computation and analytics.

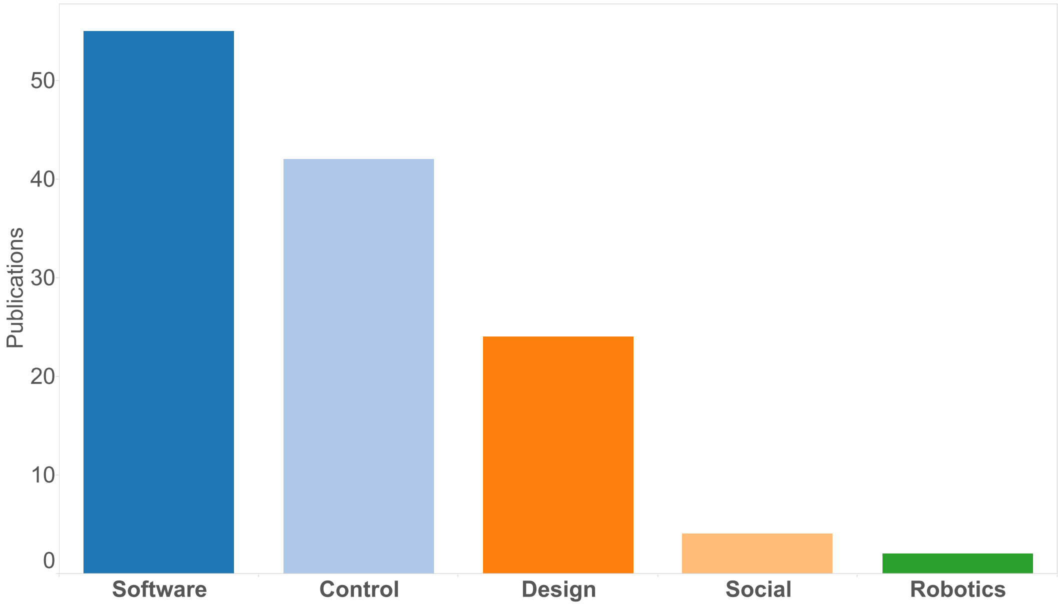



**Figure 12 Distribution of multidisciplinary design perspectives**

### 2.5.2.2 Modelling approaches

Given the existence of multidisciplinary design perspectives, different modelling approaches may be employed to align with these perspectives. Therefore, the design of industrial cyber-physical systems may comprise multiple modelling approaches. These approaches create abstract representations of the proposed system, which may be evaluated, interrogated and analysed in lieu of technical implementation. Modelling before implementation enables the early identification of technical issues, vulnerabilities and limitations, which reduce the costs associated with technical pivoting downstream, while also providing stakeholders and development partners with a common understanding and vocabulary.

Figure 13 illustrates the distribution of modelling approaches extracted from the literature. These approaches were derived during the review process to synthesise and consolidate disparate low-level approaches (i.e. terminology and technical depth varied significantly). The modelling approaches identified include (1) conceptual, (2) software and (3) mathematical modelling, with conceptual modelling frequently used to formulate an understanding of the domain (e.g. cyber-physical system components) for technical and non-technical stakeholders. Although conceptual models are particularly useful during early stages of projects, they do not embody the formal and standard notation needed for technical implementation. Such details are more precisely described using software and mathematical modelling, comprising standard methods for modelling system components and simulations. Table 9 compares the primary strengths and weaknesses of the aforementioned modelling approaches.

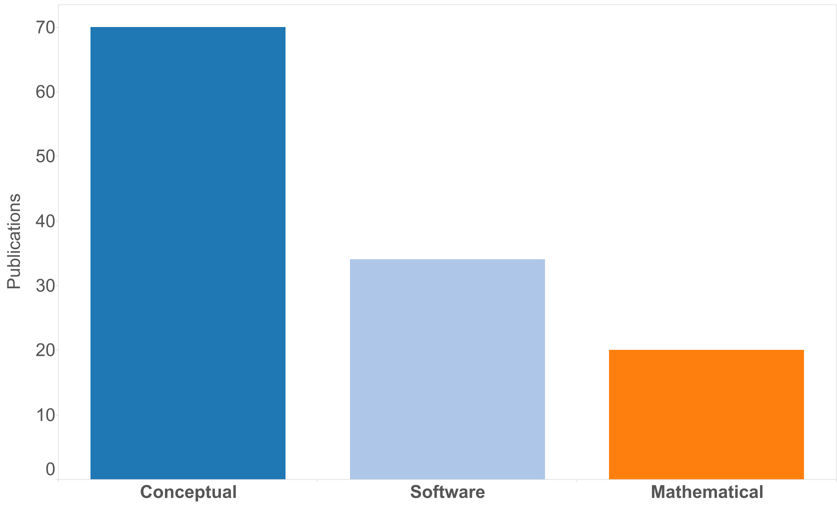



**Figure 13 Distribution of modelling approaches**

| Models | Strengths | Weaknesses | References |
|---|---|---|---|
| Conceptual | Clear representations of where the system or technology functions within the domain. | Lack of technical prescription and formalisms. | [42], [59]–[127] |
| Software | Formal and consistent way of modelling requirements and technical implementation. | Formalisms may not possess vocabulary or representations for the domain being modelled, while performance execution may not be derived using these models alone. | [71], [74], [80], [81], [91], [92], [94], [96]–[98], [101], [102], [117], [126], [128]–[135] |
| Mathematical | Formal and consist way of validating execution performance in lieu of implementation. | Removed from domain modelling, requirements analysis and technical details needed to implement broader real-world systems. | [67], [72], [73], [79], [81], [86], [98], [104], [121], [136]–[144][67], [72], [73], [104], [136]–[140], [145], [146] |

**Table 9 References for modelling approaches**

### 2.5.2.3 Modelling languages

A modelling approach depends on suitable languages to describe and formalise technical specifications. The modelling languages supporting industrial cyber-physical systems design can be broadly classified as software and mathematical. Of these classifications, languages with software origins target component and system-level design (e.g. information system architecture), while mathematical modelling languages focus more on algorithm-level design (e.g. control logic).

Figure 14 illustrates the distribution of modelling languages extracted from the literature, and Table 10 provides high-level descriptions and references supporting the data visualisation. Although the reported use of these languages for industrial cyber-physical system design is modest, the Unified Modelling Language (UML), Petri Nets, Systems Modelling Language (SysML), and Service Oriented Modelling Language (SOA ML) are the most prominent. Of the less prominent modelling languages, Domain Specific Modelling Language (DSML) represents an alternative to general-purpose (e.g. UML) modelling languages, which encourages the creation of unique modelling notations for particular business domains. Given the interdisciplinary nature of industrial cyber-physical systems, domain-specific modelling languages that capture the semantics and vocabulary of the field may prove useful to formalise design and development processes, when compared to more general-purpose languages.

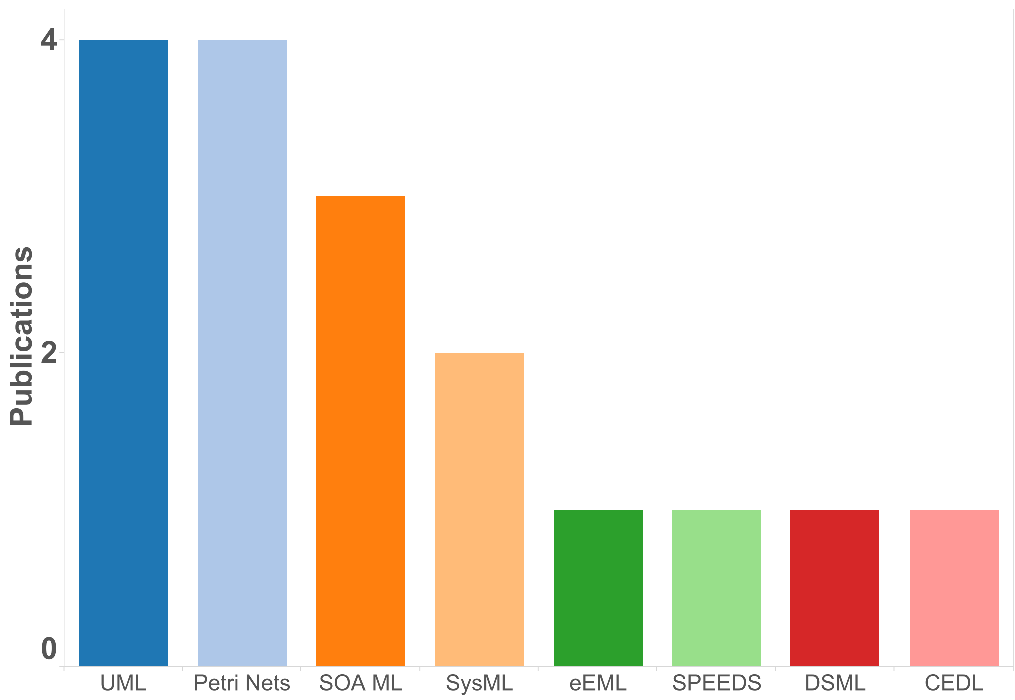



**Figure 14 Distribution of modelling languages**

| Language | Description | References |
|---|---|---|
| Domain Specific Modelling Language (DSML) | High-level modelling methodology for specific domains | [134] |
| Extended Enterprise Modelling Language (eEML) | Modelling language for enterprise systems | [109] |
| Petri Net | Mathematical modelling language for describing distributed systems | [72], [77], [121], [137] |
| SOA ML | Modelling language for designing service-oriented architectures | [42], [77], [117] |
| SPEEDS | Design approach demonstrated for real-time automation and control. | [145] |
| Systems Modelling Language (SysML) | Modelling language for analysis, design and verification of complex systems | [132], [145] |
| Unified Modelling Language (UML) | General-purpose modelling language for visualising system design | [80], [94], [108], [133] |

**Table 10 References for modelling languages**

In terms of borrowing from existing modelling approaches, the Unified Modelling Language (UML) and System Modelling Language (SysML) represent mature and standard languages for modelling software components and systems. Both languages are based on symbolic and visual notation that convey system design. UML's primary notation consists of structural (e.g. class), behavioural (e.g. use case), and interactive (e.g. sequence) diagrams. These diagrams aim to remove ambiguity and unknowns from the software design process, by enabling developers to consider system design from many different perspectives. Although UML diagrams are most commonly used to support object-oriented software development, additional notation is required for system engineering (e.g. system-to-system communication). Thus, SysML was developed to support system engineering efforts by extending UML's symbolic notation (e.g. activity diagram), and introducing additional requirements and parametric diagrams for modelling constraints and timing scenarios (e.g. automation and control).

Apart from component-level software and system modelling, cyber-physical systems may also be concerned with low-level algorithm design, process validation, distributed operation and concurrent execution, which can be more easily modelled using mathematical modelling languages. The primary mathematical modelling language observed from the literature was that of PetriNets. These models are used to describe event-driven, concurrent and distributed systems, using graphs of 'places' and 'transitions', with 'places' representing the current state (e.g. valve closed), and 'transitions' representing events (e.g. open valve) that change state (e.g. valve open).

#### 2.5.2.4 Reference architectures and models

While modelling languages can provide the methods and notation for designing industrial cyber-physical systems from the bottom-up, reference architectures and models provide standard, systematic and prescribed approaches (e.g. technical templates) for connecting industrial systems and technologies. Based on observations from the literature, both mature and emerging reference architectures and models have been proposed to support Industry 4.0 operations. These architectures and models can standardise integration, terminology, and technologies, which can greatly simplify time-consuming data acquisition, processing and analytics activities.

Figure 15 illustrates reference architectures and models extracted from the literature, while Table 11 provides a high-level description and references supporting the data visualisation. Of those identified, ISA-95 and RAMI 4.0 are the most prominent industrial architectures and models. The ISA-95 is a mature and widely adopted five-part industrial automation standard published by the International Society of Automation (ISA) that captures many of the technical layers relevant to cyber-physical systems (e.g. sensing, control etc.), while Reference Architecture Model Industry 4.0 (RAMI) is a more contemporary standard that was specifically designed to support the multifaceted nature of Industry 4.0 operations. Other reference architectures and models of industrial origin include the Manufacturing Enterprise Solutions Association (MESA) for managing business processes, Product-Resource-Order-Staff-Architecture (PROSA) for architecting holonic manufacturing systems, and Automotive Open System Architecture (AUTOSAR) for industrial automotive control. Notably, the concept of holonic systems promoted by PROSA are theoretically aligned with the Industry 4.0 ideology of self-organising, modular, embedded and decentralised decision-making. However, the less prominent architectures and models identified do not appear to embody sufficient breadth to support Industry 4.0 operations.

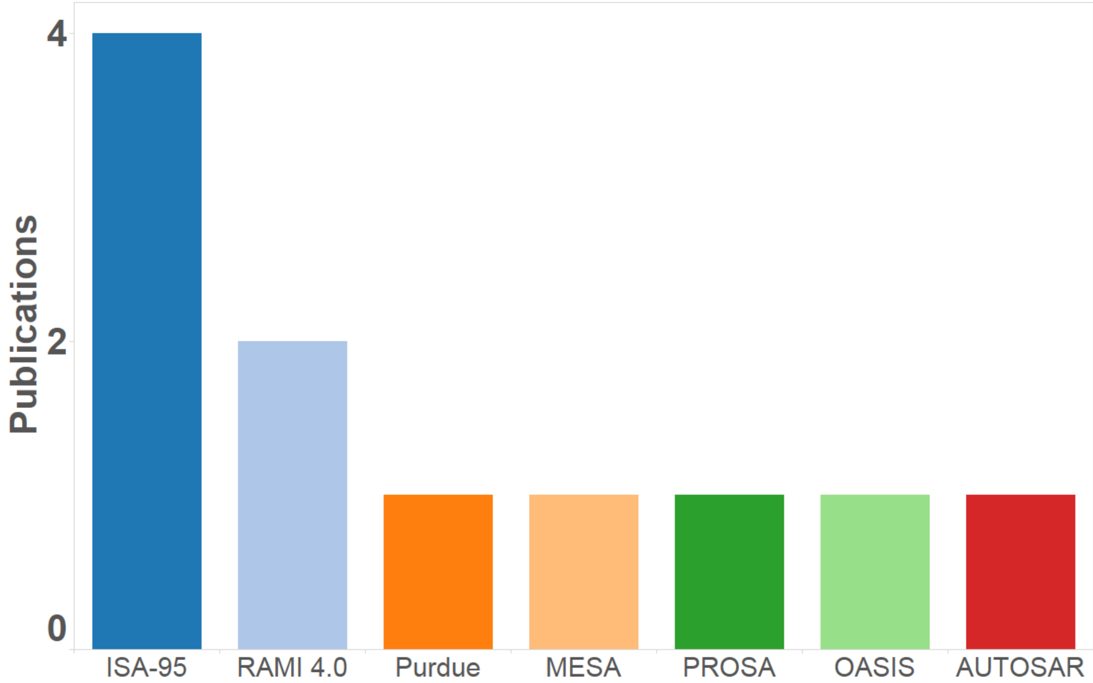



**Figure 15 Distribution of architectures and models**

| Architecture | Description | Reference |
|---|---|---|
| Automotive Open System Architecture (AUTOSAR) | Open and standard software architectures for automotive electronic control units | [145] |
| ISA-95 | Standard for automating interfaces between enterprise and control systems | [71], [77], [83], [124] |
| Manufacturing Enterprise Solutions Association (MESA) | Reference model supporting business processes in the factory | [71] |
| OASIS Reference Model for Service Oriented Architecture (SOA-RM) | Reference model for managing and unifying multiple service-oriented applications | [117] |
| Product-Resource-Order-Staff-Architecture (PROSA) | Reference model for supporting holonic manufacturing systems | [117] |
| Purdue Enterprise Reference Architecture (PERA) | Reference model for developing enterprise architectures | [71] |
| Reference Architecture Model Industry 4.0 (RAMI) | Model describing Industry 4.0 compliant production equipment and processes | [114], [120] |

**Table 11 References for architectures and models**

ISA-95's hierarchical model (Figure 16) depicts factory-to-enterprise integration as five distinct technical levels, which are process (Level 0), instrument (Level 1), monitoring and control (Level 2), operations management (Level 3), and business planning (Level 4). The hierarchical ISA-95 model was initially designed to support centralised automation and control technology (i.e. Industry 3.0), but may also be adapted to support Industry 4.0 operations. In addition to ISA-95's reference model, other parts of the standard address details regarding the data flows, objects and relationships that may exist between each layer of the model.

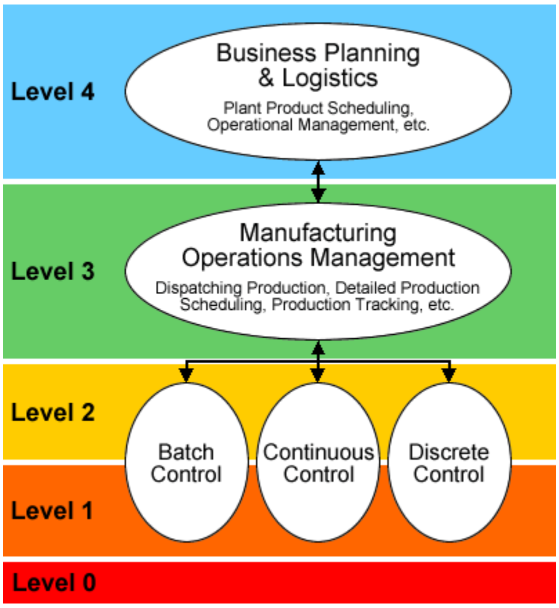



**Figure 16 ISA-95 reference architecture**

The ISA-95 model demonstrates strong vertical integration (i.e. factory-to-enterprise integration), but emerging industrial cyber-physical systems for Industry 4.0 depend on both vertical and horizontal integration. Thus, multi-directional integration scenarios are an important aspect of RAMI 4.0's three-dimensional architecture (Figure 17), comprising hierarchical levels (e.g. field device, controller etc.), product lifecycle value stream (e.g. production, maintenance etc.), and architecture layers (e.g. asset, integration etc.). Of these dimensions, the product lifecycle (IEC 62890) and hierarchical levels (IEC 62264) are based on IEC standards, with the latter dimensions sharing the same standard as ISA-95's reference model.

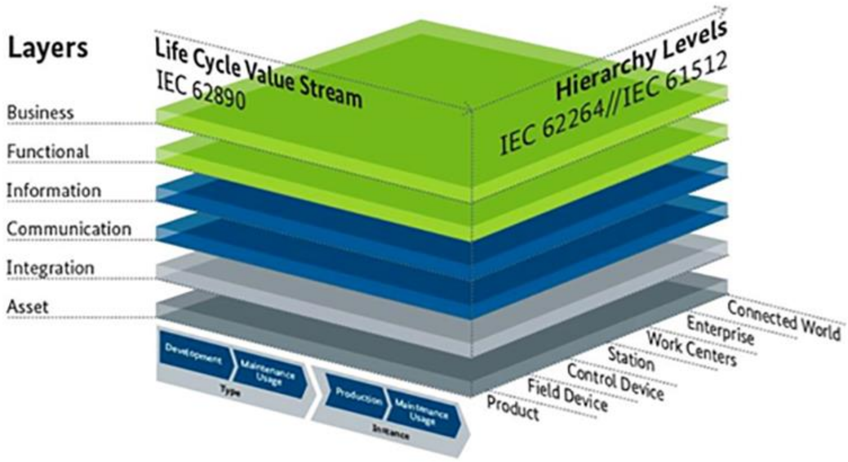



**Figure 17 RAMI 4.0 reference model**

#### 2.5.2.5 Design observations relevant to thesis

Consistent *design perspectives* and requirements for industrial cyber-physical systems are not evident in the literature. Although there may be many reasons for poor consensus within the literature, some of the contributing factors proposed by this research include the (a) contemporary nature of research relating to industrial cyber-physical systems, (b) inconsistent understanding regarding how cyber-physical technology relates to traditional fields of engineering, computer science, and information systems, and (c) poor integration and acceptance of multidisciplinary concerns and perspectives. Hence, much of the current literature hold discipline-specific perspectives, with control and software perspectives being particularly prominent. The control perspectives inform concurrent control and guaranteed process execution (e.g. real-time performance), while software perspectives enable the extension of traditional control systems to embed additional manufacturing intelligence, scenario simulation and advanced analytics (e.g. machine learning for self-optimisation).

The most prominent *modelling approaches* found in the literature employed conceptual, software and mathematical models. Theoretical and conceptual models convey high-level details and schematics for industrial cyber-physical systems. These models are valuable when developing a common understanding of the domain's primary components (e.g. programmable controllers), communicating methodological approaches (e.g. workflows), and illustrating engineering applications (e.g. equipment maintenance). However, conceptual models do not typically follow a formal convention, or provide the technical details to support the transition from concept to implementation. Technical transitions depend on software models to visualise the components and interactions that occur at different levels of abstraction, from high-level analysis, to low-level implementation. Unlike conceptual models, software models typically adhere to formalisms that encourage consistency, continuity and collaboration. These formal software models may be used to validate software designs against functional requirements, but reported demonstrations encountered during the review process did not indicate how these models could be used to measure execution performance (e.g. worst case execution time), which is an important part of embedded systems (e.g. control networks) that depend on code execution within a set time-window, or where systems are comprised of distributed components that execute concurrently. This type of performance analysis may be realised using mathematical models to simulate performance and verify system, control and algorithm design in lieu of implementation. Therefore, the benefits and concerns associated with these modelling approaches (i.e. concept, software and performance) should be considered an integral part of a *unifying design methodology* for industrial cyber-physical systems.

The *modelling techniques and tools* used to design industrial cyber-physical systems naturally mirror the disciplinary design perspectives of the researcher, with existing modelling languages (e.g. UML) and tools used to guide design efforts. Although using existing modelling techniques and tools makes sense to pioneer initial efforts, many of these tools were not created to address the requirements, concepts and vocabularies relevant to industrial cyber-physical systems, and therefore one might consider how to integrate and connect these modelling tools to form a coherent and *unified design methodology* for industrial cyber-physical systems.

### 2.5.3 Application of industrial cyber-physical systems

#### 2.5.3.1 Targeted operations and deployments

The deployment of cyber-physical systems within industrial operations can be broadly classified as those relating to (a) specific engineering applications (e.g. process optimisation, fault detection etc.), or (b) enabling engineering technology (e.g. frameworks to execute predictive engineering models). While engineering applications leverage advanced computation, simulation and analytics to inform operational decision-making, enabling engineering technologies provide frameworks, infrastructures and processes that ensure robust and reliable operation. In essence, enabling engineering technologies are utilised by engineering applications to access, process and report operational information. Although there are many possible engineering applications applicable to Industry 4.0, they each share the objective of contributing some intelligence to realise self-optimising and self-configuring operations.

Figure 18 illustrates the current distribution of targeted operations extracted from the literature (i.e. areas where cyber-physical systems have been deployed), while Table 12 provides a high-level description and references supporting the data visualisation. Of those identified, research targeting contributions towards advancing platform enabling technology (e.g. architectures and methods) was most prominent, which was followed closely by research specifically applying cyber-physical technology to process, control and automation. However, given enabling technology transcends any one particular operational area, and logically precedes the development of more specific engineering applications, the prominence of enabling technology may be attributed to the early stage of the field. The benefits of less prominent cyber-physical engineering applications relating to maintenance, planning and energy are largely driven by Industry 4.0 objectives relating to operating reliability (e.g. 100% uptime) and energy efficiency.

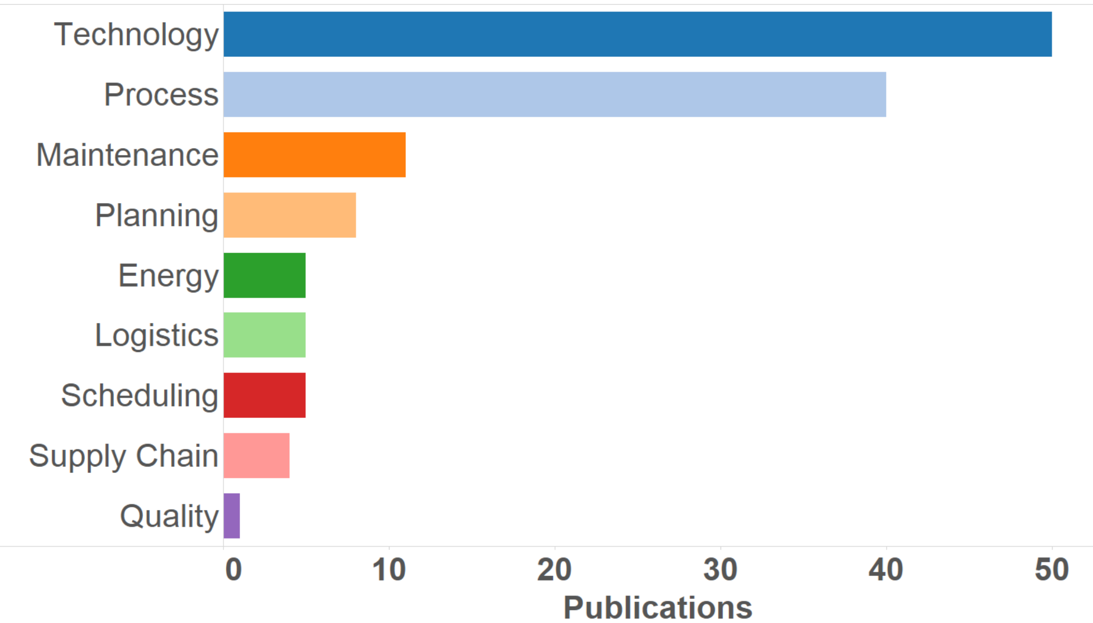



**Figure 18 Distribution of targeted operations and deployments**

| Target | References |
|---|---|
| Technology | [42], [59], [61], [63]–[66], [68], [70], [72]–[74], [76]–[80], [84], [85], [87], [88], [90], [92], [94]–[100], [106], [109], [117]–[120], [124], [126], [127], [129]–[135], [145], [147]–[149] |
| Process | [64], [67], [71], [75], [77], [79], [81], [86], [88], [90], [91], [93], [97], [98], [101], [102], [104], [107], [109], [111], [113], [115], [117], [121], [123], |

| | |
|---|---|
| | [128], [129], [131], [133]–[136], [138]–[143], [146], [150] |
| Maintenance | [24], [60], [62], [68], [76], [78], [89], [105], [110], [122], [125] |
| Planning | [24], [86], [90], [112], [123], [141], [142], [146] |
| Energy | [82], [110], [116], [118], [144] |
| Logistics | [83], [86], [90], [103], [114] |
| Scheduling | [86], [90], [107], [123], [137] |
| Supply Chain | [86], [99], [108], [120] |
| Quality | [74] |

**Table 12 References for targeted operations and deployments**

#### 2.5.3.2 Engineering and computing applications

An industrial cyber-physical system can be programmed using a wide-range of computing, mathematical, and statistical methods, and applied to any number of engineering applications and scenarios. Although capturing such diversity can be difficult, themes and classifications (Figure 19) derived from the literature include applications for (1) *information technology* supporting fundamental networking, infrastructure, security and management, (2) *operation technology* enabling the automation, control and monitoring of industrial processes, and (3) *Industry 4.0 technology* (e.g. cyber-physical systems) capable of delivering predictive and self-regulating operation. These classifications may be considered somewhat hierarchical, with Industry 4.0 technologies residing at the top of the stack, extending principles from both information and operational technology (i.e. technology convergence between information and operation technology). In a similar manner, operation technology extended many principles and components (e.g. computer networks, database systems etc.) of information technology to realise industrial automation and control, while information technology provides the foundational layer for these engineering informatics domains.

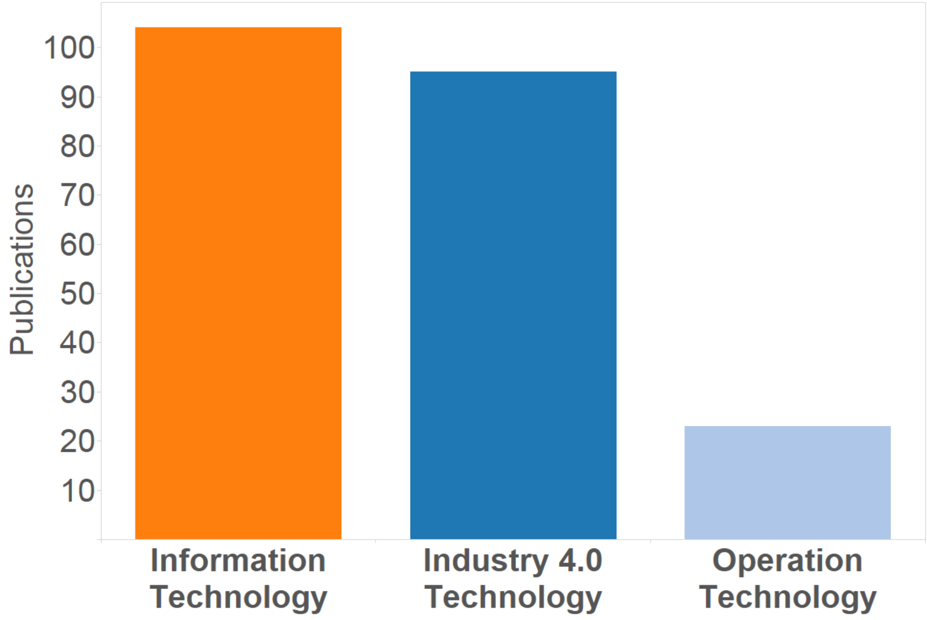



**Figure 19 Classification of engineering and computing applications**

Figure 20 illustrates the distribution of engineering and computing applications extracted from the literature, while Table 13 provides descriptions and references supporting the data visualisation. Of the engineering and computing applications identified, those focusing on remote monitoring, remote control and self-optimisation are most prominent. In the case of remote monitoring and control, such results should be expected given the scope of these applications are rather broad and generic, while arguably being one of the most common applications of technology in modern factories. A more insightful trend is the existence of self-oriented applications (i.e. all derivations of self-*), which demonstrates the field of industrial cyber-physical systems shares a common understanding and vocabulary with recognised Industry 4.0 objectives. In addition, the presence of applications focusing on smart connections emphasises the need for connectivity between systems, devices and repositories (e.g. internet-of-things, legacy interfaces) to facilitate interconnectivity for Industry 4.0.

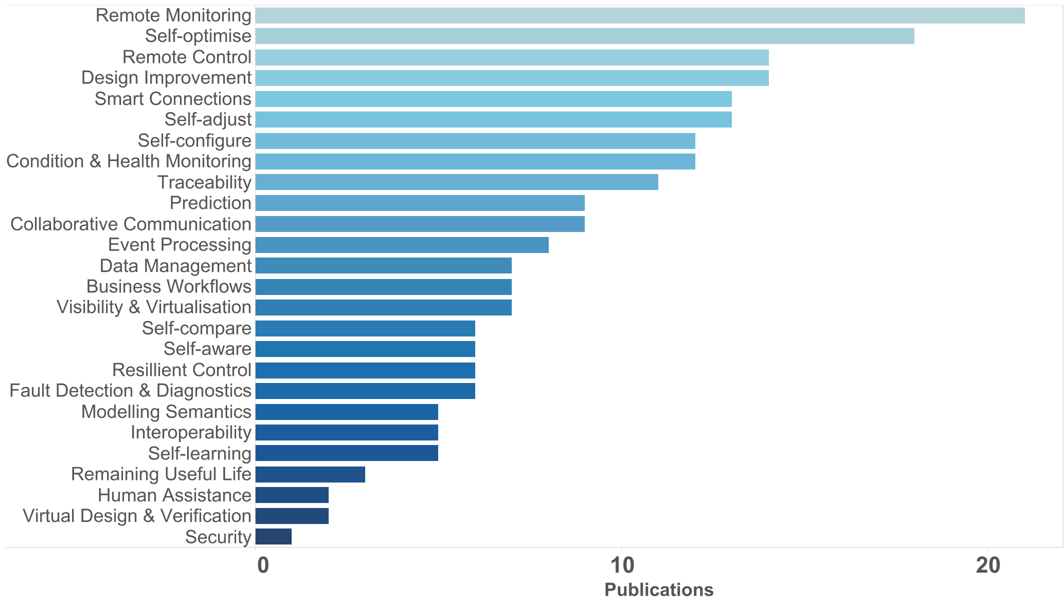



**Figure 20 Distribution of engineering and computing applications**

#### 2.5.3.3 Application insights relevant to thesis

The literature indicates that the general application and purpose of industrial cyber-physical systems are consistent with Industry 4.0 goals and objectives, such as self-configuration, self-optimisation, self-awareness etc. These high-level classifications overlap with the type of analytics models used to fulfil engineering operations (e.g. self-configuration may suggest the use of prescriptive or recommendation models), while they also differentiate industrial cyber-physical systems from similar cyber-physical technologies servicing other domains. Another differentiating characteristic relates to the presence of data streams with differing latencies, where some engineering scenarios (e.g. control) depend on real-time data streams, while others (e.g. modelling) employ batch data streams. Despite the potential importance of these data streams to Industry 4.0, industrial analytics, and industrial cyber-physical systems, the literature largely overlooks the relationship and potential relevance of these data streams to industrial cyber-physical systems. Given poor data governance and management can increase the cost, and reduce the return-on-investment of industrial analytics initiatives, applications of industrial cyber-physical systems should aim to demonstrate a strong understanding of data flows and lifecycles throughout the factory.

Although the number of reported real-world industrial cyber-physical systems are modest, there are two prominent themes relating to the application of cyber-physical system technology to industrial operations. These themes are summarised below;

- **Engineering Informatics** applications that focus on delivering information and operation technology convergences that enable cyber-physical interactions between traditional manufacturing environments, and more contemporary technologies. In essence, such research efforts aim to deliver the technical artefacts (e.g. system architecture) needed to extend current technologies, and incorporate advanced analytics, optimisation and simulation.

- **Engineering Encoded** applications focus on encoding engineering first principles to monitor, optimise and control specific industrial operations (e.g. equipment maintenance, energy optimisation etc.). These encoded applications are almost entirely concerned with the results derived from execution, rather than making meaningful contributions to the underlying computing or technology supporting the application.

Given the complimentary contributions and insights derived from both classifications of engineering applications, *engineering encoded* and *engineering informatics* applications must be embraced to support the development of well-balanced industrial cyber-physical systems that can be applied to real-world Industry 4.0 factory operations.

| Classification | Application | References |
|---|---|---|
| Industry 4.0 | Collaborative Communication | [67], [77], [79], [80], [82], [84], [98], [100], [103] |
| Industry 4.0 | Human Assistance | [76], [117] |
| Industry 4.0 | Resilient Control | [62], [100], [114], [126], [129], [135] |
| Industry 4.0 | Self-adjust | [51], [59], [65], [77], [81], [87], [88], [126], [138], [139], [143], [144], [151] |
| Industry 4.0 | Self-aware | [51], [65], [67], [77], [110], [113] |
| Industry 4.0 | Self-compare | [51], [62], [69], [113], [150], [151] |
| Industry 4.0 | Self-configure | [51], [67], [68], [77], [81], [87], [111], [113], [124], [127], [150], [151] |
| Industry 4.0 | Self-learning | [67], [87], [114], [118], [132] |
| Industry 4.0 | Self-optimise | [42], [62], [67], [72], [77], [88], [92], [98], [102], [123], [125], [137], [140]–[143], [146], [151] |
| Industry 4.0 | Traceability | [83], [88], [91]–[93], [102], [108], [121], [134], [141], [142] |
| Industry 4.0 | Visibility & Virtualisation | [90], [91], [93], [94], [102], [121], [123] |
| Information Technology | Business Workflows | [72], [83], [85], [90], [92], [102], [110] |
| Information Technology | Data Management | [72], [74], [86], [93], [122], [128], [130] |
| Information Technology | Design Improvement | [64], [79], [115], [117], [118], [122], [124], [127], [129], [131]–[133], [135], [145] |
| Information Technology | Interoperability | [121], [124], [125], [127], [132] |
| Information Technology | Modelling Semantics | [127], [133]–[135], [145] |
| Information Technology | Prediction | [60], [64], [89], [90], [98], [108], [123], [125], [128] |
| Information Technology | Security | [95] |
| Information Technology | Smart Connections | [42], [51], [62], [65], [66], [80], [86], [89], [117], [119], [120], [126], [130] |
| Operation Technology | Condition & Health Monitoring | [51], [62], [64], [65], [69], [89], [90], [98], [122], [125], [140], [151] |
| Operation Technology | Event Processing | [74], [88], [101], [103], [110], [120], [121], [123] |
| Operation Technology | Fault Detection | [60], [64], [65], [121], [122], [128] |
| Operation Technology | Remaining Useful Life | [51], [64], [69] |
| Operation Technology | Remote Control | [90]–[92], [97], [98], [127], [145] |
| Operation Technology | Remote Monitoring | [61], [67], [68], [73]–[75], [87], [88], [90], [91], [97], [101], [103], [107], [109], [110], [116], [119], [128], [150], [152] |
| Operation Technology | Virtual Design & Verification | [133], [134] |

**Table 13 References for engineering and computing applications**

## 2.5.4 Implementation of industrial cyber-physical systems

### 2.5.4.1 Technology paradigms and methods

The current industrial cyber-physical system ecosystem comprises both mature and contemporary technologies. Figure 21 illustrates technology paradigm classifications extracted from the literature. Of these classifications, Internet (e.g. cloud and web services) and enterprise (e.g. object-oriented programming) are the most prominent. These represent mature and familiar technologies capable of managing distributed computing scenarios using enterprise computing, networking and power capabilities (e.g. personal desktop computer). However, such requirements differ from the restricted power, connectivity and compute capacity synonymous with industrial cyber-physical systems (e.g. sensors and microcontrollers). Although some mobile and emerging technologies may address these requirements, they are less prominent in the literature. Logically, this could be expected given potentially smaller communities, unfamiliar development tools, and limited legacy system support. However, mobile and emerging technologies demonstrate strong alignment with Industry 4.0 objectives (e.g. interoperability, product customisation, decentralisation), industrial cyber-physical systems (e.g. resilient low-power communication), and cross-sector 'smart' paradigms (e.g. smart grid, smart home etc.).

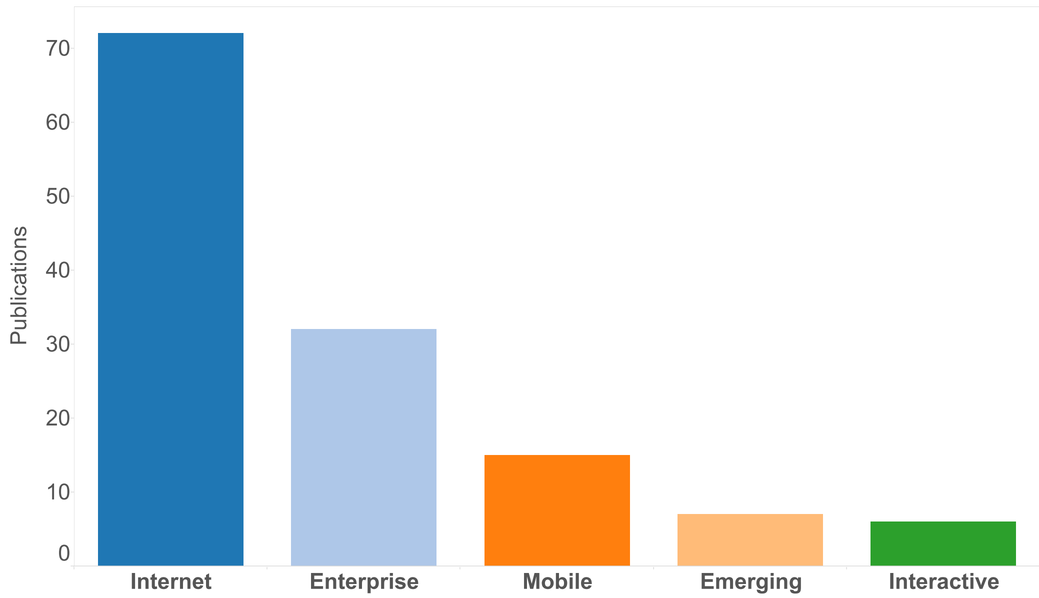



**Figure 21 Classification of technology paradigms and methods**

Figure 22 illustrates the distribution of technology paradigms and methods extracted from the literature, while Table 14 provides descriptions and references supporting the data visualisation. Of the technology paradigms and methods identified, service-oriented computing and cloud computing are most prominent, which may be somewhat attributed to Industry 4.0 design principles encouraging modularity and service-orientation. These technical characteristics can facilitate reusability, maintenance and distributed processing using open interfaces. These interfaces can be used to support interoperability between disparate enterprise systems, and commonly deployed on cloud computing platforms to promote scalability and fault tolerance using dynamic (i.e. elastic) provisioning, where compute resources (e.g. memory, processors etc.) are programmatically initiated on-demand. In essence, the amalgamation of cloud computing with service-oriented architectures provide the fundamental technologies for cloud manufacturing.

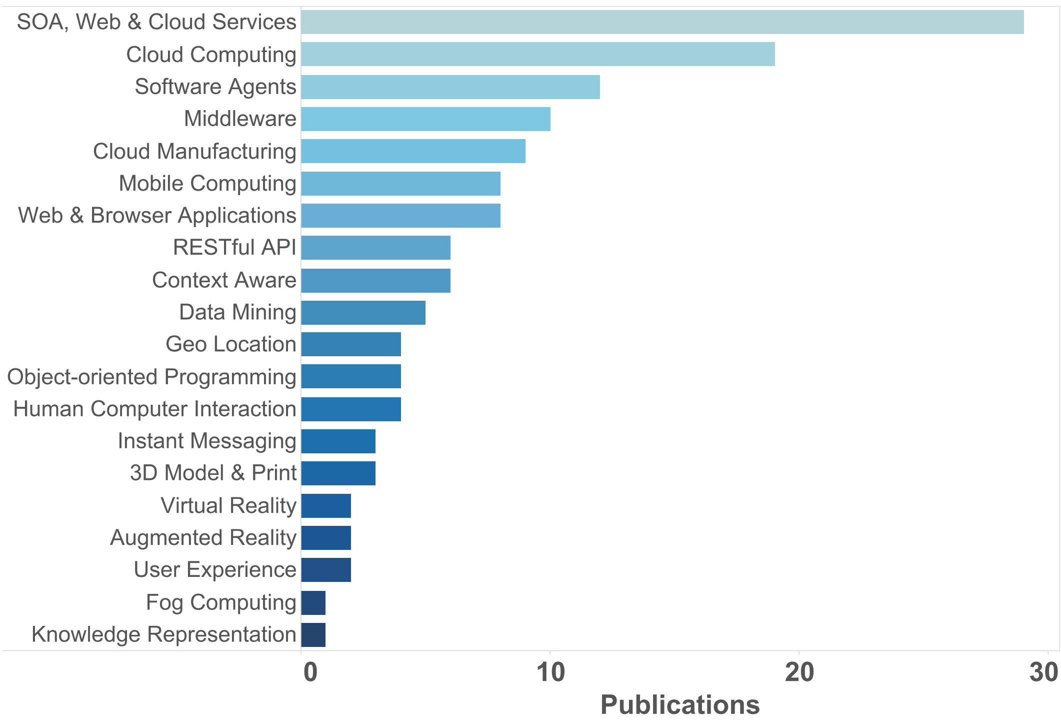



**Figure 22 Distribution of technology paradigms and methods**

Although cloud and service-oriented computing can support distributed engineering scenarios, intelligence and processing (e.g. decision-making) typically remain central (e.g. cloud server), with distributed client software depending on consistent and resilient connections to the primary services. However, given industrial cyber-physical systems comprise networks-of-networks with varying bandwidth, architectures dependent on persistent connections to centralised services are not naturally suited to real-time automation and control scenarios. To address such scenarios, decentralised paradigms supporting distributed and autonomous decision-making may be considered. Both software agents and fog computing exemplify decentralised paradigms, where compute nodes can operate autonomously to deliver intelligence on the outer edge of pervasive networks. In addition to removing dependencies on cloud connectivity, decentralised paradigms may also reduce network traffic, improve scalability and enhance security for industrial cyber-physical systems.

In terms of application user interfaces, web and mobile paradigms are the most prominent. Although web applications operate solely within web browsers, mobile applications can be delivered using web browsers, or native mobile platforms (e.g. iOS, Android etc.), with browser-based delivery particularly convenient for supporting cross-platform scenarios. More contemporary application interfaces enabled by virtual and augmented reality paradigms are gaining more attention, but widely accepted standards for development and deployment are not evident. Therefore, while these contemporary and interactive interfaces offer potential for industrial innovation, their demonstrated application to industrial cyber-physical systems appears limited.

| Classification | Technology | Purpose | References |
|---|---|---|---|
| Interactive | Context Aware | Sense physical environment and adapt behaviour | [80], [81], [94], [130], [132] |
| Interactive | Human Computer Interaction | Optimise human interactions with the cyber world | [75], [76], [90], [117] |
| Interactive | User Experience | Design and implement human centred systems | [76], [92] |
| Emerging | 3D Modelling and Printing | Facilitate additive manufacturing for Industry 4.0 | [75], [117], [118] |
| Emerging | Augmented Reality | Overlay cyber objects in the physical world | [76], [117] |
| Emerging | Cloud Manufacturing | Enable distributed and service-oriented manufacturing | [59], [63], [66], [72], [83], [84], [103], [107], [109] |
| Emerging | Fog Computing | Propagate cloud services to the edge of networks | [108] |
| Emerging | Software Agents | Execute actions on behalf of another entity (i.e. virtual surrogates) | [75], [77], [81], [98], [100], [109], [111], [114], [116], [117], [124], [130] |
| Emerging | Virtual Reality | Model physical objects in a virtual world | [109], [117] |
| Enterprise | Data Mining | Examine data repositories to derive insights | [63], [65], [81], [82], [98] |
| Enterprise | Knowledge Representation | Encode knowledge in a form understood by computers | [71] |
| Enterprise | Middleware | Mediate communication between different systems | [74], [81], [85], [87], [91], [97], [99], [102], [108], [110] |
| Enterprise | Object-oriented Programming | Combine data and functions representing real-world entities | [65], [115], [131], [133] |
| Internet | Cloud Computing | Deliver elastic compute and storage for performance and scalability | [51], [62]–[66], [68]–[70], [73], [74], [77], [82], [83], [96], [103], [111], [116], [120] |
| Internet | RESTful API | Provide lightweight and open methods for remote program execution | [71], [80], [119], [124], [133], [147] |
| Internet | SOA, Web & Cloud Services | Provide enterprise-level methods for remote program execution | [42], [61], [63], [65]–[67], [70]–[72], [77], [80]–[82], [87], [91], [92], [94], [103], [108]–[110], [117], [119], [120], [124], [128], [147], [150], [153] |
| Internet | Web & Browser Applications | Enable front-end interaction with backend applications | [24], [62], [71], [75], [78], [96], [122], [153] |
| Mobile | Geo Location | Track and trace physical objects using location services | [75], [85], [90], [102] |
| Mobile | Instant Messaging | Push notifications and messages to end-users and programs | [74], [76], [96] |
| Mobile | Mobile Computing | Embed mainstream mobile devices across industrial systems | [51], [74], [85], [96], [110], [122], [125], [128] |

**Table 14 References for technology paradigms**

#### 2.5.4.2 Programming languages and frameworks

Any number of programming languages and frameworks may support the development of industrial cyber-physical systems, including those tools, language and frameworks synonymous with current and emerging paradigms. Figure 23 illustrates technology classifications for programming languages and frameworks that were extracted from the literature. Based on these classifications, programming languages and frameworks associated with enterprise technologies are the most prominent. Many of these enterprise technologies support distributed computing scenarios and demonstrate strong adoption across different sectors, but were not initially developed for low-power and resource-limited scenarios (e.g. internet-of-things). Hence, such scenarios may be better served by programming languages and frameworks associated with emerging mobile and internet-of-things technologies, which are specifically designed to address such constraints.

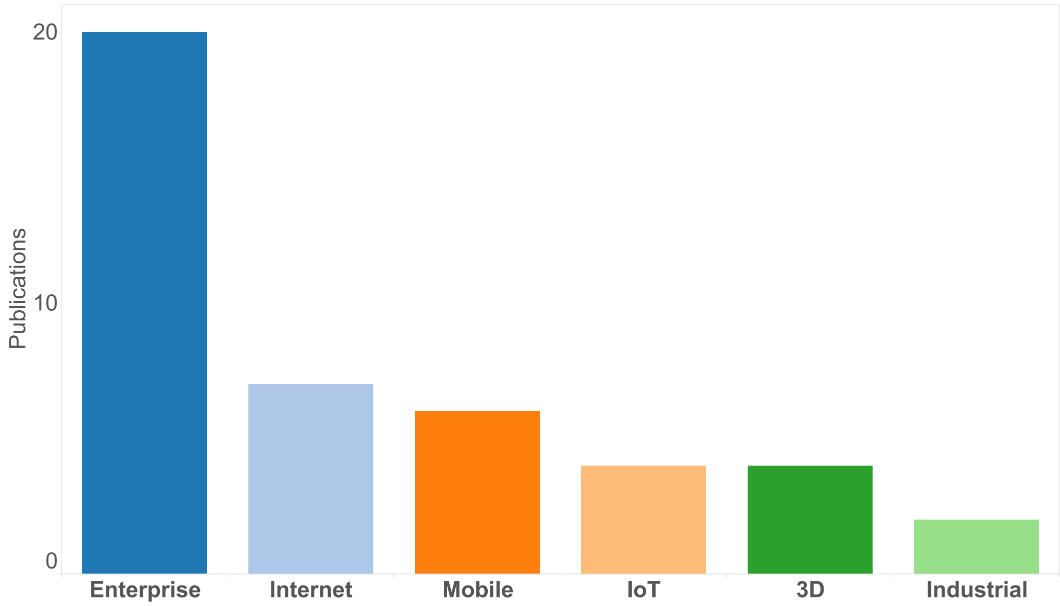



**Figure 23 Classification of programming languages and frameworks**

Figure 23 illustrates the distribution of programming languages and frameworks extracted from the literature, while Table 15 provides descriptions and references supporting the data visualisation. Of the programming languages and frameworks identified, Java was most commonly used to develop software modules for industrial cyber-physical systems. Although one could focus on Java's prominence, the broader utilisation of object-oriented programming languages seems more significant, with Java, C# and other .NET languages evident in 20% of reviewed publications. Arguably, these mature languages and frameworks may be deemed more favourable than contemporary counterparts due to the availability of stable production-ready libraries (e.g. distributed communication), significant development communities, previous development experience, and integrated tool support (e.g. development environments). Such resources are difficult to ignore, given they have the potential to greatly reduce technical effort, when compared to emerging technologies that are at an early stage.

Apart from traditional object-oriented languages and frameworks, more contemporary development technologies demonstrating some adoption include Android for developing end-user mobile applications, Java 3D for supporting additive manufacturing and immersive human-computer interaction, Bosch XDK Kit for programming the internet-of-things, and Jade for enabling decentralised multi-agent and holonic industrial systems. While these development technologies may gain more industry adoption over time, both Android and Jade platforms are particular well-aligned with Industry 4.0 design principles focusing on decentralised processing and decision-making.

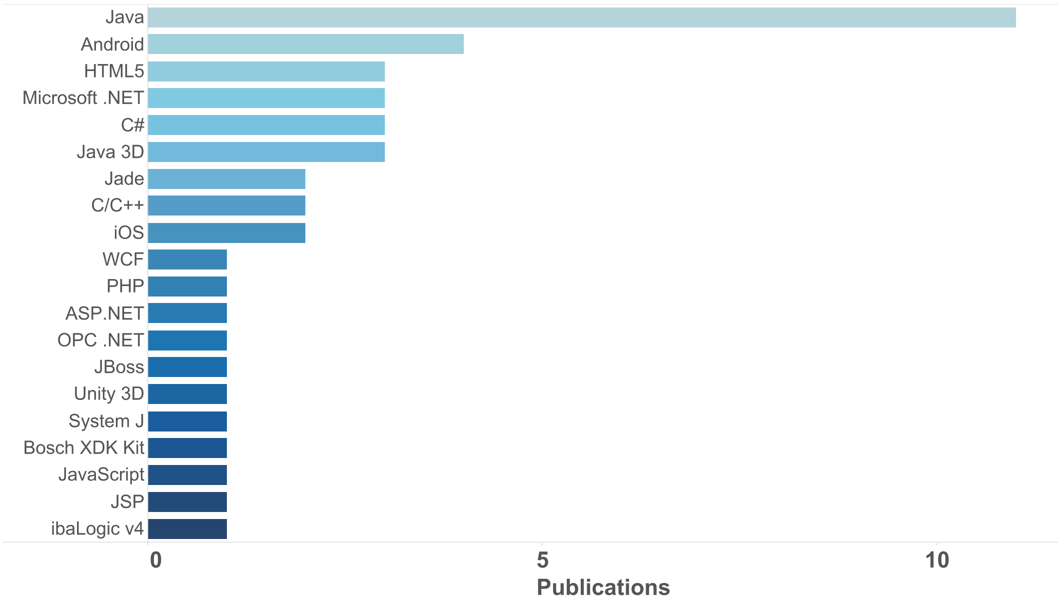



**Figure 24 Distribution of development technologies**

| Classification | Technology | Description | Reference |
|---|---|---|---|
| 3D | Java 3D | 3D API for OpenGL and Direct3D | [80], [106], [107] |
| 3D | Unity 3D | Cross-platform game engine with 3D rendering | [109] |
| Enterprise | C/C++ | General-purpose programming language | [67], [106] |
| Enterprise | C# | Standards-based general-purpose programming language | [109], [128], [142] |
| Enterprise | Java | General purpose cross-platform programming language | [76], [106], [107], [109], [117], [126], [133], [139], [150], [152] |
| Enterprise | JBoss | High-performance and scalable middleware | [77] |
| Enterprise | Microsoft .NET | Windows-based development framework for mobile, web and desktop applications | [101], [108], [128] |
| Industrial | ibaLogic v4 | System for signal processing and automation | [146] |
| Industrial | OPC .NET | .NET library for OLE Process Control | [61] |
| Internet | ASP.NET | Open source sever-side web application framework | [108] |
| Internet | HTML 5 | Language for presenting content on the web | [71], [103], [150] |
| Internet | Java Server Pages (JSP) | Server-side language for creating dynamic web applications | [152] |
| Internet | JavaScript | Client-side scripting language for interactive web applications | [152] |
| Internet | PHP | Open source general-purpose scripting language | [108] |
| Internet | Windows Communication Foundation (WCF) | Framework for building service-oriented applications | [61] |
| Internet-of-Things | Bosch XDK Kit | Development kit for IoT programmable sensors and platforms | [133] |
| Internet-of-Things | Jade | Java framework for developing intelligent agent systems | [75], [154] |
| Internet-of-Things | System J | Language for designing and programming concurrent and distributed systems | [139] |
| Mobile | Android | Linux-based mobile platform | [68], [74], [76], [117] |
| Mobile | iOS | Mobile platform for Apple devices | [62], [78] |

**Table 15 References for development technologies**

#### 2.5.4.3 Data formats and message exchange interfaces

While service-oriented computing and architectures enable messaging between disparate systems, these messages must encode data using formats that can be interpreted by participating systems (e.g. open and standard formats), and interoperate through interfaces for sending and receiving communications (e.g. RESTful). Figure 25 illustrates the distribution of data formats and message exchange interfaces extracted from the literature, while Table 16 provides descriptions and references supporting the data visualisation. Of the data formats identified, the general-purpose and human-readable Extensible Markup Language (XML) and Java Script Object Notation (JSON) are most prominent. Although encoding data using XML or JSON facilitates open, standard and human-readable data representation, XML provides a more comprehensive convention for encoding data documents. Examples of such conventions include support for metadata attributes, prefixing, and mixed content (e.g. data, images and files). These conventions contrast with JSON's lightweight format, where encoding relies predominantly on key/value pairs of strings, and hierarchal relationships are depicted using simple nesting structures. In turn, this lightweight and intuitive encoding approach can produce smaller packets to facilitate data transmission.

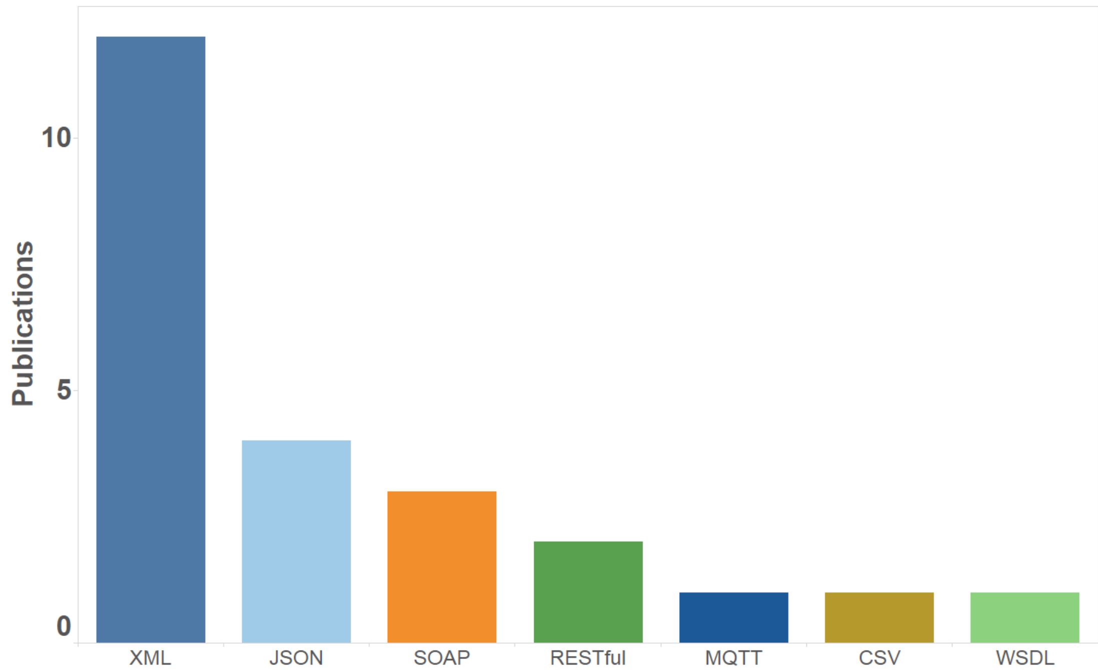



**Figure 25 Distribution of data formats and messaging**

| Method | Description | Reference |
|---|---|---|
| Extensible Markup Language (XML) | Markup language for encoding document formats for human and machine consumption | [67], [71], [75], [80], [81], [83], [88], [108], [116], [119], [128], [147] |
| JavaScript Object Notation (JSON) | Lightweight data encoding and exchange format supporting system interoperability | [80], [116], [147], [148] |
| Simple Object Access Protocol (SOAP) | Protocol specification supporting data exchange and invocation for web services | [108], [119], [128] |
| Representation State Transfer (RESTful) | Architectural style for enabling interoperability and modularity across Internet-based systems | [71], [147] |
| Message Queue Telemetry Transport (MQTT) | Lightweight publish-subscribe protocol for decentralised low-power communication | [133] |
| Comma Separated Values (CSV) | Flat file format for storing one-dimensional data | [116] |
| Web Services Description Language (WSDL) | XML-based language for describing web service specifications and rules | [108] |

**Table 16 References formats and messaging**

Encoded data messages are transmitted between disparate systems using message exchange interfaces. Based on findings from the literature, the Simple Object Access Protocol (SOAP) and Representation State Transfer (REST) are most commonly used for cyber-physical interactions. While XML and JSON are independent of data retrieval methods, they are synonymous with web service implementations based on SOAP and REST, which provide open, standard and consistent interfaces for exchanging messages and executing distributed software modules. These web services can be developed using tools and libraries from mainstream technology platforms and frameworks (e.g. Java, .NET). Although SOAP and REST can produce similar functionality (i.e. distributed execution), their underlying characteristics are different. In particular, REST is an architectural style that employs Hypertext Transfer Protocol (HTTP) transmission and semantics exclusively to relay operations to remote applications (e.g. GET = *retrieve*, POST = *create*, PUT = *update*, DELETE = *delete*), while SOAP is a full-featured XML-based message protocol for information exchange that supports many different communication protocols (e.g. HTTP, SMTP etc.). Choosing an approach depends on scenarios requirements, with REST's minimal specification potentially advantageous when lightweight data transmissions are required, and SOAP's comprehensive specification more aligned with larger data transmissions, programmatic rules and constraints, and enterprise security integration.

Of the less prominent formats and interfaces found in the literature, the Message Queue Telemetry Protocol (MQTT) seems particularly relevant to industrial cyber-physical systems supporting Industry 4.0 operations. MQTT is a lightweight machine-to-machine connectivity protocol for the internet-of-things, which utilises a publish-subscribe model to exchange information amongst disparate systems and devices. The publish-subscribe model offers many benefits over request-response models (i.e. SOAP and REST) for certain cyber-physical systems with low-power and low-compute characteristics (e.g. embedded devices used instead of computers), including (a) lower energy consumption, (b) better bandwidth efficiency, and (c) smaller footprint. These benefits are largely derived from more efficient exchanges, where the publish-subscribe model removes the need for participating systems to continually poll each other looking for updates or changes. Although not directly evident from the literature, the use of MQTT may increase as mainstream tools and technologies provide better integration for the standard, and more cyber-physical implementations progress beyond proof-of-concept (i.e. reusing existing knowledge of SOAP and REST provides an easier route for proof-of-concepts).

#### 2.5.4.4 Network infrastructures and technologies

Emerging industrial cyber-physical systems depend on networks to connect and monitor physical phenomena. Figure 26 illustrates the different type of network technologies extracted from the literature, which have been classified as those relating to (a) enterprise, (b) internet-of-things and (c) industrial networks. Generally, enterprise networks utilise mainstream technologies, process data centrally (e.g. server), support line-of-business applications and possess static network boundaries. These characteristics contrast with internet-of-things networks, which possess elastic network boundaries (e.g. mobile device or sensor residing off-site to relay information to industrial systems and operations) and decentralised processing capabilities to deliver scalability, flexibility and pervasive computing. Interestingly, industrial networks share characteristics with both enterprise and internet-of-things computing, with static network boundaries traditionally used to define control networks, and embedded computing utilised to enable decentralised control operations.

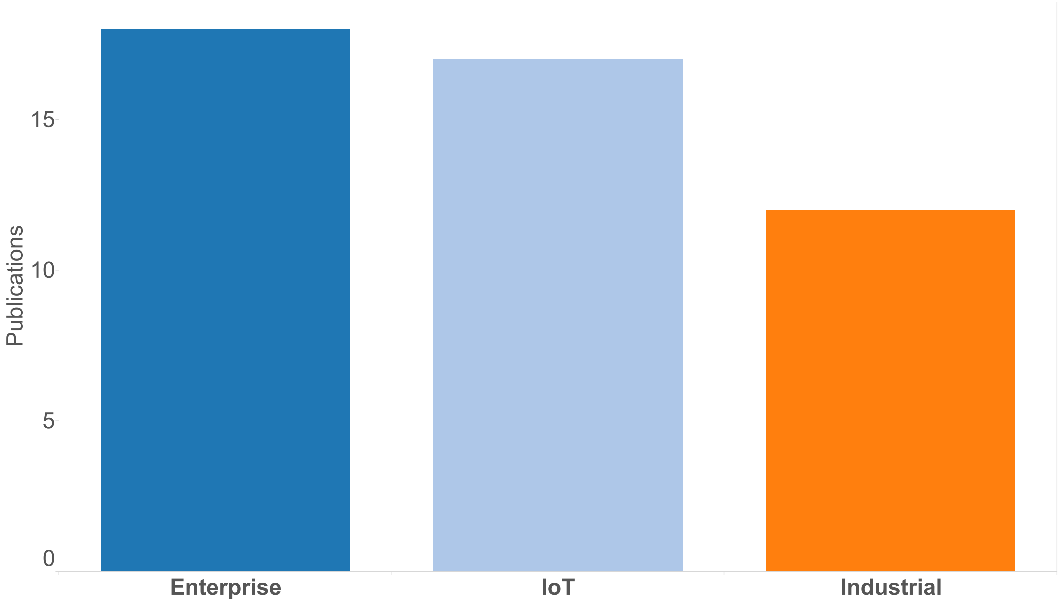



**Figure 26 Classification of network technologies**

Figure 27 illustrates the distribution of network technologies extracted from the literature, while Table 17 provides descriptions and references supporting the data visualisation. Of the networking technologies found in the literature, Wireless Local Area Networks (WLAN) are the most prominent. These networks provide a wide-range of devices (e.g. mobile, laptop etc.) with untethered connectivity, which can support communications between software and hardware components residing within the physical layer (i.e. factory floor) of cyber-physical systems. Another prominent wireless networking technology that goes beyond connectivity alone is that of Wireless Sensor Networks (WSN). WSN's consist of distributed autonomous sensors that monitor physical conditions (e.g. temperature), and execute operations to maintain a desired state (e.g. minimum temperature) - these operations may be undertaken by individual sensors, or through collaboration with other sensors. In terms of relevance to industrial applications, WSN's can enable sensor deployment in difficult to reach areas (e.g. rotating blade), and provide a minimally evasive and scalable approach for extending sensing capabilities (e.g. disruptions or costs associated with cabling).

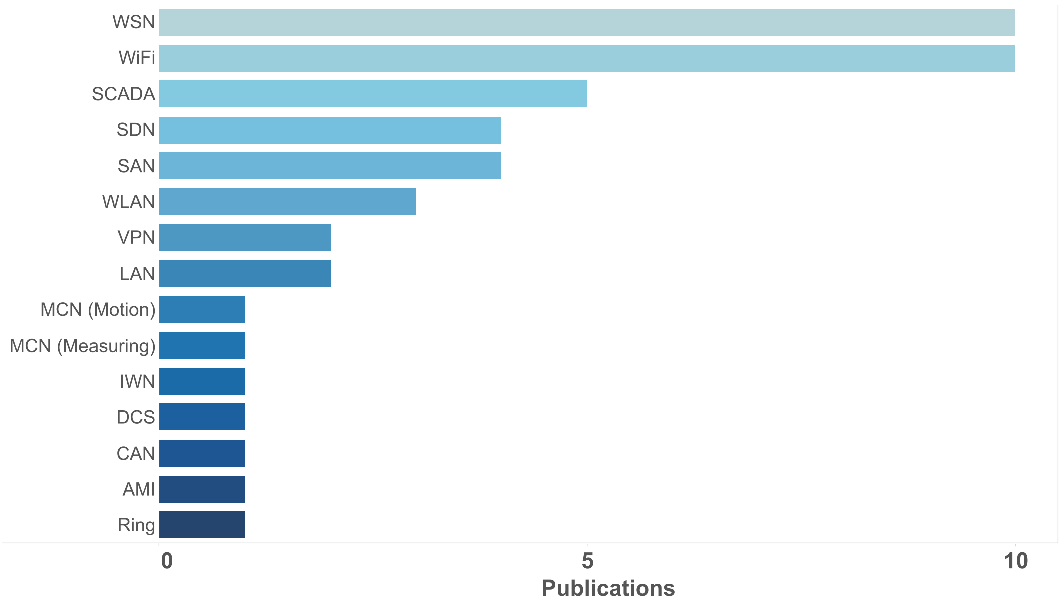



**Figure 27 Distribution of network technologies**

In the context of Industry 4.0 and industrial cyber-physical systems, Software Defined Networks (SDN) may represent an important network technology. SDN's are programmable computer networking abstractions, which enable the programmatic creation of dynamic, flexible and scalable networks. Thus, SDN's may be used to centralise network provisioning, refine security control, lower operating costs, and virtually integrate networks (e.g. WSN and cloud infrastructures). Due to the dynamic, flexible and scalable nature of SDN's, they may be particularly useful when supporting data-intensive industrial applications (e.g. big data, virtualisation etc.), and enabling industrial cyber-physical systems to enact self-configuring network operations without the need for human intervention.

#### 2.5.4.5 Implementation observations relevant to thesis

At present, mature and well-established technologies are being chosen to implement industrial cyber-physical systems, with fewer real-world demonstrations of contemporary cyber-physical and internet-of-things technologies. The strongest themes within the literature highlight the use of *cloud computing*, *service-oriented computing*, *object-oriented programming* and *wireless sensor networks* to support cyber-physical system implementation. Apart from wireless sensor networks, these technologies are synonymous with mainstream and enterprise computing, where they are typically used to centrally store and process information. However, design principles and commentary relating to Industry 4.0 promote the notion of decentralised real-time decision-making and control on low-power and low-compute platforms (e.g. machine-to-machine communication undertaken on the outer edge of networks). Thus, emerging technologies supporting decentralised computing architectures and low-latency communications should be considered for industrial cyber-physical system implementation.

The decentralised and autonomous technologies identified from the literature include *multi-agent systems*, *holonic systems* and *fog computing*, with each possessing unique methods, concepts and tools to support implementation. Of these technologies, fog computing may be considered the least mature, but the extension and relationship with cloud computing provides the benefit of familiarity, which one might hope reduces the friction commonly associated with industrial technology adoption. In addition, some similarities associated with developing cloud and fog computing services should reduce the learning curve and decrease implementation time for engineers and developers. The primary challenges relating to fog computing implementations include (a) insufficient supporting frameworks mean developers must write and manage more code, (b) less prescription regarding the low-level and high-level technical components needed to support the implementation, and (c) getting buy-in from internal stakeholders regarding the introduction of new methods or technologies.

| Classification | Network | Description | Reference |
|---|---|---|---|
| Enterprise | Local Area Network (LAN) | Network connecting devices using cables within a limited area | [91], [128] |
| Enterprise | Ring Network | Network topology where each node connects to two other nodes | [129] |
| Enterprise | Virtual Private Network (VPN) | Network connecting two private networks using public wires | [69], [119] |
| Enterprise | Wireless Local Area Network (WLAN) | Short-range local network connecting devices wirelessly that can be implemented using different wireless standards | [62], [64], [65], [83], [87], [91], [96], [101], [114], [152] |
| Enterprise | Wi-Fi | Wireless implementation of IEEE 802.11 standard | [68], [83], [91] |
| Industrial | Advanced Meter Infrastructure (AMI) | Network of sensors and meters capturing data for load control and energy management | [144] |
| Industrial | Control Area Network (CAN) | Enables microcontrollers and devices to communicate directly without host computers | [68] |
| Industrial | Distributed Control System (DCS) | Computerised control system for connecting and managing controllers | [120] |
| Industrial | Sensor Area Network (SAN) | Interactive and cooperative sensing network comprised of wired and wireless sensors | [64], [82], [92], [130] |
| Industrial | Supervisory Control and Data Acquisition (SCADA) | Software and hardware components for remotely controlling and managing industrial processes | [61], [77], [81], [120], [124] |
| IoT | Industrial Wireless Network (IWN) | Robust and resilient wireless sensing for industrial environments | [96] |
| IoT | Motion Control Network | Embedded sensing to obtain motor operation data | [93] |
| IoT | Measuring Control Network | Embedded sensing to obtain precise position information | [93] |
| IoT | Software Defined Network (SDN) | Programmatically initialised controlled and managed networks | [96], [97], [147], [148] |
| IoT | Wireless Sensor Network (WSN) | Distributed and autonomous sensors used to monitor physical or environmental conditions | [63], [65], [74], [84], [85], [106], [108], [110], [119], [148] |

**Table 17 References for network paradigm**

## 2.5.5 Standards for industrial cyber-physical systems

### 2.5.5.1 Connectivity and communication

Emerging industrial cyber-physical systems depend on standards and methods to promote seamless connectivity. If these standards are embraced during system design and implementation, the produced industrial cyber-physical systems should exhibit openness and interoperability. Figure 28 illustrates classifications for connectivity and communication standards extracted from the literature, with the main categories including (a) internet-of-things, (b) networking, (c) industrial automation, (d) internet technologies and (e) mobile technologies. Of these classifications, standards associated with the internet-of-things are most prominent, which may be due to Industry 4.0 requirements pertaining to low-power, pervasive, and decentralised networks. These emerging networks and standards differ from traditional enterprise and industrial networks, with the latter possessing greater compute and power resources, static network boundaries, and centralised processing and storage topologies (e.g. clients connect to server for intelligence, processing and reporting capabilities). Although there are many variables to consider before adopting particular standards, those supporting decentralised decision-making have the advantage of being directly aligned with Industry 4.0 design principles, while such approaches may also facilitate system scalability, and low-latency processing (e.g. real-time performance).

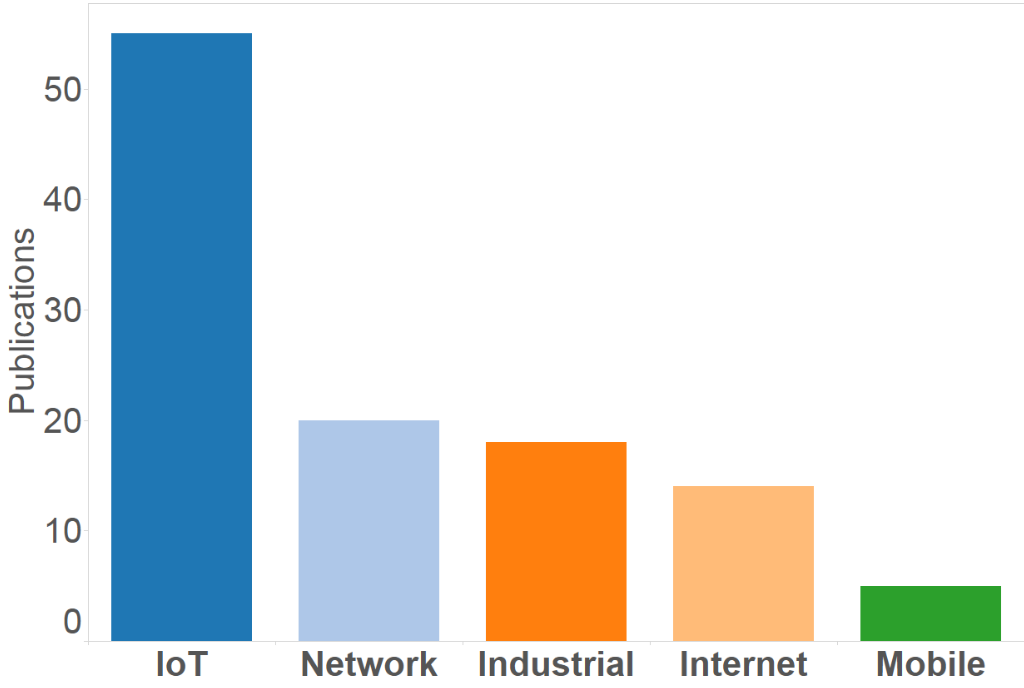



**Figure 28 Classification of connectivity and communication standards**

Figure 29 illustrates the distribution of connectivity and communication standards extracted from the literature, while Table 18 provides descriptions and references to support the data visualisation. The most prominent connectivity and communication standard utilised in industrial cyber-physical systems was Radio Frequency Identification (RFID). This wireless tagging technology embeds or attaches information to physical objects, which facilitates seamless object identification and the transmission of metadata using RFID readers. Thus, many applications using RFID technology typically embody some aspect of tracking (e.g. product moving downstream). Although less prominent in the literature than RFID, Near Field Communication (NFC) supports similar usage scenarios, and can be considered a subset of RFID. One of the key differentiators of both technologies relates to operating range - NFC requires tags and readers to be in close proximity (e.g. 20cm), while RFID tags can be read from up to 15 meters.

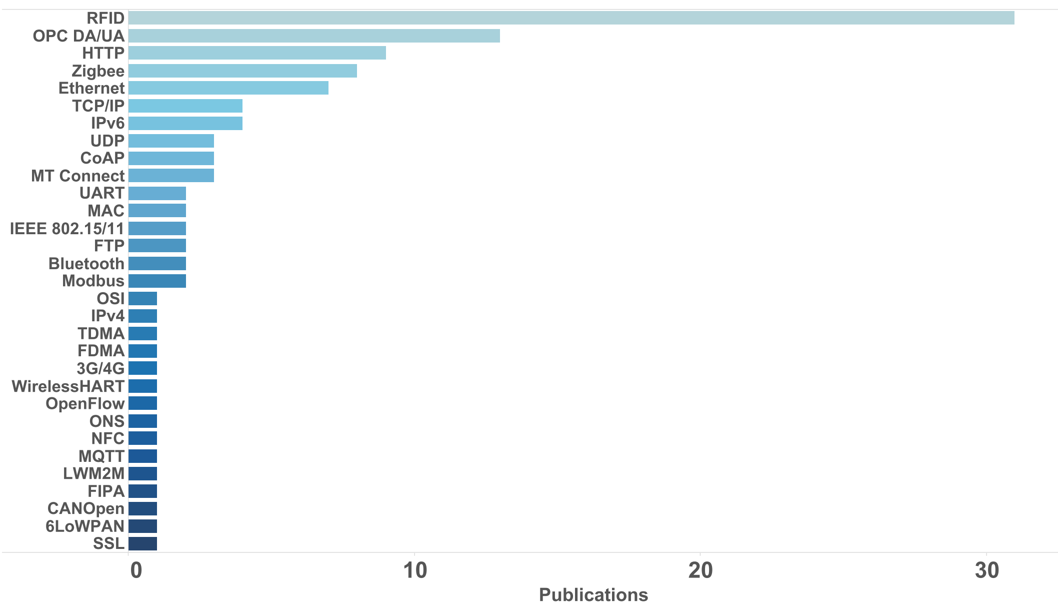



**Figure 29 Distribution of connectivity and communication standards**

While RFID, ZigBee and NFC standards naturally align with the pervasive, low-power and decentralised aspects of industrial cyber-physical systems, more traditional control and automation communication standards are also needed to facilitate legacy connectivity. Indeed, the findings from the literature highlight the use of OLE Process Control Unified Architecture (OPC UA), MT Connect and Modbus within industrial cyber-physical systems. Similarly, the adoption of existing open internet standards and conventions are also evident, with Hypertext Transfer Protocol (HTTP) being widely used to support messaging between factory, enterprise and cloud platforms. Thus, the availability of emerging technology standards, should not mean current or legacy standards are discarded, but rather form an important part of an industrial cyber-physical system's compatibility and integration strategy.

A number of less prominent emerging standards were found in the literature, which may increase in popularity over the coming years due to natural alignments with Industry 4.0 design principles and requirements. Firstly, agent computing and architectures have been demonstrated in other domains as a means of promoting decentralised decision-making, and therefore the Foundation for Intelligent Physical Agents (FIPA) could gain more acceptance for industrial cyber-physical system implementation. Secondly, software defined networks enable the programmatic creation and administration of computer networks, which may be used to good effect to create highly scalable and flexible networks for the many devices and systems that shall comprise Industry 4.0 operations. Hence, standards such as OpenFlow (or similar) shall be used to abstract conventional computer networking, much in the same way cloud computing removed the need to configure physical servers. Finally, lightweight communication protocols that are more suited to low power and compute environments (e.g. dispersed smart sensors) should gain wider adoption, which should eventually result in Message Queue Telemetry Transport (MQTT) replacing HTTP for distributed messaging and communication.

| Classification | Standard | Description | References |
|---|---|---|---|
| Internet | File Transfer Protocol (FTP) | Protocol for transferring files between client and server | [116], [128] |
| Internet | Hypertext Transfer Protocol (HTTP) | Protocol supporting distributed hypermedia applications | [61], [80], [96], [107], [116], [119], [124], [128], [147] |
| Internet | Secure Socket Layer (SSL) | Cryptographic protocols for secure network communications | [119] |
| IoT | Constrained Application Protocol (CoAP) | Application protocol for Internet-based resource-constrained devices | [80], [133], [148] |
| IoT | Control Area Network Open (CANOpen) | Communication protocol for automation and embedded systems | [61] |
| IoT | Foundation for Intelligent Physical Agents (FIPA) | Collection of standards for heterogeneous and interacting agents | [124] |
| IoT | Internet Protocol Version 6 (IPv6) | Communication protocol for node identification and traffic routing throughout the Internet | [63], [64], [82], [133] |
| IoT | IPv6 Over Lower Power Wireless Personal Area Network (6LoWPAN) | Conceptual model supporting devices with low-power and limited processing capabilities | [87] |
| IoT | Lightweight Machine-to-Machine (LWM2M) | Communication protocol for resource-constrained device management and data transmissions between clients and servers | [133] |
| IoT | Message Queue Telemetry Transport (MQTT) | Lightweight publish/subscribe message protocol for distributed communication | [133] |
| IoT | Near Field Communication (NFC) | Communication protocols enabling devices to exchange information when they are in close proximity | [96] |
| IoT | Object Naming Services (ONS) | Lookup service for discovering product and service information using an electronic product code | [99] |
| IoT | OpenFlow | Communication interface for programmatically manipulating network control and routing (i.e. software-defined networks) | [97] |
| IoT | Radio Frequency Identification (RFID) | Electromagnetic tagging system for object tracking and identification | [63], [65], [73], [74], [80]–[82], [84], [85], [87], [90]–[93], [96], [99], [101], [102], [108], [110], [114], [115], [117], [118], [121], [123], [130], [141], [142], [152] |
| IoT | Wireless Highway Addressable Remote Transducer (HART) | Field device protocol for self-organising and self-healing mesh architectures | [148] |
| IoT | Zigbee | High-level specification for low-power and low-bandwidth wireless applications | [61], [64], [65], [74], [82], [119], [148], [152] |
| Mobile | 3G/4G | Mobile communications technology supporting voice and data transmission | [64] |
| Mobile | Bluetooth | Wireless standard for building close-proximity personal area networks | [64], [65] |
| Mobile | Frequency Division Multiple Access (FDMA) | Method for managing radio frequency communications using multiple sub-channels | [73] |
| Mobile | IEEE 802.15 | Group of several standards for different applications of wireless personal area networks | [65], [96] |
| Mobile | Time Division Multiple Access (TDMA) | Method for managing radio frequency communications using timeslots | [73] |
| Network | Ethernet | Network protocol controlling data transmissions across local area networks | [62], [64], [65], [67], [68], [73], [152] |
| Network | Internet Protocol Version 4 (IPv4) | Communication protocol for node identification and traffic routing throughout the Internet | [64] |
| Network | Machine Access Control (MAC) | Unique identifier for network interfaces | [73], [74] |
| Network | Open Systems Interconnection (OSI) | Standard communication model for telecommunication and computer systems | [119] |
| Network | Transmission Control Protocol/Internet Protocol (TCP/IP) | Protocols describing the addressing, transmission and routing of network packets | [63], [111], [128], [146] |
| Network | Universal Asynchronous Receiver Transmitter (UART) | Hardware supporting configurable data format and transmission speeds | [68], [71] |
| Network | User Datagram Protocol (UDP) | Connectionless protocol for sending short messages to hosts | [111], [128], [133] |

**Table 18 References for connectivity and communication standards**

#### 2.5.5.2 Control and automation

Given many industrial cyber-physical systems shall extend current control and automation networks, knowledge pertaining to prominent standards and processes may be necessary to inform design and implementation. Figure 30 illustrates the main classifications for control and automation standards extracted from the literature, including those standards (a) supporting device integration and interoperability, (b) guiding the creation and implementation of control logic, and (c) governing the design and implementation of control systems.

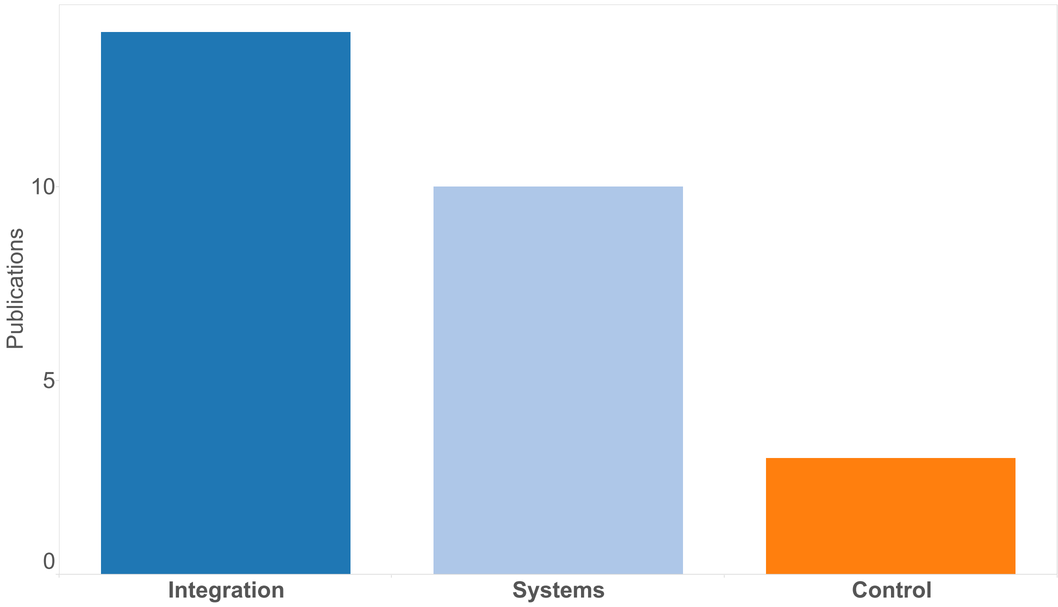



**Figure 30 Classification of control and automation standards**

Figure 31 illustrates the distribution of control and automation standards extracted from the literature, while Table 18 provides descriptions and references to support the data visualisation. Of the specific control and automation standards identified, the ISA-95 standard was most prominent. ISA-95 is a five-part standard for establishing interfaces between automation and business systems, with the intention of achieving factory-to-enterprise integration. While the literature predominantly refers to the standard as ISA-95, the standard also may be referred to internationally as IEC/ISO 62264. A key benefit of ISA-95 is that it provides organisations with a common vocabulary and methodology upon which to centre operations, and guide technical integrations between business and industrial systems. In the context of industrial cyber-physical systems, ISA-95 may reduce the complexity associated with data and system integration, when compared to industrial environments that are not standards compliant.

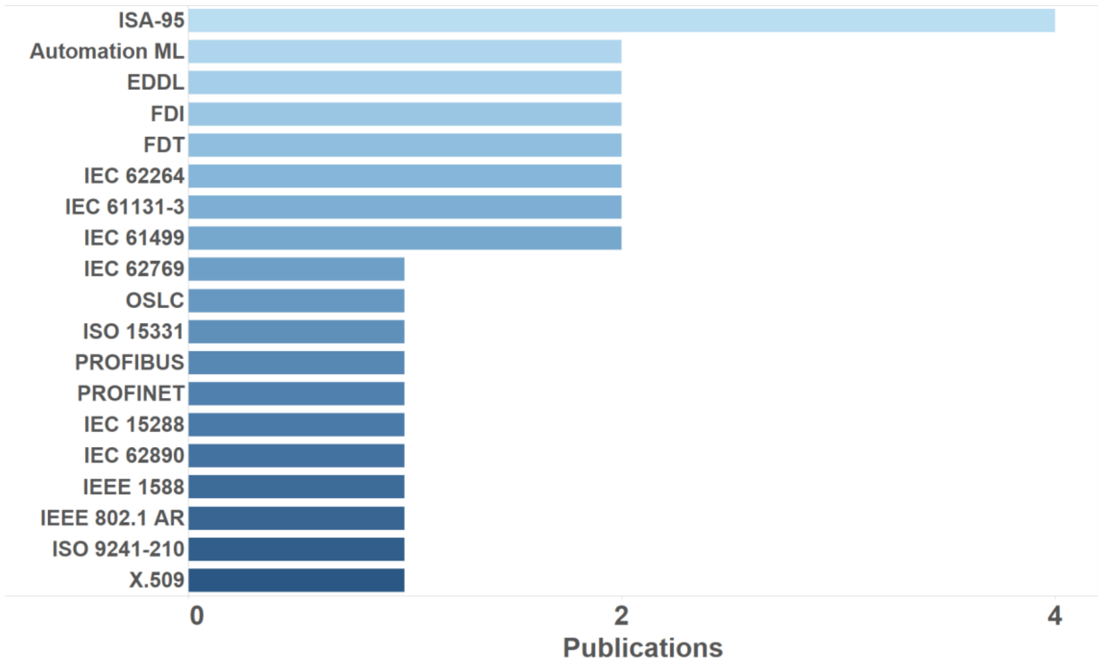



**Figure 31 Distribution of control and automation standards**

Of the control standards identified, AutomationML was identified as potentially important to Industry 4.0. Automation ML is an XML-based open standard for storing and exchanging industrial engineering and automation information, which enables industrial systems and tools to interoperate seamlessly. The format comprises four distinct types of information – (1) topology consisting of properties, and relationships to other components, (2) geometry containing graphical and dimensional information about the component, (3) kinematics relating to chains of dependency that affect motion planning, and (4) logic comprising algorithms, processing instructions and read-write operations. Although not demonstrated in the literature, industrial cyber-physical systems could update these dimensions to realise self-configuration.

The prominent standards relating to device-level integration and interoperability include Electronic Device Description Language (EDDL), Field Device Tool (FDT) and Field Device Integration (FDI). Of these standards, EDDL simply describes the accessible information on digital devices, which may be used by process control systems to support device diagnostics, configuration and collaboration. As internet-of-things and other Industry 4.0 technologies are adopted in the factory, technologies such as EDDL shall be needed to manage and control the use of on-premises devices. Indeed, the information layer of RAMI 4.0 utilises EDDL for device management, which emphasises EDDL's relevance to Industry 4.0 and cyber-physical systems. Although the device descriptors provided by EDDL are important, the FDI standard combines EDDL and FDT with the intention of unifying device integration for automation and control environments, and thereby providing end-users with the ability to present real-time information from the factory with minimum effort. Although similar standards already exist, some of the primary benefits associated with FDI include platform and protocol independence, open specification, international standardisation, and OPC-UA compatibility. Thus, the FDI standard demonstrates strong alignment with Industry 4.0 design principles (e.g. use of open standards), while also facilitating compatibility with other industry standards (e.g. OPC-UA).

#### 2.5.5.3 Standards observations relevant to thesis

Historically, the adoption of communication and control standards in real-world industrial and manufacturing environments has proved challenging. However, modern facilities are certainly more aware of the benefits of standards, such as reducing implementation costs and promoting system interoperability, to name a few. Given these benefits are now well-known, one may question why standards are not commonplace throughout modern facilities. The following points discuss some reasons why achieving complete standards adoption within many factories may prove difficult;

- **Lifetime of hardware and software** in industrial environments can extend to decades, which provides engineers and technology personnel with unique challenges relating to legacy system management and integration. Given these legacy controllers and systems may use proprietary technologies, integrating them within the factory's current standards can be difficult without some element of technology replacement. Therefore, legacy artefacts that cannot be replaced can represent gaps in a facility's standards policy and coverage.

- **Operation technology expertise** in industrial environments traditionally outweighs information technology expertise. Although this category of technology expertise is imperative to traditional manufacturing, operation technology personnel may not be familiar with mainstream and enterprise communication standards, which continue to permeate the factory floor due to the introduction of more contemporary technologies (e.g. cloud-based real-time dashboards for production lines). Thus, sufficient knowledge of contemporary technologies shall be needed to enforce holistic standards policies.

The use of communication standards synonymous with the internet-of-things was prevalent throughout the literature, with RFID or HTTP evident in almost half of the reviewed publications. Although the prevalence of internet-of-things related standards may initially appear to contradict previous comments regarding impediments relating to contemporary technology adoption across industrial environments, many of these publications relate to proof-of-concept and controlled demonstrations (e.g. offline lab environment), and therefore are not real-world deployments. However, these findings emphasise the intent of the research community to develop cyber-physical applications using embedded sensors and internet-of-things technology, which shall invariably provide the foundations for many Industry 4.0 operating scenarios.

In comparison to communication standards, control and automation standards were given much less consideration in the literature pertaining to industrial cyber-physical systems. This imbalance may be attributed to the strong emphasis on computing and technology throughout the literature, where the contributing researchers do not possess knowledge of control or automation standards. Of those control standards identified from the literature, standards focused on device and systems integration are most prevalent, which makes sense given the disparate and disconnected nature of many real-world industrial environments.

Given the number of options pertaining to standards, facilities must appreciate that standards compliance depends on the adoption of standards at different levels within their information and cyber-physical systems. Indeed, choosing and committing to one standard may be of little benefit where lower-level components or services depend on proprietary technologies, and therefore the promotion of "standards thinking" at each level of system design seems prudent. In addition, facilities should not confuse *standards* and *open standards*. Although both can decrease the cost of implementation and improve interoperability, the former can still manifest scenarios where facilities are locked-in to using particular hardware, software and services.

As real-world industrial operations and environments are highly secure and extremely risk adverse, cyber-physical system implementations related to this research should aim employ standards that already exist within the facility, rather introducing new standards which are not understood by internal stakeholders. This approach enables any cyber-physical implementation to inherit policies, procedures and security measures, while also limiting potential disruption and quality assurance procedures that may arise from the introduction of new standards. In addition, proof-of-concept projects and research demonstrations are unlikely to possess sufficient influence or resources to change internal policies pertaining to standards. Finally, the multidisciplinary nature of this research also means sufficient time, attention or expertise cannot be realistically given to evaluating, comparing and integrating control and automation standards that cover factory and enterprise operations.

| Classification | Standard | Description | References |
|---|---|---|---|
| Control | Automation Markup Language (Automation ML) | Data format for storing and exchanging factory information | [81], [120] |
| Control | IEC 61131-3 | Software architecture and programming languages for controllers | [77], [133] |
| Control | ISO 15331 | Standard for managing industrial manufacturing data | [124] |
| Control | PROFIBUS | Standard for fieldbus communication | [94] |
| Control | PROFINET | Standard for fieldbus communication over industrial ethernet | [135] |
| Integration | Electronic Device Description Language (EDDL) | Technology for describing information accessible on digital devices | [42], [120] |
| Integration | Field Device Integration (FDI) | Unifying device integration technology encompassing EDDL, FDT and OPC UA | [42], [120] |
| Integration | Field Device Tool (FDT) | Technology for communicating and configuring field devices | [42], [120] |
| Integration | IEC 62264 | International standard for enterprise-control system integration based on ISA-95 | [120], [124] |
| Integration | IEC 62769 | Standard for technology mapping FDI concepts | [42] |
| Integration | ISA-95 | Standard for automating interfaces between enterprise and control systems | [71], [77], [83], [124] |
| System | IEC 15288 | Standard for system engineering processes and lifecycles | [124] |
| System | IEC 61499 | Generic model for distributed control systems | [67], [77] |
| System | IEC 62890 | Lifecycle management for industrial processes, control and automation | [120] |
| System | IEEE 1588 | Protocol for synchronising network clocks to realise sub-microsecond control | [129] |
| System | IEEE 802.1 AR | Cryptographic and immutable device identifier | [95] |
| System | ISO 9241-21 | Standard focused on ergonomics for human-computer interaction | [76] |

**Figure 32 References for control and automation standards**

## 2.6 Chapter conclusions

This chapter considered industrial cyber-physical systems in the context of the impending smart manufacturing revolution and Industry 4.0, while also identifying current perspectives, approaches and applications in the field. Thus, the review intends to capture, synthesise and present highly contemporary and multidisciplinary information relating to disparate technologies, methods and challenges, with the intention of informing collaboration across control, engineering and software disciplines. Indeed, insufficient interdisciplinary context could be contributing to the manufacturing domain demonstrating poor adoption of industrial cyber-physical system and internet-of-things technology. Other possible factors inhibiting adoption include the integration of legacy systems, compliance and quality policies, concerns regarding performance and security, and inadequate technology prescription.

After observing the current state of the field, there is an obvious need for unifying design methodologies to formalise the multidisciplinary design of industrial cyber-physical systems for Industry 4.0, incorporating the necessary control, software and engineering perspectives to provide facilities with holistic, standardised and prescribed approaches to guide implementation. Ideally, the unifying design methodology should integrate existing modelling approaches (e.g. UML or SysML) to simplify adoption, and focus on prescribing the underlying processes that connects these models. Thus, the remainder of this thesis focuses on the creation of a *unified design methodology* that connects some of the disparate modelling approaches from the literature, and facilitates interdisciplinary comprehension, development and implementation of industrial cyber-physical systems. Thereafter, the proposed unified design methodology is employed to develop and deploy an industrial cyber-physical system within the industrial partners manufacturing facility, which serves to integrate and demonstrate many of the contemporary technologies (e.g. machine learning, internet-of-things and big data) on the facility's Industry 4.0 adoption roadmap.

# Chapter 3
# Design Methodology

## 3.1 Chapter introduction

This chapter presents a multidisciplinary design approach for industrial cyber-physical systems and auxiliary architectures, which support embedded predictive analytics (e.g. machine learning) for Industry 4.0 engineering applications. Given the reported shortcomings of industrial cyber-physical system design found in the literature, a novel unified design methodology addressing conceptual, software and performance concerns for industrial cyber-physical system design is proposed. Figure 33 illustrates the unified design methodology's phases, beginning with conceptual domain modelling, and culminating with performance analysis.

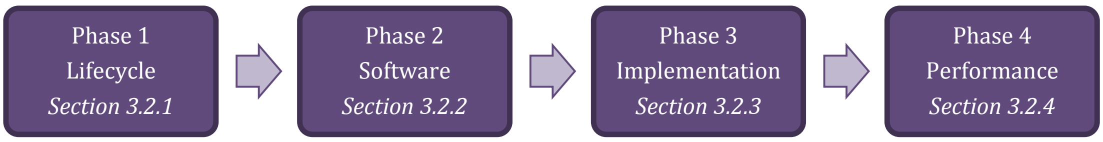



**Figure 33 Proposed design process and transitions**

## 3.2 Design process

Given the contemporary and multidisciplinary nature of industrial cyber-physical systems, formal design methodologies and architectures must emerge to support implementation. Existing design approaches found in literature may be classified as conceptual, software or mathematical modelling. The approaches chosen depend on the concerns of the system designer, with conceptual models suited to establishing domain understanding, software models needed to support technical implementation, and mathematical models used to simulate performance. However, designing industrial cyber-physical systems for Industry 4.0 are too complex to rely on a single modelling approach, and therefore, the proposed unified design methodology focuses on providing a framework to connect and integrate multiple modelling approaches, while ensuring technical implementations adhere to Industry 4.0 design principles, stakeholder concerns and minimum performance specifications. Figure 34 illustrates the proposed unified design methodology, while each phase is summarised below;

- **Phase 1** develops a common representation of data flows supporting industrial analytics and cyber-physical interactions in the factory. This phase produces a conceptual architecture depicting systems, component, and devices to support the building and deployment of analytics models.

- **Phase 2** produces formal software models to support implementation. This phase synthesises conceptual models to derive static and dynamic models of the system, which reduces functional ambiguity by providing multiple perspectives of the proposed system.

- **Phase 3** implements an industrial cyber-physical system capable of delivering real-time embedded analytics in the factory. This phase commissions technologies and develops components to realise system implementation, which are continuously evaluated using Industry 4.0, stakeholder and functional acceptance criteria.

- **Phase 4** measures system performance (i.e. worst-case execution) to determine suitability for different industrial scenarios. This phase measures and observes system operations for a particular timeframe, before building a simulation to estimate worst-case execution time (i.e. guaranteed execution time).

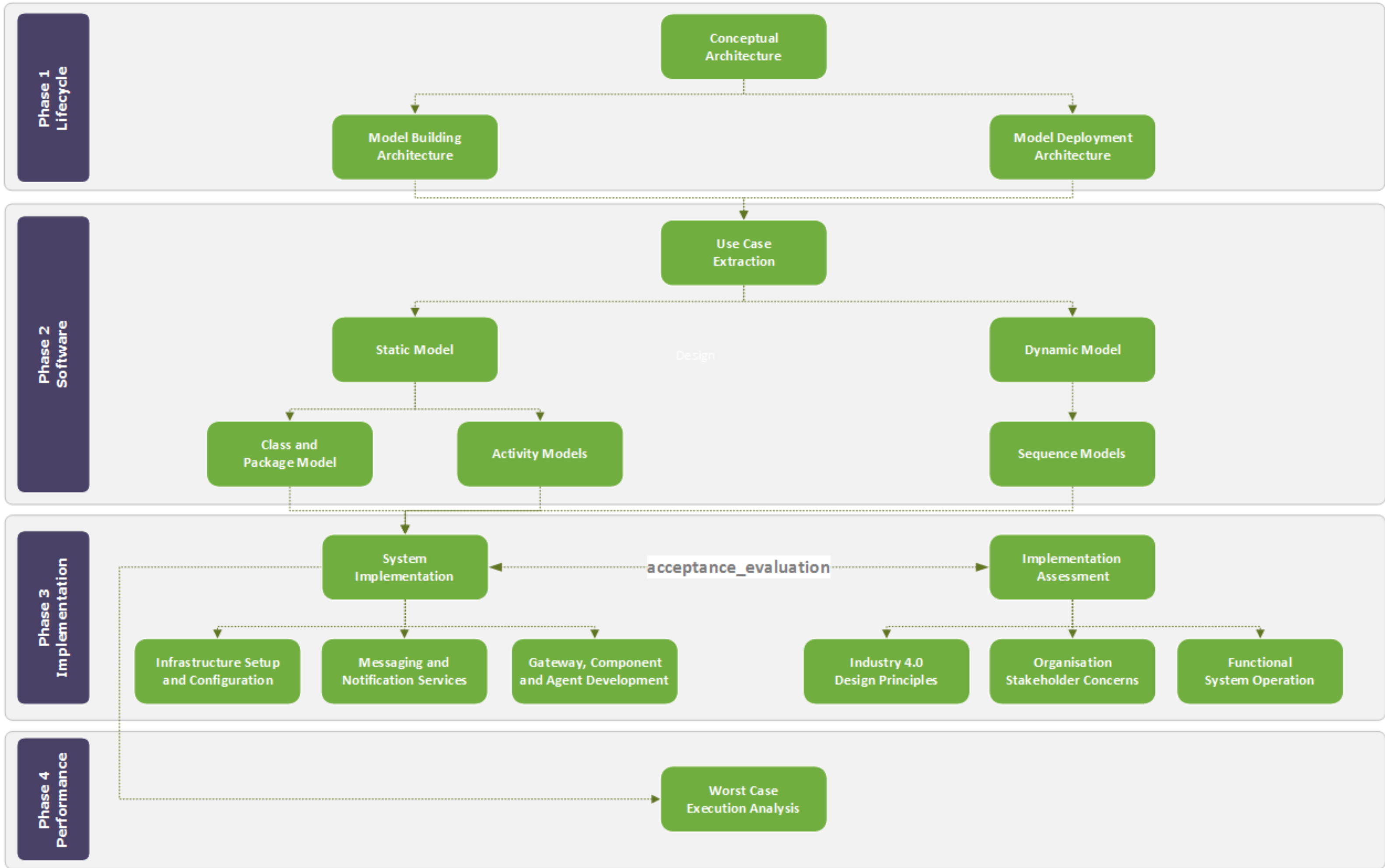



**Figure 34 Proposed unified design methodology**

### 3.2.1 Phase 1: Lifecycle model

This phase aims to produce a conceptual lifecycle model that can be easily interpreted by technical and non-technical teams from numerous disciplines. The lifecycle overlays data flows across the factory, and conceptualises technical components needed to bridge gaps in connectivity, integration, contextualisation and interoperability. Given each factory's technical ecosystem shall differ, lifecycle modelling depends extensively on investigation, elicitation and collaboration with internal stakeholders. Once sufficiently refined, the conceptual lifecycle can be used to inform the creation of technical software models (i.e. phase 2).

Figure 35 illustrates the lifecycle modelling process for producing an industrial lifecycle. The process begins with fundamental requirements gathering and design steps, before identifying components needed to facilitate a closed-loop model building (e.g. explore data, train model, test model and deploy model) and model execution (e.g. send real-time data to production-ready model to make prediction about factory operations) lifecycle. The individual sub-processes and steps for this phase are discussed in the following sections.

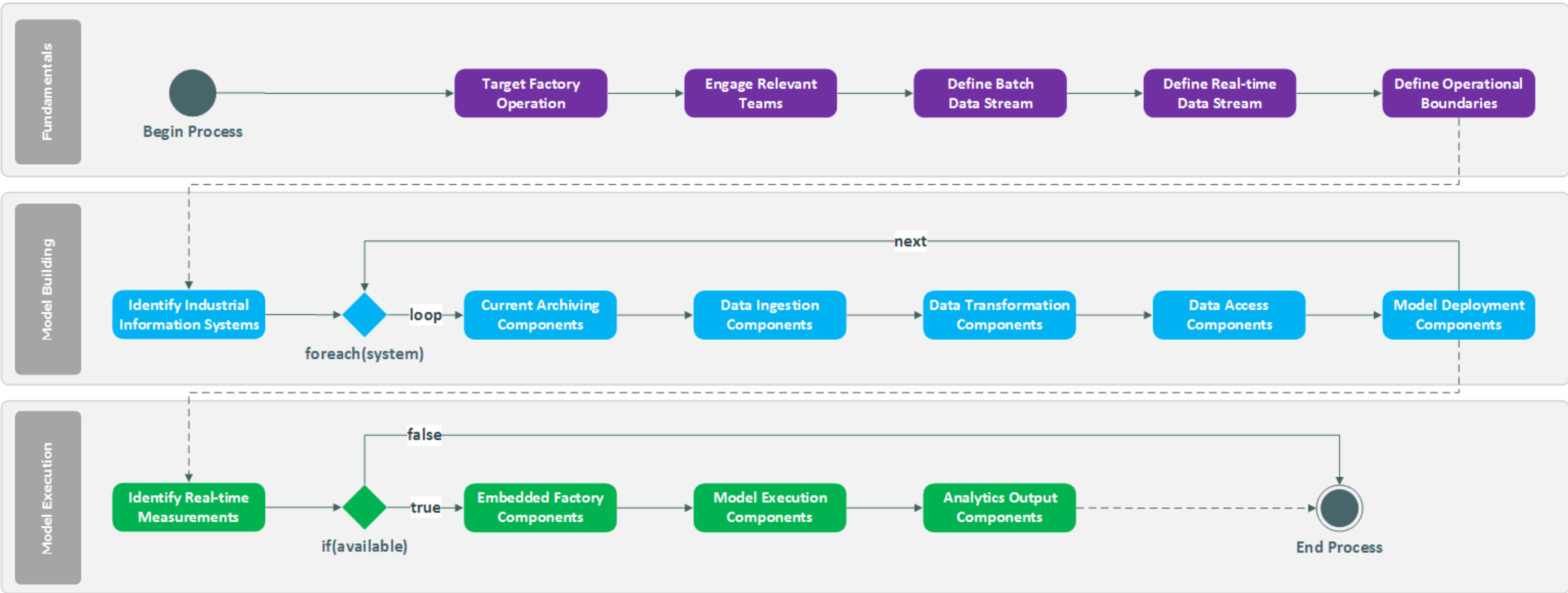



**Figure 35 Conceptual modelling process**

### FUNDAMENTALS

Fundamental actions define the focus, scope and boundaries of the lifecycle. Initially, the particular area of operations (e.g. energy) being considered for cyber-physical transformation should be specified (e.g. energy), which enables researchers and practitioners to focus their attention on specific operational teams, processes and information systems. Thereafter, the identified operational teams should be engaged to derive an appropriate understanding of current operations and systems before undertaking modelling activities.

An important aspect of industrial cyber-physical system design not given due consideration in the literature relates to the management of information system latency. Given industrial systems can demonstrate both batch and real-time operations (e.g. building systems control environmental conditions in real-time, but can archive operating data in batch), these operating latencies must be incorporated to ensure components, systems and operations are appropriately classified. Typically, batch streams contain components to support model building (i.e. train machine learning model), while real-time streams contain components to support model execution (i.e. inform factory operations).

Finally, defining operational boundaries partitions the roles and responsibilities of industrial teams and personnel, while also highlighting natural interdisciplinary relationships. These boundaries can be aligned with current operational units (e.g. operational technology, information technology etc.) to reuse existing skills and knowledge, which should limit the amount of upskilling required to deliver cyber-physical technologies. Considering there are no widely accepted standards, methods or practices for designing contemporary industrial cyber-physical systems, operational boundaries can provide structure, understanding and formalism that could otherwise be overlooked during the design process.

### MODEL BUILDING

Model building incorporates components needed to support data access, exploration and modelling. These components are aligned with common tasks one would undertake to construct a pipeline to build data-driven models (e.g. collect data, clean data etc.). The ultimate objective of model building is to produce high-quality and insightful analytics models, which can inform decision-making and positively impact factory operations. However, given such models are entirely dependent on the availability of accurate historical operating data, current information systems relating to the target factory operations (e.g. energy) must be identified. Once these industrial information systems are known, the associated components for archiving, ingestion, transformation, access and deployment should be added to the conceptual lifecycle model (i.e. appropriate operational area and data latency stream). Where certain technical components are not found, they should be added to the lifecycle and highlighted for implementation. A similar process should be followed when components are inadmissible due to conflicts with particular Industry 4.0 design principles (e.g. closed and proprietary components). The following points summarise the main types of components related to model building;

- **Data archiving components** periodically collect operational data and updating an underlying historical data repository
- **Data ingestion components** facilitate integration scenarios between the underlying historical data repository and another data source
- **Data transformation components** clean, format and consolidate data ingested from the underlying historical data repository
- **Data access components** support data requests from end-users, applications and systems for historical operational data
- **Model deployment components** enables the transmission of analytics models to locations accessible to other end-users, applications or systems

### MODEL EXECUTION

Model execution incorporates components to embed, score and propagate analytics intelligence throughout the factory. The primary objective of model execution is to embed analytics insights from high-quality models in real-time factory operations, and thereby facilitate the provision of timely and continuous knowledge that can positively impact operations. However, given this desired state depends on continuous real-time data, appropriate sources of operating measurements (e.g. instruments for air handling unit) must be identified. Once sources of real-time measurements exist, current technical components facilitating data streams, model execution and analytics integrations should be added to the conceptual lifecycle model. In a similar manner to the model building process, components that are not present, or conflict with Industry

4.0 design principles, should be included in the lifecycle and highlighted for technical implementation. The following points summarise the main types of components related to model execution;

- **Embedded factory components** provide analytics accessibility and visibility to end-users, applications and systems across the targeted factory operations
- **Model execution components** enable scoring and evaluation of real-time measurements streamed from embedded components
- **Analytics output components** control dissemination and notification of data insights to end-users, applications and systems

### 3.2.2 Phase 2: Technical model

This phase aims to transition conceptual components to formal software models that can inform implementation. To achieve this transition, high-level roles, components and connections in the lifecycle can be used to identify candidate use cases and classes. Of course, these use cases and classes shall need refinement and pruning to ensure sufficient implementation details are captured, while limiting redundancy in the final software model. The final software model should include consolidated use cases for the cyber-physical system, with static (i.e. structural) and dynamic (i.e. runtime) perspectives of each proposed use case.Figure 36 further illustrates this software design and modelling process. The process begins with information extraction (e.g. use cases) from the conceptual lifecycle model, before this information is used to construct the static and dynamic software perspectives, which encompass structural, algorithmic and runtime perspectives that inform technical implementation.

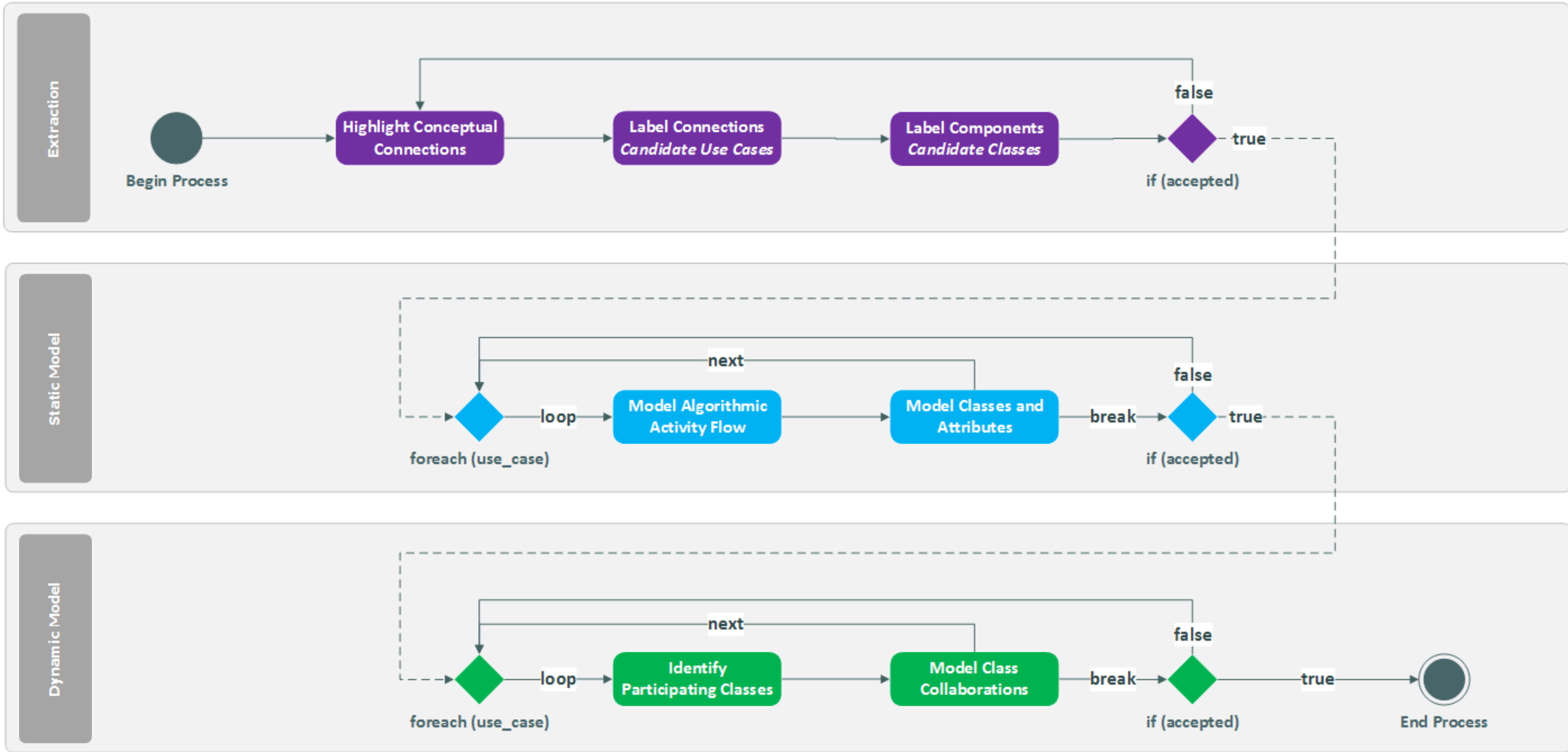



**Figure 36 Software design and modelling process**

### EXTRACTION

Given elicitation and collaboration was undertaken previously, fundamental requirements can be extracted from the conceptual lifecycle model, which include use cases and classes to generate static models to specify objects and attributes for the proposed system implementation, and dynamic models to illustrate collaborations between these objects. The use cases and classes can be identified from the lifecycle model where connections (i.e. data flows) between components are evident. These connections indicate primary system use cases relating to data production, processing and transmission, to name a few. Similarly, components residing on connection endpoints should indicate system classes. However, given the high-level nature of conceptual models, candidate use cases and classes may need further evaluation and refinement before inclusion in static or dynamic models. Therefore, some acceptance criteria (e.g. technical meetings, design principles etc.) should be established to ensure the authenticity and accuracy of the candidate specifications. Where acceptance is not achieved, knowledge of previous shortcomings may be incorporated in future iterations of the extraction process to realise a positive outcome. Indeed, negative outcomes throughout the modelling process should be considered a trigger to iterate and refine.

### STATIC MODEL

Static modelling further develops candidate use cases and classes previously identified during the extraction, and encodes these details using the Unified Modelling Language (UML). UML was chosen to encode static models due to its prominence in the literature and application within industry. The refinement of use cases focuses on describing algorithmic flows (i.e. step-by-step instructions describing the scenario) to avoid ambiguities by encoding precise and interpretable instructions using symbolic notation (i.e. UML activity diagrams), while the refinement of classes focuses on defining attributes and properties using similar symbolic methods (i.e. UML class diagram). Similar to the extraction process, acceptance criteria should be established to encourage reflection and refinement of the static model.

### DYNAMIC MODEL

Dynamic modelling provides another perspective of the proposed software system. Although activity diagrams from the static model provide details regarding functions and operations for each use case, they do not specify the classes responsible for invoking those actions, or how classes may collaborate to fulfil use cases. These dynamic perspectives can be realised using the symbolic notation of UML sequence diagrams, which depict runtime collaborations between class objects during their execution lifetime. Initially, the static model is used to guide the creation of the dynamic model. Previously developed activity flows (i.e. symbolic use cases) are synthesised to identify classes that collaborate to fulfil the described scenario (e.g. upload data from factory). These classes are placed in the order they appear in the activity diagram, which should be indicative of their place in the collaborative sequence. Similarly, the order of actions and decisions from activity diagrams are also used to identify other potential collaborations between classes. This iterative process may continue until such time the dynamic model contains sufficient detail for technical implementation. Given the measure of sufficient detail can be considered highly subjective, acceptance criteria should be established to encourage objectivity and control against over-engineering.

### 3.2.3 Phase 3: Implementation

This phase aims to supplement the technology agnostic software models with more concrete implementation details. These technical and design details are needed to guide real-world development and deployment, which should encompass the hardware, software and service requirements for each use case.

#### System implementation

The overall system architecture can be derived by iterating use cases, and specifying artefacts (e.g. hardware) needed to achieve the specified behaviour. First, the need for infrastructure components are evaluated. Possible components considered may include computers, servers, cloud services, gateways, routers, networks, mobile devices, and internet-of-things sensors, to name a few. The identified infrastructure components can then be commissioned using the appropriate channels within the organisation. Second, messaging requirements to support distributed communication are assessed. While infrastructure components provide the platform and compute resources upon which to build communications, messaging services are needed to facilitate programmatic distributed communications (e.g. message queues). Finally, components and classes defined for use cases should be evaluated to determine if similar functionality already exists within the organisation. However, given the contemporary nature of industrial cyber-physical systems, components relating to emerging technologies and practices shall inevitable need some initial development (e.g. machine learning model deployment). However, technical development should decrease naturally overtime as more use cases are implemented.

### ACCEPTANCE EVALUATION

Once use cases have been technically implemented, the focus should be placed on deploying an increment of the system for theoretical, stakeholder and technical acceptance evaluations, which server as quality control mechanisms for the process. In the context of industrial cyber-physical system for Industry 4.0, these theoretical evaluations and assessments include (a) Industry 4.0 design principles (Table 19) to ensure sufficient alignment with emerging guidelines and recommendations for smart manufacturing technology, (b) stakeholder concerns (Table 20) pertaining to commercial sensitivity, privacy and security (e.g. where and how is data stored), and (c) functional performance (Table 21) relating to scalability, reliability and resilience (e.g. number of sensors processed, performed correct calculation). In the event of assessment results not yielding sufficiently positive outcomes, the system architecture can be refined with this knowledge to improve the proposed technical implementation, before triggering another iteration of assessments to validate improvements.

| **Principle** | **Description** |
| --- | --- |
| Interoperability | Technologies and systems should be capable of exchanging communications using open and consistent standards, which supports the development of collective system intelligence. |
| Virtualisation | Capability to virtualise factory-level components, machinery and processes should be developed to enable complex permutations to be evaluated using simulation and decision-support. |
| Decentralisation | Emerging network architectures are highly heterogeneous (e.g. mobile, laptops, augmented reality etc.) and distributed, which industrial cyber-physical systems must manage with robust, reliable and responsive designs. |
| Real-time | Many industrial operations (e.g. process control) depend on real-time sensing and actuation, and therefore, real-time constraints and boundaries shall exist for some cyber-physical applications. |

**Table 19 Industry 4.0 design principles**

The facility stakeholder assessment addresses internal concerns regarding Industry 4.0 and cyber-physical system adoption. Although the unified design methodology prescribes fundamental stakeholder concerns, these should be adjusted and extended to capture the technology and operating concerns of each facility, with the intention of building stakeholder confidence in the proposed technical implementation.

| Concern | Description |
|---|---|
| Data security | Concerns regarding the transmission and execution of operational data beyond the factory's corporate and automation networks |
| Legacy integration | Concerns regarding compatibility with legacy technologies and systems that received significant financial and time investments |
| Regulation | Concerns regarding compliance with current quality assurance and regulation policies |
| Performance | Concerns regarding both operating (e.g. accuracy, real-time capability etc.) and end-user performance (i.e. reduced productivity) |
| Disruption | Concerns regarding operational disruptions (e.g. downtime) resulting from implementing emerging and untested technologies |
| Knowledge | Concerns regarding inadequate technical and engineering knowledge to rollout emerging technologies |

**Table 20 Fundamental stakeholder concerns**

Finally, functional assessments verify the system operates consistently, predictably and accurately. In some instances, these assessments can be automated using predefined test cases and scenarios, consisting of input data and expected outcomes, while other assessments may require some manual evaluation. Where automated assessments can be used, outputs generated by the system (e.g. file processed) can be compared with the expected outcome from predefined test cases to verify technical and logical integrity.

| Dimension | Description |
|---|---|
| Data ingestion | Functions pertaining to data integration across information systems, programmable logic controllers, and emerging devices |
| Data processing | Functions pertaining to harmonising and contextualising operations data using cleaning and transformation routines |
| Model execution | Functions supporting model deployment and execution within an industrial analytics workflow |
| Real-time scoring | Functions pertaining to the real-time execution and scoring of predictive analytics models in the factory (i.e. embedded analytics) |
| Model accuracy | Functions pertaining to the assessment of an analytics model accuracy and validity |
| Decision outputs | Functions pertaining to the propagation of analytics outputs to third party components, systems and nodes |

**Table 21 Functional assessment dimensions**

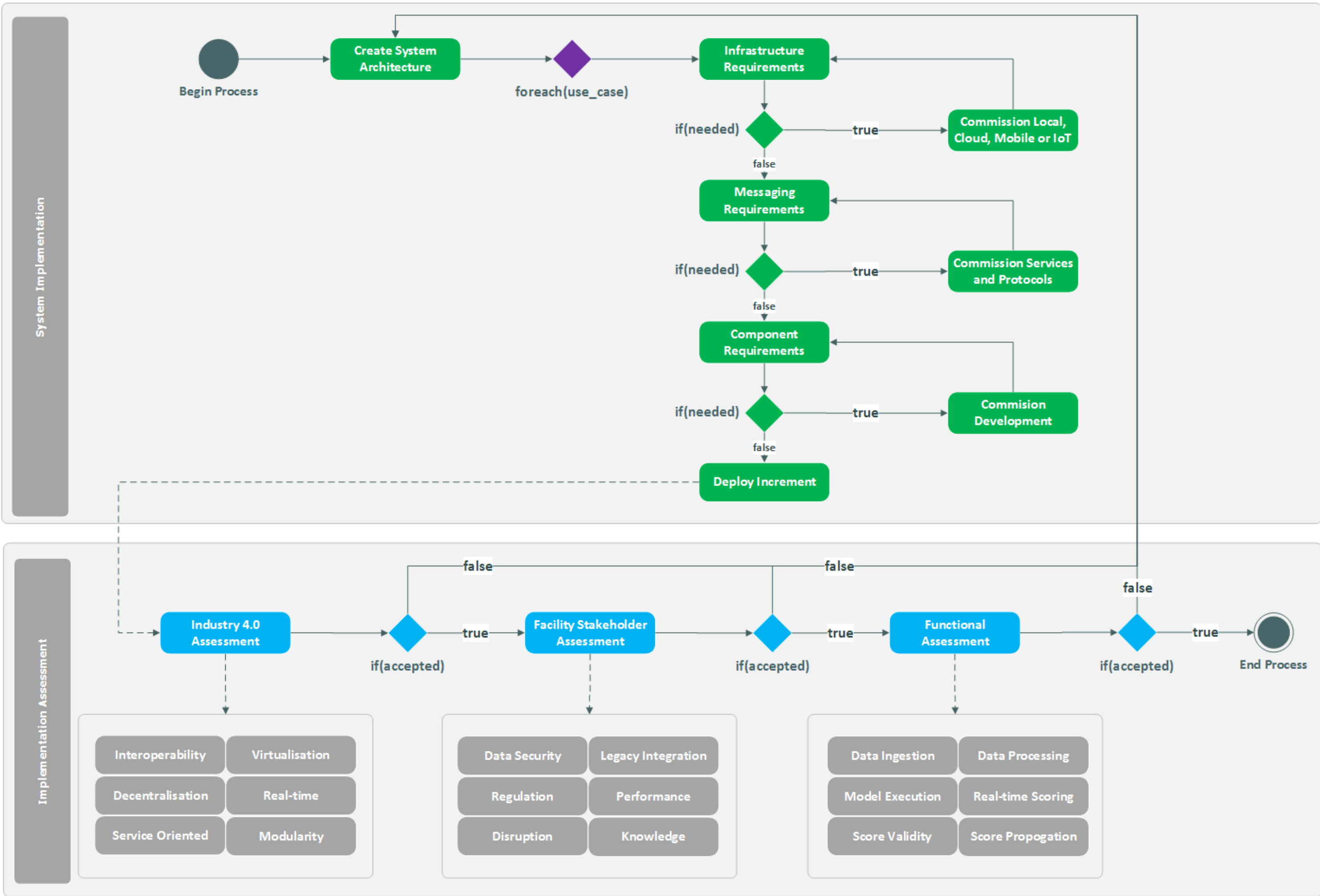



**Figure 37 Technical implementation process**

### 3.2.4 Phase 4: Performance measurement

The phase aims to estimate the worst-case execution performance of the implemented industrial cyber-physical system, which essentially determines the systems suitability for time-dependent Industry 4.0 engineering applications, while also being of importance to Industry 4.0 design principles (i.e. real-time capability) and stakeholder concerns (i.e. performance). In addition to these primary concerns, performance measurements may also be used to benchmark and compare different system implementations throughout the organisation, or highlight positive performance changes due to system modifications.

Generally, performance analysis approaches can be classified as (1) pre-implementation and (2) post-implementation. Pre-implementation approaches build physical models to simulate execution of the proposed system, which enables performance and impediments to be determined in lieu of technical implementation. However, this approach depends on detailed modelling of the engineering scenario (e.g. conditions and branches), and enabling technology platforms (e.g. hardware, software and networking). If details used to build these models are inaccurate, the simulated performance may not be indicative of real-world performance observed post-implementation. In contrast, post-implementation analysis collects data by observing system execution to estimate best and worst-case performance. Although such approaches do not provide upfront performance guarantees, incorporating real-world data ensures performance results are representative of the target environment. Regardless of the performance method chosen, the basic idea of performance assessment is to identify constraints and bottlenecks that can be resolved through incremental improvement.

Given the generic nature of the unified design methodology (i.e. not coupled to single industrial application), and inherent complexities associated with modelling the diversity of industrial technologies, the design methodology can only describe post-implementation analysis to measure performance. Figure 38 illustrates the main stages of this process. Initially, planning aims to establish the primary experiment parameters (e.g. timeframe). Once these parameters have been clearly defined, technical components (e.g. agent to log communication latency) that are needed to measure execution are deployed and configured. Finally, the system operations being measured are continuously executed and persisted in a log file, which informs the development of analytics and/or simulation to estimate system performance.

### Planning

Although many environmental and technical variables could potentially impact system performance, three fundamental variables should be identified during the planning stage. First, the location of the proposed embedded analytics application should be identified so that performance measurements can be taken from the target location. While this may not always be possible due to internal governance policies, location-specific testing can be useful due to potential differences in bandwidth and network constraints throughout the factory. In a similar manner, hardware upon which embedded applications are deployed may also impact performance, and therefore, performance analysis should aim to replicate or utilise the target technology environment. Second, communication endpoints (e.g. cloud server, internet-of-things gateways etc.) responsible for relaying analytics scores (i.e. results) to factory-embedded applications should be identified. Given communication latency increases based on endpoint proximity and traffic routing, performance should be measured using the same type of endpoints that are exposed by the implemented system. Finally, the specific industrial analytics engineering application (e.g. predictive maintenance for air handling units) should be identified. Given model complexity (e.g. inputs, outputs, linear/non-linear etc.) may impact execution time, performance measurements must utilise the same industrial analytics model.

### Setup

After experimental variables have been established (i.e. planning), the setup focuses on deploying the technical components and configurations needed to collect performance data. Initially, software agents for logging communication requests are deployed to the target location. These agents are responsible for continuously monitoring and logging execution times, from acquiring real-time data from factory data sources (e.g. controllers), to executing and scoring industrial analytics models. Ideally, these agents can obtain real-time data streams, and utilise this real-time data as input for the analytics model. However, static input data can also be used when solely concerned with cyber-physical interactions (i.e. excluding the performance of data acquisition from controllers). Regardless of the data source employed, logging agents should continuously acquire and score real-time data at set intervals (e.g. 1-minute frequency), until the agent's runtime exceeds the experiment timeframe (e.g. 1-hour), or completes a particular number of iterations (e.g. 50,000 requests sent). Although these parameters should align with time-constraints and characteristics of the industrial application being considered (e.g. building control may have greater time windows than process control), they may also be used to observe system operation under different levels of stress (e.g. 50 or 100 concurrent connections).

#### PERFORMANCE ANALYSIS

Estimating the execution performance of the system consists of two parts. The first part of the process describes the procedure for classifying, observing and logging communication and execution times (e.g. data acquisition, analytics execution etc.), while the second part of the process may construct physical models to simulate operating boundaries, or present data analysis using the execution logs to estimate worst-case execution time, and summarise general system performance (e.g. reliability, consistency etc.). During the first part of the process, a single loop encapsulates the logging sequence (i.e. *while(measuring)*), and continues to execute until the agent's runtime exceeds the experiment timeframe (i.e. *measurement* == *false*). This timeframe is evaluated before initiating each new loop. At the beginning of each loop, the agent's background timer is reset to ensure the previous logs are cleared, before logging execution times for data acquisition, message transmissions, and analytics scoring. In the event of the observed worst-case execution time exceeding the timing constraints of the industrial engineering application, the technical components that are negatively impacting performance should be identified and modified before undertaking additional performance analysis.

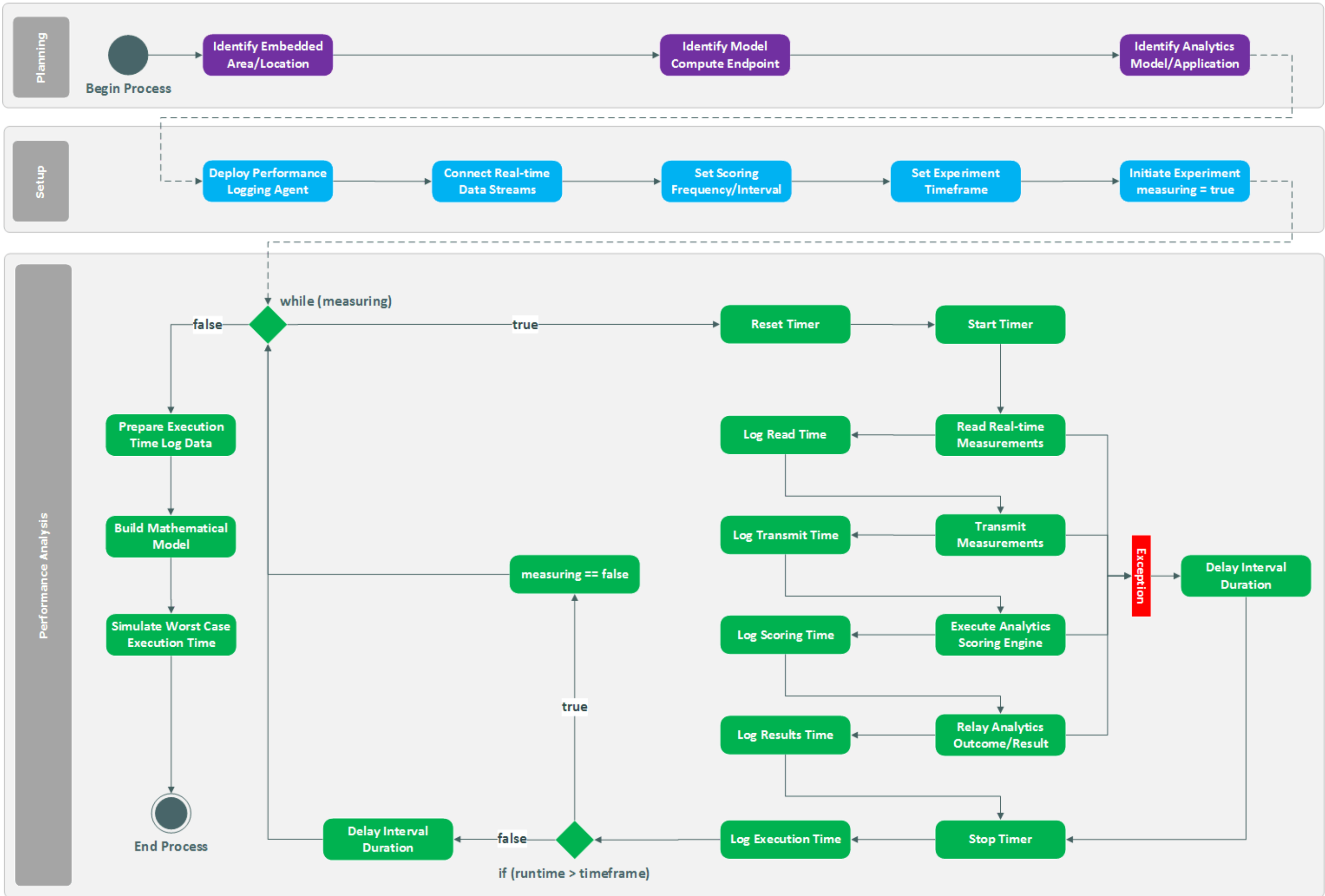



**Figure 38 Post implementation performance measurement process**

## 3.3 Chapter conclusions

Given industrial cyber-physical systems for Industry 4.0 require new theories and methods to inform implementation, this chapter presents a multidisciplinary unified design methodology that integrates and connects prominent modelling approaches (i.e. conceptual, software and mathematical) evident in the literature, which are needed to address specific design concerns for industrial cyber-physical systems (e.g. domain understanding, performance etc.). In particular, the methodology emphasises and highlights criteria pertaining to Industry 4.0 design principles, common stakeholder concerns, and technical functionality. Of course, while the methodology aims to deliver structure and formalism to multidisciplinary design processes for industrial cyber-physical systems, the methodology cannot be overly prescriptive due to the unique combinations of people, technologies and applications that may exist. Thus, engineers, designers and developers using methodology may need to interpret, customise and extend parts of the proposed processes to meet their own requirements.

# Chapter 4
## Implementation & Demonstration

# 4.1 Chapter introduction

This chapter presents the application of the unified design methodology during the development of an energy-oriented industrial cyber-physical system, which was undertaken in a large-scale manufacturing facility located in Cork, Ireland. The following summary outlines the main sections of the chapter;

- **Phase 1 lifecycle modelling (section 4.2)** describes the development of a conceptual model based on the industrial partners current operations, which was used to develop an understanding amongst all stakeholders.
- **Phase 2 software modelling (section 4.3)** demonstrates the extraction of primary use cases from the conceptual model, and the construction of software and system models to define technical specifications.
- **Phase 3 technical development (section 4.4)** presents the technical implementation and evaluation of a fog computing architecture supporting real-time and embedded cyber-physical interactions in the factory, and a cloud computing architecture facilitating the development of analytics models.
- **Phase 4 performance measurement (section 4.5)** outlines the setup, configuration and data collection procedures used to stress test the cyber-physical system implementation, and compare the performance of different cyber-physical interfaces (i.e. fog versus cloud computing).

# 4.2 Phase 1: Lifecycle modelling

## 4.2.1 Fundamentals

This research was undertaken with an industrial partner in the form of DePuy Johnson & Johnson. The collaborator was engaged at different stages to (a) interact with operational teams across automation, operation technology, information technology, big data, and data analytics, and (b) evaluate the operation of an industrial cyber-physical system in a large-scale manufacturing facility. However, given significant governance policies and procedures surrounding production systems, the technical design, development and performance aspects of this research were applied to existing auxiliary industrial energy systems. In particular, the real-time and continuous monitoring of industrial AHU's that supply the production environments.

Industry 4.0 adoption is an important aspect of DePuy Johnson & Johnson's strategic roadmap. Of the operational teams engaged, many were focused on developing smart data-driven methods to improve operations. These methods typically differed from team-to-team, resulting in many disparate architectures, technologies, and tools being deployed in the facility. Such approaches may suffer from duplicated effort, increased licensing costs, and poor technology cohesion. High-level improvements could potentially be achieved across teams by formalising methodologies, reducing dependencies on proprietary technology, and promoting greater standards adoption.

Figure 39 illustrates the primary roles and responsibilities identified for the conceptual lifecycle model. These roles and responsibilities were defined through engagement with industry, and deliberations with fellow researchers. The four identified roles were Operation Technology, Information Technology, Data Analytics and Embedded Analytics. Of these roles, Operation Technology and Information Technology support data integration and management, while Embedded Analytics and Data Analytics facilitate analytics model building and deployment. Furthermore, each role should identify and adopt open standards and technologies whenever possible.

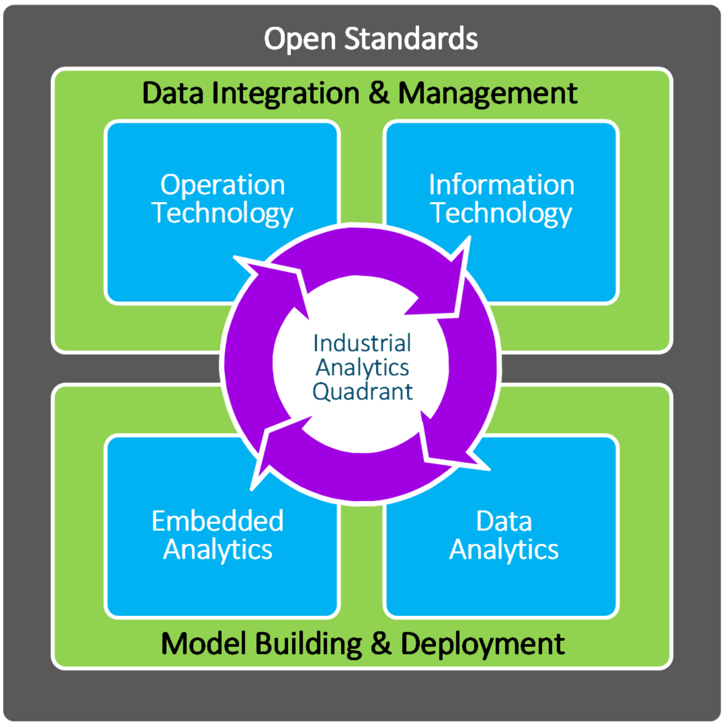



**Figure 39 Operational boundaries identified in DePuy Johnson & Johnson**

The following section describes the interdisciplinary roles, responsibilities and operating boundaries identified in DePuy Johnson & Johnson.

- **Operational Technology** manages industrial information and automation systems, which are responsible for monitoring, persisting, archiving, and integrating operation data across the factory. They typically collaborate with Information Technology to realise factory-to-cloud integration, and facilitate the adoption of contemporary technologies synonymous with Industry 4.0 (e.g. internet-of-things, big data, machine learning etc.).

- **Information Technology** fulfil a wide-range of roles across the enterprise including the provisioning of compute resources, information system development, business intelligence, data management and integration. They ensure Data Analytics can access analytics-ready data by providing information system infrastructures and architectures that transmit and clean operation data from heterogeneous data sources in the factory.

- **Data Analytics** employ data-driven methods and tools (e.g. machine learning) to derive insights that may positively affect operations. They must engage with Embedded Analytics to (a) identify and comprehend operating-specific analytical questions, and (b) evaluate models that attempt to answer these questions.

- **Embedded Analytics** operationalise applications, tools and models in the factory to affect real-time decision-making. They utilise subject matter expertise of specific operations to guide Data Analytics efforts, while ensuring Operation Technology provide the appropriate real-time measurements to factory embedded models.

### 4.2.2 Model building

An on-premises Cylon Building Management System (BMS) was identified as the main source of AHU operational data, and was therefore chosen as the primary industrial information system needed for building machine learning models. The BMS was deployed on a dedicated computer (i.e. BMS-PC) to display real-time measurements, implement setpoint alarms, and archive operational data at 5AM each day. The archiving process collected operational data for the previous 24-hours from PLC's, and appended each sensor's CSV log file with the newly acquired measurements. The BMS archive consumed 1.06 GB of disk space, and consisted of 838 log files, with each storing historical measurements for a single sensor. The timespan of measurements in each log file ranged from a couple of months to 4 years. The largest file in the archive measured 13 MB, while the smallest file measured a mere 1 KB. Approximately 5% of files were larger than 5 MB, which could loosely be associated with log files containing 4 years of data, while files smaller than 1 MB were somewhat indicative of (a) new sensors that recently began archiving, or (b) legacy sensors where archiving was disabled. However, file size cannot be used reliably to predict date ranges. For example, two log files measuring 5MB and 12MB may contain the same date ranges and measurement resolutions, but the latter may contain higher precision readings, which simply consumes more bytes on disk.

| Requirement | Summary | Currently Exists |
|---|---|---|
| Data archiving | Current archiving capabilities are provided by a software module referred to as CC Reports. This module collects measurements from PLC's every 24-hours and persists data for each sensor in CSV files. | Yes |
| Data ingestion | There are currently no components for ingesting or integrating BMS data with third-party systems. However, building analytical models depends on being able to ingest operational data. | No |
| Data transformation | Given there are many anomalies (e.g. missing values) may be observed in operational data, components to transform and clean data are necessary as a precursor to analytics and model building activities. | No |
| Data access | For engineering, statistical and operational personnel to building analytics models, BMS data must be easily accessible using a standard and common interface. | No. |

**Table 22 Primary technical concerns for model building**

Figure 40 illustrates the conceptual model building stream that connects the BMS to the cloud (i.e. factory-to-cloud connectivity), produces analytics-ready operational data for inspection, and provides an open data interface to enable data exploration, model building and deployment. The illustration utilises the previously identified operational roles to define theoretical data processing stages, which serve to partition technical components and systems, and highlight the interconnection between different operational units. DePuy Johnson & Johnson reported the use of ISA-95 to guide information system integration in the factory. Based on discussions and interactions with those implementing the ISA-95 standard, Levels 0 to 3 of the ISA-95 architecture were allocated to Operation Technology. These levels represented (a) factory environment at Level 0, (b) sensors and instrumentation at Level 1, (c) automation and control at Level 2, and (d) industrial information systems at Level 3 (e.g. Cylon BMS and associated data repository). Although multiple levels of the ISA-95 architecture are depicted, the model building process is mainly concerned with the manifestation of operational data in Level 3 – with an ingestion engine embedded in the factory to stream operational data to the Information Technology stage. Once received from the factory, the Information Technology stage focuses on delivering analytics-ready data, and ensuring analytics activities are simplified by promoting data accessibility for those associated with the Data Analytics stage. The following sections provide a more in-depth description of each of these stages.

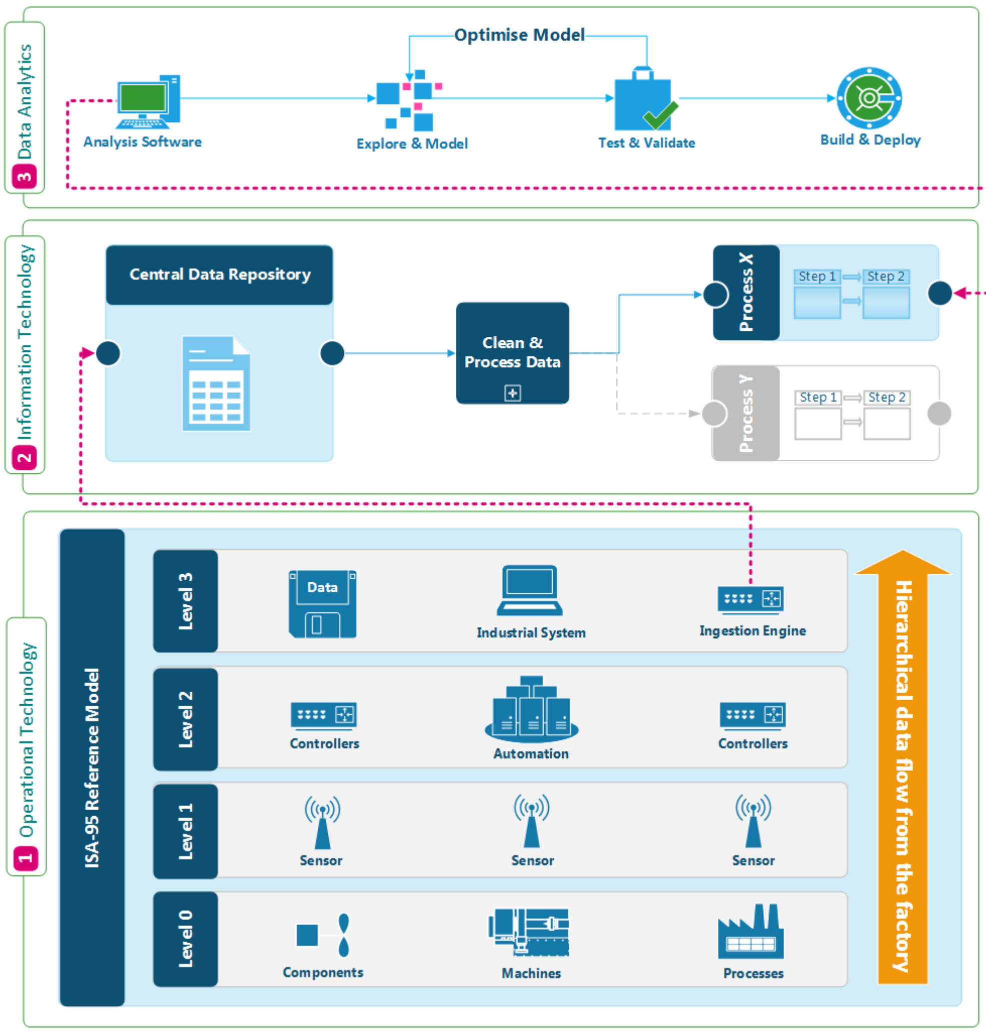



**Figure 40 Conceptual model building stream**

### OPERATION TECHNOLOGY

Components in the Operational Technology stage are used to archive operational data and establish communication with Information Technology. The reference architecture illustrates an example of operational data being transmitted from the factory to an industrial information system (BMS), which periodically executes an archival process that persists data in a log file. These logs are stored on disk and accessed by an ingestion engine component before being transmitted to a global data repository. Although the specific systems and storage formats may change from scenario-to-scenario, the ISA-95 architecture should ensure processing and data flows remain the same.

Given analytical questions emanating from the factory cannot be answered without having access to high-quality historical data, Operating Technology components are fundamental to data exploration and model building. Of the components depicted in the model building data stream, industrial information systems and associated archives are commonplace in modern manufacturing facilities. Where these components do not exist, the proposed lifecycle dictates that Operation Technology are responsible for implementing the necessary archiving solutions. In contrast to the common availability of industrial information systems, data ingestion components and communication endpoints are less likely to exist due to the pronounced separation between Operation and Information Technology. However, such components are needed to encourage greater convergences between these stages. Therefore, Operation and Information Technology must collaborate to agree specifications and protocols to support ingestion and integration, with Information Technology being primarily responsible for implementation and deployment.

### INFORMATION TECHNOLOGY

Components in the Information Technology stage are used to store, process and prepare operational data transmitted from Operation Technology. The collaborative effect of these components automates the production of analytics-ready data. The model building stream illustrates interactions between these components using the example of AHU data from the BMS being prepared for analysis. Initially, AHU data transmitted from Operation Technology is tagged and stored in the data lake. This transmission triggers a call to the workflow engine, which checks if any active workflows are associated with AHU's. Given the existence of an AHU workflow, the workflow engine constructs and executes a data processing job. These jobs consist of multiple processing modules, with each performing a single operation on the AHU data (e.g. sort by timestamp, remove duplicates etc.) to produce the final data set.

The Information Technology stage exposes communication endpoints connected to Operation Technology and Data Analytics. As previously mentioned, the inbound endpoint for Operation Technology facilitates the transmission of data from the factory, while a second inbound endpoint from Data Analytics enables those undertaking analysis to access the (a) final output (i.e. clean analytics-ready data) from a particular workflow (e.g. AHU workflow), or (b) intermediary output from a particular workflow stage. To illustrate the usefulness of this concept, consider a scenario where the final output from a workflow aggregates a time-series using daily averages, but analytical questions later arise that require access to 15-minute datasets. By following the multistage approach, it should be possible to access 15-minute data from an earlier stage of the workflow, before daily average was applied. Thus, reducing duplicated processing effort, decrease maintenance, and promote reusability.

Information Technology components depicted in the model building stream may exist in facilities where cloud-based big data infrastructures for business intelligence and analytics have been adopted. Where these components exist in a business enterprise context, they may require some configuration and customisation to work with typical operational data (e.g. time-series). In facilities where these components and capabilities do not exist, Information Technology personnel should be primarily responsible for their design, development and implementation. These components abstract Data Analytics personnel from time-consuming and complex processing of ad hoc and proprietary operational data, while also being fundamentally important to the scalability and resilience offered by the eventual technical implementation.

The workflow processes responsible for data preparation comprise multiple data processing modules, which are positioned in a particular order to produce a clean analytics-ready dataset (i.e. the order depends on the transformations needed for the inbound data). Analytics-ready datasets are similar to tidy datasets, where each column refers to a single variable/feature/measurement, and each row refers to a single observation at a point in time. Each processing module in a workflow exposes inputs and outputs, which enables chains of processing modules to be interconnected (i.e. the output from one providing the input for the other).

**DATA ANALYTICS**

Components in the Data Analytics stage use data-driven methods to derive insights that can positively impact operations. To facilitate these particular objectives, Data Analytics personnel must acquire analytics-ready data, build insightful data-driven models, and prepare these models for factory deployment. The Data Analytics stage exposes communication endpoints connecting to Information Technology and Embedded Analytics. The outbound endpoint to Information Technology facilitates the acquisition of analytics-ready data from workflows, while the outbound endpoint to Embedded Analytics (i.e. after the model building process) enables the deployment of models to real-time data streams in the factory, which shall be described later in the model execution section. The inbound endpoint from Information Technology may trigger actions on existing models (e.g. re-training) when new training data becomes available from Information Technology workflows. Data Analytics components should demonstrate turnkey data analysis (i.e. no pre-processing), model building, model standardisation and deployment capabilities. Thereby automating low-value and manual activities, which can afford more time to high-value activities.

Data Analytics components may exist in facilities that currently utilise business and data analysis. Examples of common components include those relating to statistical software tools (e.g. R and SAS). However, less components may include those supporting the automation of model building, training and deployment. Where such components do not exist, Data Analytics and Information Technology personnel must collaborate to define specifications, with Information Technology leading implementation.

### 4.2.3 Model execution

Component-level sensors in AHU's may be used to predict system health and energy inefficiencies. Examples of such measurements include mechanical component positions, temperature, and airflow. These measurements in DePuy Johnson & Johnson are transmitted in real-time across automation and control networks, which can be read using an OPC server installed on the BMS-PC. Table 23 outlines the real-time measurements identified for the AHU used in this research (AHU9).

| Section | Measurement | Section | Measurement |
|---|---|---|---|
| Return air | Temperature [°C] | Heating coil | Off coil air [°C] |
| Return air | Humidity [%] | Cooling coil | Supply water [°C] |
| Return air | Enthalpy [kJ/kg] | Cooling coil | Return water [°C] |
| Return air | CO2 [ppm] | Cooling coil | Valve position.[%op] |
| Return air | VSD [%] | Cooling coil | Off coil air [°C] |
| Return air | Flow [m3/s] | Humidification | Temperature [°C] |
| Return air | Motor power [kW] | Humidification | Humidity [%] |
| Return air | Damp. Position [%op] | Humidification | Dew-point [°C] |
| Exhaust air | Damp. Position [%op] | Humidification | Status [1/0] |
| Outside air | Temperature [°C] | Humidification | Valve position. [%op] |
| Outside air | Temperature 2[°C] | Reheat coil | Supply water [°C] |
| Outside air | Humidity [%] | Reheat coil | Return water [°C] |
| Outside air | Humidity 2[%] | Reheat coil | Valve position. [%op] |
| Outside air | Enthalpy [kJ/kg] | Reheat coil | Off coil air temperature[°C] |
| Outside air | Enthalpy 2 [kJ/kg] | Supply air | Temperature [°C] |
| Outside air | Damp. Position [%op] | Supply air | Humidity [%] |
| Frost coil | Supply water [°C] | Supply air | Enthalpy [kJ/kg] |
| Frost coil | Return water [°C] | Supply air | CO2 [ppm] |
| Frost coil | Valve position [%op] | Supply air | VSD [%] |
| Frost coil | Off coil air [°C] | Supply air | Flow [m3/s] |
| Mixed air | Temperature [°C] | Supply air | Motor power [kW] |
| Mixed air | Humidity [%] | Supply air | Pressure [Pa] |
| Mixed air | Enthalpy [J/kg] | Zone | Temperature [°C] |
| Heating coil | Supply water [°C] | Zone | Humidity [%] |
| Heating coil | Return water [°C] | Zone | CO2 [ppm] |
| Heating coil | Valve position.[%op] | | |

**Table 23 Real-time measurements identified for AHU9**

Table 24 presents the technical modules specified in the methodology, and summarises their relationship to the real-time AHU measurements previously identified. Gaps in requirements are later addressed using a conceptual model, which depicts how required technical components may interact to embed machine learning in factory operations.

| **Requirement`** | **Summary** | **Currently Exists** |
|---|---|---|
| Embedded analytics | Being able to embed analytics in different locations in the factory provides the necessary intelligence to affect real-time operations. Although some large LCD displays report high-level production and energy consumption to promote awareness, advanced real-time analytics (e.g. machine learning) capabilities are not currently embedded in factory operations. | No |
| Model execution | A technology agnostic process or platform for executing production-ready analytics models enables machine learning models to be executed outside traditional desktop statistical software. Currently, advanced analytics models developed by data analytics teams in DePuy Johnson & Johnson are executed as batch routines within particular software applications (e.g. SAS, R etc.). | No |
| Analytics output | Where real-time analytics models predict a particular event or occurrence, other systems, people or models may need to be notified of the result. This notification and interoperability was not evident in DePuy Johnson & Johnson, but this may be expected given the lack of real-time analytics capabilities. However, such fluid communications would appear unavoidable for those aspiring to achieve self-adaptive Industry 4.0 factories. | No |

**Table 24 Requirements for real-time and continuous model execution**

Figure 41 illustrates the proposed execution stream of the conceptual model, which delivers real-time and embedded analytics to the factory. In essence, the model execution stream depicts the high-level structure of an industrial cyber-physical system for Industry 4.0, with operating data streaming from the physical-world (i.e. factory) to the cyber-world (i.e. cloud or other compute platform) for analysis. Although not restricted to the analytics classifications presented, models focusing on event processing (e.g. issue identified and raised), self-awareness (e.g. equipment's understanding of how issues impact overall health and efficiency), self-compare (e.g. equipment's ability to compare operating performance with other units in the fleet), and self-configure (e.g. equipment's intelligence to adjust operation based on insights) are significantly aligned with the objectives of Industry 4.0 and cyber-physical systems. These analytical insights are returned to the physical-world (i.e. factory) to inform human-in-the-loop and automation decision-making, and invoke triggers that notify other systems of these decisions and actions.

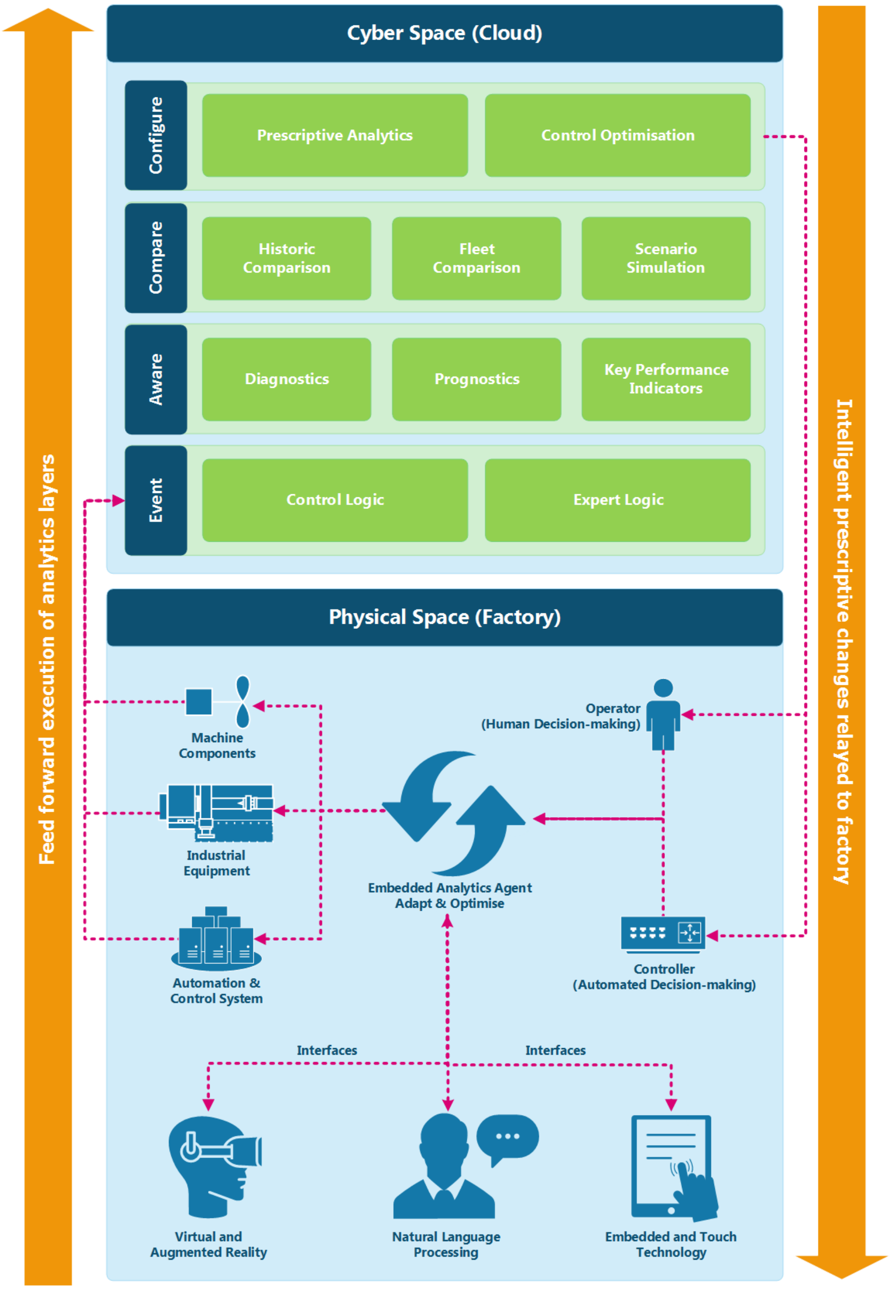



**Figure 41 Conceptual model execution stream**

### Physical Space

The granularity of monitoring and virtualisation (i.e. cyber-twinning) can be considered at different levels, including (a) component (e.g. valve position in AHU), (b) machine (e.g. all measurements comprising an entire AHU) and (c) process (e.g. measurements from multiple systems). A factory embedded agent can read measurements associated with these sources, and continuously stream real-time measurements to the cyber-world for analysis. Similarly, these software-based agents listen for responses from the cyber-world, and notify the appropriate personnel and systems (e.g. CMMS for maintenance requests) of these outcomes.

Given technical components did not exist in DePuy Johnson & Johnson to support the embedding of machine learning models in factory operations, it was necessary to provision the development of an embedded analytics factory agent comprising factory-to-cloud (e.g. HTTP) and OPC client libraries to (a) read real-time measurements using an OPC-UA server, (b) transmit these measurements to the cyber-world, and (c) push notifications broadcast notable results to third-party systems.

#### CYBER SPACE

The cyber-world establishes digital representations of components, machines and processes that reside in the factory. These digital representations are commonly referred to as digital twins, while virtualisation describes the technical process used to create digital twins. Once entities and processes from the physical world are virtualised in the cyber-world, the manifested digital twin can be interrogated and analysed to derive analytics insights. Although many analytics methods may be used to generate useful operational insights, the conceptual model execution stream highlights prominent model classifications based on Industry 4.0 objectives, and methods extracted from the literature. These layered classifications are placed in logical order, with insights from lower layers being useful to the layers above (e.g. equipment should be aware of its own state, before looking to compare its operation to others). In many respects, such prescribed analytics classifications differentiate industrial cyber-physical systems from generic cyber-physical and internet-of-things implementations.

- **Event processing layer** embodies deductive and deterministic logic based on engineering first-principles, which can highlight fundamental issues and assert corrective action. In terms of implementation, facilities may mirror control logic in the cloud to simulate factory operations, or encode specific subject matter expertise using conditional rules and statements.

- **Self-aware layer** considers models constructed using deductive and inductive reasoning, which determine the health or state of digital twins. Typically, models promoting self-awareness for components and equipment utilise diagnostic and prognostic methods to (a) determine current operating health, and (b) estimate remaining useful life.

- **Self-compare layer** promotes secondary and broader analytical reasoning to determine normal operating health and performance, with the intention of providing greater context to insights of self-awareness. Establishing greater context may be achieved by (a) evaluating historic operating behaviours and patterns, (b) comparing current operating patterns with those of digital twins in the same fleet, and (c) simulating different scenarios to highlight more efficient operating patterns.

- **Self-configure layer** comprises prescriptive models that consume analytics insights propagated from the lower layers, and produce corrective actions to improve operational efficiencies. In the context of Industry 4.0 and industrial cyber-physical systems, self-configuration relates to control logic optimisation changes, which are relayed to the factory and automatically pushed to the control network.

Although the model execution and cyber-physical stream depicts many industrial, technical and analytics components interacting, implementation efforts undertaken in DePuy Johnson & Johnson only incorporate those necessary to (a) demonstrate the design methodology, and (b) implement an AHU issue identification scenario (i.e. self-aware component-level model). In terms of existing real-time industrial analytics capabilities, current efforts relating to the cyber-physical systems and digital twins were limited, and mainly consisted of approaches that did not naturally align with Industry 4.0. In particular, there was an overreliance on proprietary technologies, poor levels of system interoperability, and insufficient understanding of how to address real-time and scalability challenges.

### 4.2.4 Conceptual analytics lifecycle

A closed-loop conceptual lifecycle supporting industrial analytics development and deployment can be realised by merging the proposed model building and execution streams. Figure 42 illustrates the relationship and interconnectivity between the respective streams, which provides a basic level of detail to inform software modelling and technical implementation. These basic details aim to strike the balance between technology neutrality, while also trying to highlight important details that may be important to the final specification.

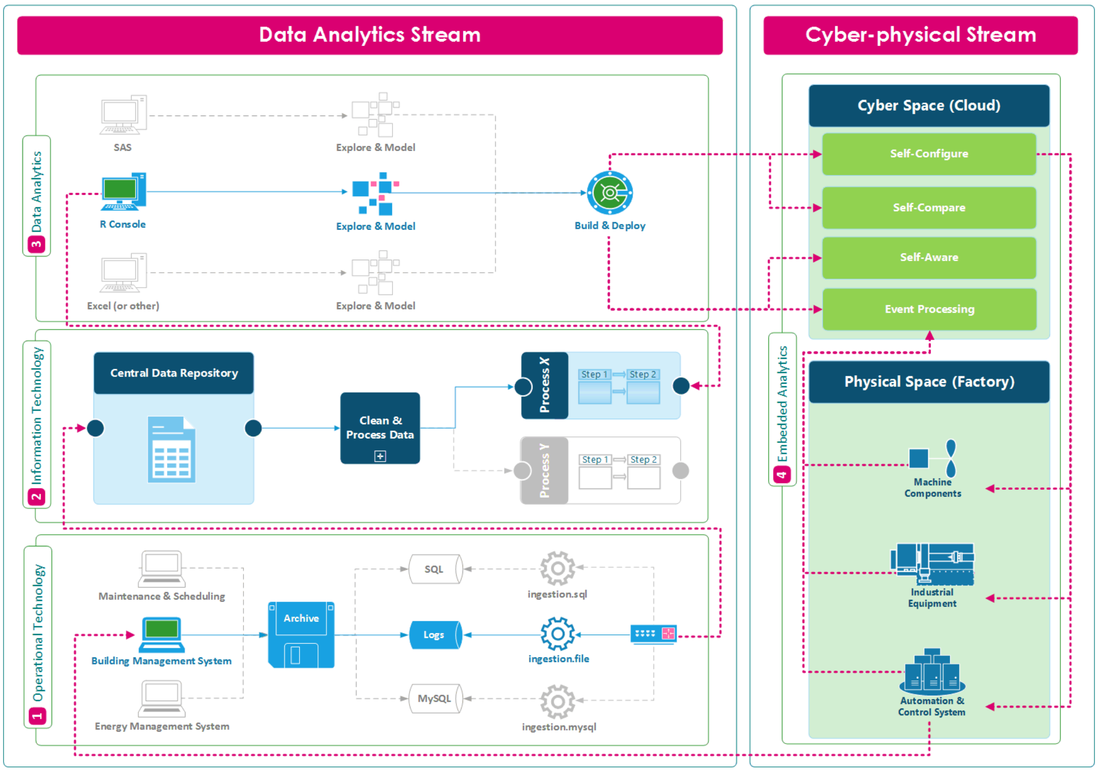



**Figure 42 Relationship between model building and model execution streams**

# 4.3 Phase 2: Software modelling

## 4.3.1 Extraction

To begin moving towards a more formal software or requirements model, easily identifiable use cases were extracted from the conceptual model. Figure 43 presents the use case model developed during the extraction process. The primary actors (i.e. people or systems) are displayed outside the diagram boundary, which for this scenario are the main operating teams from the conceptual model. Although this choice of actor may be considered quite broad, the overlap with the conceptual model provided a strong association between both representations, while specific sub-systems can be introduced during static and dynamic modelling.

The main use cases and system behaviours linking different parts of the conceptual model are represented as labelled ovals within the diagram, and connected to the primary actors (i.e. teams) that initiate these behaviours. Table 25 summarises each use case, while the static and dynamic models presented later in this chapter provide further details relating to the actions and execution of each case.

| Use Case | Description |
|---|---|
| *Log Data* | The data archiving process undertaken by the building management system, and other industrial information systems. |
| *Collect Data* | The data collection process for accessing the building management system's data archives. |
| *Transmit Data* | The data transmission process for sending 24-hour data from the factory, to a scalable and fault tolerant repository (e.g. cloud). |
| *Store Data* | The process for accepting data transmissions from the factory, and persisting the transmitted files. |
| *Clean Data* | The processing workflows responsible for preparing analytics-ready data, which may comprise several cleaning actions. |
| *Expose Data* | The contextual processes that create and manage the naming of directories and files, with the intention of simplifying data access. |
| *Access Data* | The process undertaken by analytics end-users to acquire the data needed to train, build and test models. |
| *Build Model* | The general process of building a production-ready machine learning model. |
| *Deploy Model* | The process of deploying and sharing an open and technology-agnostic machine learning model, with the intention of making the model accessible and executable by other systems. |
| *Stream Data* | The acquisition of real-time measurements from devices or controllers within the factory, which are needed to serve as input to production-ready and executable machine learning models. |
| *Score Model* | The real-time scoring and execution of machine learning models, which are accessible by systems embedded in the factory floor. |
| *Relay Score* | The third-party systems, endpoints and processes that should be notified of model results and predictions. |

**Table 25 Description of use cases**


**Figure 43 Primary use cases extracted from conceptual model**

## 4.3.2 Static model

The static model supplements previous use case descriptions by highlighting prominent actions, and illustrating typical workflows between these actions. Although many alternative scenarios and actions could be considered, these use cases are extremely procedural, and comprise immutable operating data. Therefore, use cases were modelled assuming correct operation. An initial activity diagram is provided below for reference, with the remaining activity diagrams presented in Appendix A.

Figure 44 illustrates the activity and workflow of the log data use case. The process begins with the building management system (or other) initiating the archive data, which typically occurs at the same time each day. During this process the data from the previous 24-hours is read from each controller, before being appended to an appropriately labelled log file (e.g. sensor identifier), and saved to an appropriate network drive or location.

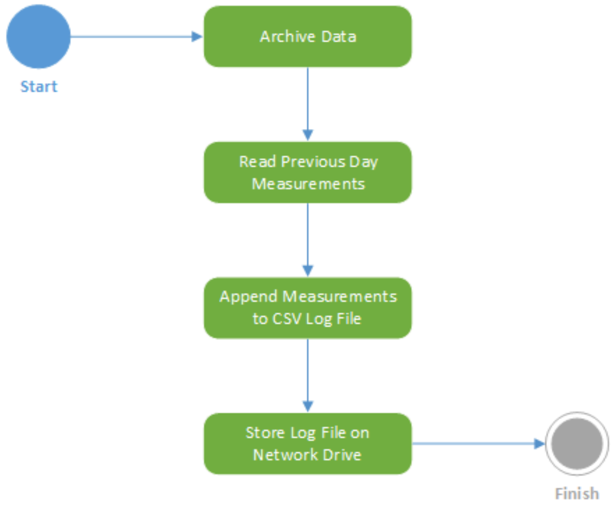



**Figure 44 Activity diagram for log data**

### 4.3.3 Dynamic model

The dynamic model supplements the previous use case and static model with (a) details about how users, components and systems may interact with each other to fulfil the necessary behaviour, and (b) identify more specific actors (e.g. systems) that were naturally abstracted by the use of teams within the use case model. Pseudo-code was used within the sequence diagrams to convey intent, which could inform and guide the subsequent technical implementation. An initial sequence diagram is provided below for reference, with the remaining sequence diagrams presented in Appendix B.

Figure 45 illustrates the sequential actions and interactions for the log data use case. The building management system (Cylon BMS) initiates the data archiving routine every 24-hours, and determines the controllers from which to collect data. A natural looping sequence occurs, whereby the building management system reads the data stored within each controller's memory, appends the retrieved data to the current log file, and saves the amended file to the network.

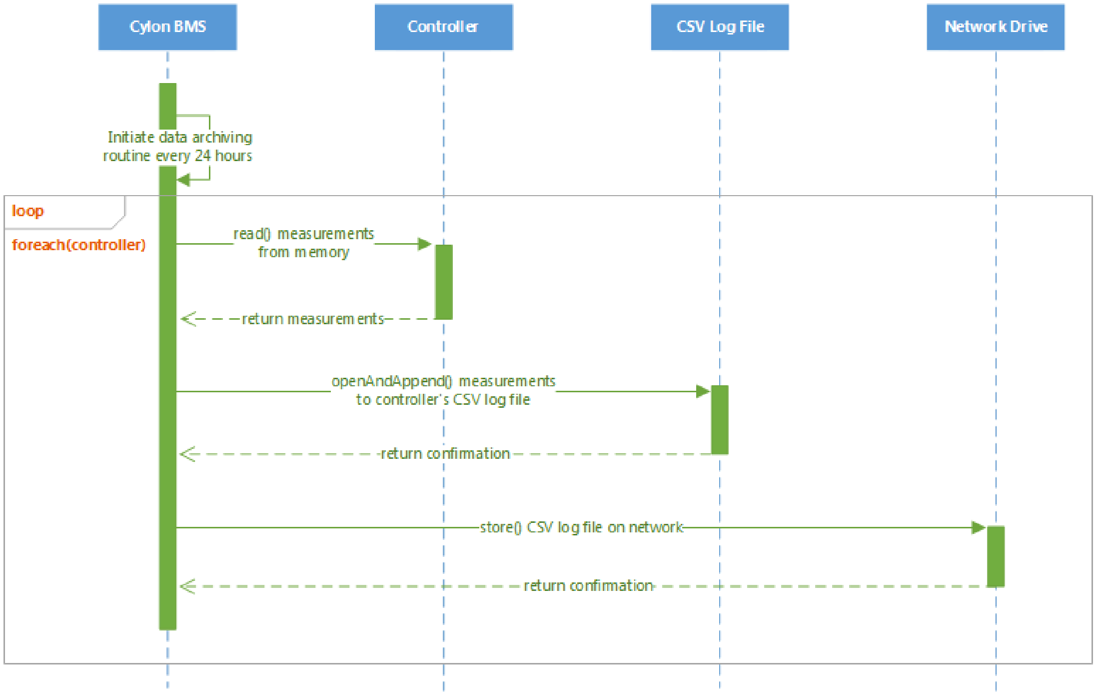



**Figure 45 Sequence diagram for log data**

## 4.4 Phase 3: Technical development

### 4.4.1 System implementation

The technical system implementation utilised multiple architectures. A fog computing architecture was used to implement the industrial cyber-physical system and enable real-time embedded machine learning model execution, while a more common cloud computing architecture was employed to implement batch data integration, processing and exploration. The details, reasons and benefits of these architectural choices are described in the following sections.

#### Fog computing cyber-physical system for model execution

Figure 46 illustrates the composition of the fog architecture for delivering real-time embedded machine learning using cyber-physical interactions. In essence, the cloud platform stores production-ready machine learning models for different engineering application scenarios (e.g. equipment prognostics), which can be disseminated to fog nodes that are capable of delivering local and secure execution. Each fog node's identity and engineering scenarios are registered on the cloud platform to ensure only relevant PMML models are downloaded and synchronised. Once machine learning models exist on the fog (i.e. shadow copy of model), they can be continuously polled using streaming operational data to deliver predictions that inform decision-making (e.g. control changes), without depending on external connectivity or services (e.g. broadband).

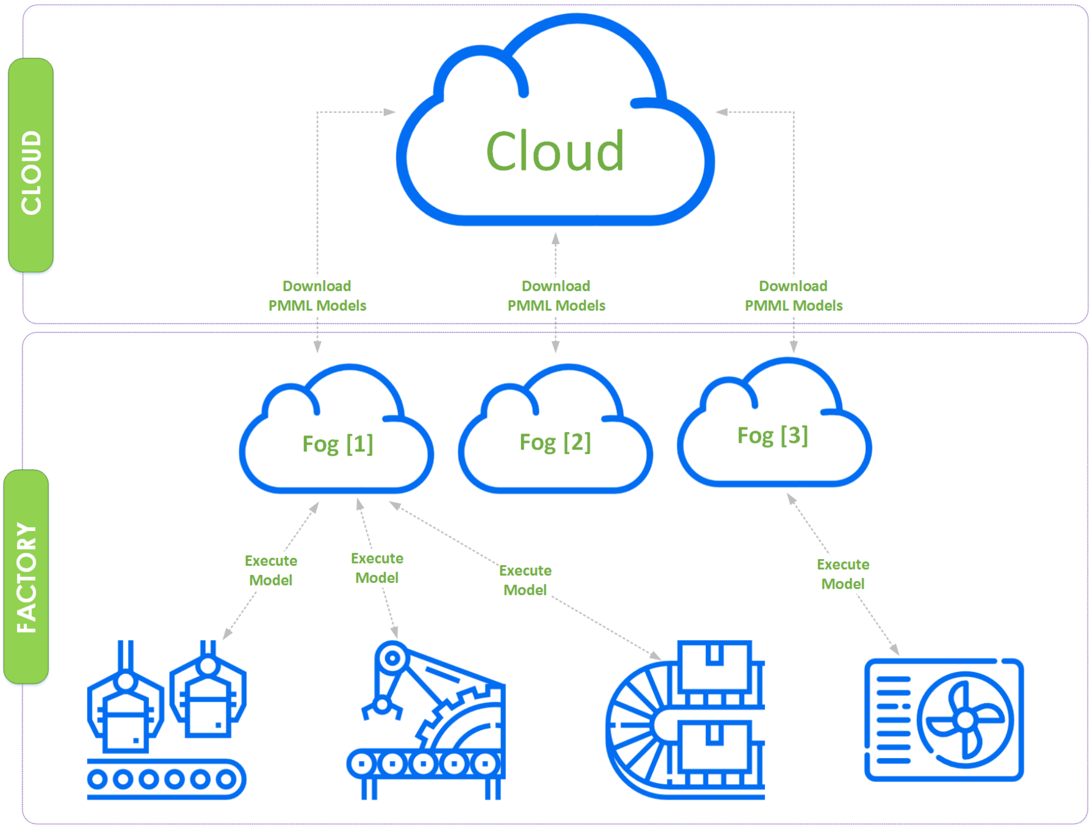



**Figure 46 Composition of fog computing with cyber-physical interactions**

Figure 47 illustrates the primary components and technologies used to implement the industrial cyber-physical system for embedding machine learning models in Industry 4.0 engineering applications.

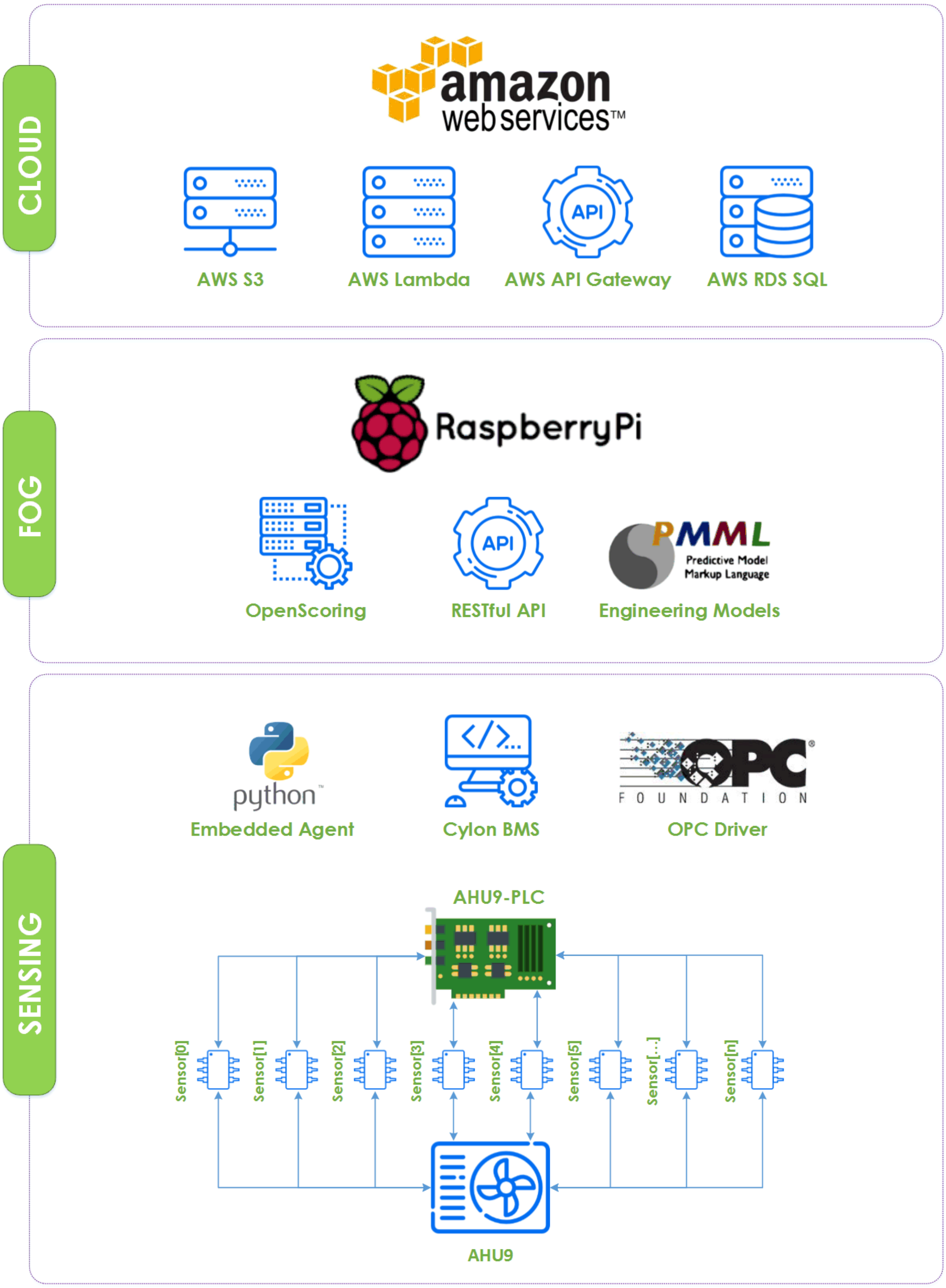



**Figure 47 Fog computing architecture, components and technologies**

### Sensing Layer

The sensing layer contained much of the current equipment and systems needed to continuously acquire real-time measurements, with additional components developed to mediate communications between physical and cyber environments. A summary of sensing components and operations are described below;

- **AHU9** represents the overall equipment being monitored for this demonstration, which consisted of several instruments measuring temperatures and mechanical positions (e.g. valves and dampers).

- **PLC-AHU9** controls the operation of AHU9, using the real-time measurements of the unit's instrumentation to determine appropriate heating and cooling operations. In turn, the cyber-physical system architecture leverages the PLC's existing real-time stream to connect the physical and cyber worlds.

- **Cylon BMS** monitors energy and environmental conditions in the factory. The information system continuously reads measurements from PLC's, displays these measurements on-screen, raises alarms when necessary, and archives measurements every 24-hours.

- **OPC-UA** server exposes a consistent and standard interface for programmatically acquiring real-time measurements from PLC-AHU9, which could be engaged using OPC client libraries or manually constructed SOAP web service requests.

- **Embedded software agents** mediate communications between the physical and cyber worlds, using an OPC client library to acquire and transmit real-time measurements from PLC-AHU9 to the fog gateway (i.e. Raspberry Pi), and pushing pertinent notifications to third-party systems. A simple configuration file was designed to guide the agent's execution process, primarily consisting of the agent's unique identifier and jobs matching PLC's with URL's of machine learning models hosted on the fog.

#### *FOG LAYER*

The fog layer contains the technical components needed to accept incoming measurements, automatically execute machine learning models and return the analytical outcomes to the factory. A summary of fog components and operations are described below;

- **Raspberry Pi** provided the platform for the fog gateway, which hosted the service API for factory-to-fog communications, machine learning models for particular engineering applications (e.g. AHU issue identification), and analytics execution engine for real-time scoring. The portability of the Raspberry Pi aligned with the principles of fog computing, whereby one may strategically deploy gateways to improve latency, and reduce dependencies on external communication channels (e.g. broadband).

- **RESTful API** provided an open and standards-driven interface that enabled embedded software agents in the factory, to invoke an AHU issue identification machine learning model residing on the fog gateway, using real-time measurements acquired from PLC-AHU9.

- **OpenScoring engine** supported the execution of PMML-encoded machine learning models stored on the fog gateway, using the RESTful API to obtain the input parameters for the model (e.g. AHU measurements), and exact name of the machine learning model to execute.

- **Engineering models** relates to a repository on the fog gateway that can store multiple PMML-encoded machine learning models. In the cyber-physical system implementation, a single AHU issue identification model was stored in the repository to demonstrate and measure the operation of the cyber-physical system. Given the embedded software agent must always be aware of the machine learning model to execute, the RESTful API and OpenScoring engine use this information to choose the engineering model from the repository.

- **Synchronisation engine** maintains the engineering models repository using a background service, which downloads new models, and updates existing models. A basic tagging system using comma separated values describes the fog gateway's engineering application (i.e. AHU monitoring), which is passed to the cloud's RESTful API to identify relevant PMML models for synchronisation.

#### *CLOUD LAYER*

The cloud layer contains the technical components needed to maintain information about the fog gateways deployed in the factory (e.g. identity, engineering applications etc.), persist machine learning models in a global repository, and discharge models to fog gateways when contextually relevant models are added or updated. A summary of cloud components and operations are described below;

- **AWS API Gateway** provided a secure and scalable service for implementing the cloud layers RESTful API, which was primarily used to communicate with factory deployed fog gateways (i.e. Raspberry Pi), and orchestrate the retrieval and dissemination of relevant PMML models. Apart from the auto-scaling and concurrent connectivity offered by the AWS API Gateway service, other benefits included in-built version management, and seamless integration with other AWS compute services (e.g. HTTP requests from the fog were connected to AWS's serverless computing service to distribute PMML models).

- **AWS Lambda** delivered a serverless compute platform for hosting and enacting software modules written in Python (e.g. lookup PMML models with particular tag), which enabled the technical process to focus on modular task-level development, rather than being concerned with the inherent overheads associated with server-level hosting, deployment and maintenance. The primary serverless modules deployed were (1) an authentication module to verify the credentials of each fog gateway, and (2) PMML synchronisation module to ensure each fog gateway could be automatically updated. These modules were connected to the AWS Gateway API so inbound HTTP requests from fog gateways were routed automatically to invoke the appropriate module. As with most AWS services, the AWS Lambda service possessed in-built auto-scaling capabilities to handle variable throughput, with no significant technical overhead in terms of administration or management.

- **AWS S3** persisted a global repository of production-ready and deployable PMML engineering models, which provided a scalable, fault tolerant, secure and version controlled environment for storing files. A simple tagging system was used to label PMML files (e.g. AHU for issue identification model) so the AWS Lambda service could match fog gateways with relevant models.

- **AWS RDS SQL** stored metadata about deployed fog gateways, including authentication credentials, factory environment, and tags denoting relevant engineering applications (e.g. AHU), which was consumed by AWS Lambda modules as needed. In addition to scalability, the AWS RDS SQL service provided automatic backup and disaster recovery features that were of particular importance and interest to the industrial partner's stakeholders.

### FACTORY-TO-CLOUD DATA ANALYTICS PIPELINE FOR MODEL BUILDING

A factory-to-cloud data pipeline was implemented to automate the common pre-processing routines for predictive modelling and data analysis. These routines included the data acquisition, cleaning and storage of energy data archives from the facilities building management system. Figure 48 illustrates the implemented system and cloud layers of the data pipeline, with components in the system layer comprising components that connect existing industrial systems to the cloud, and the cloud layer comprising cloud services configured to receive, process and serve the clean data.

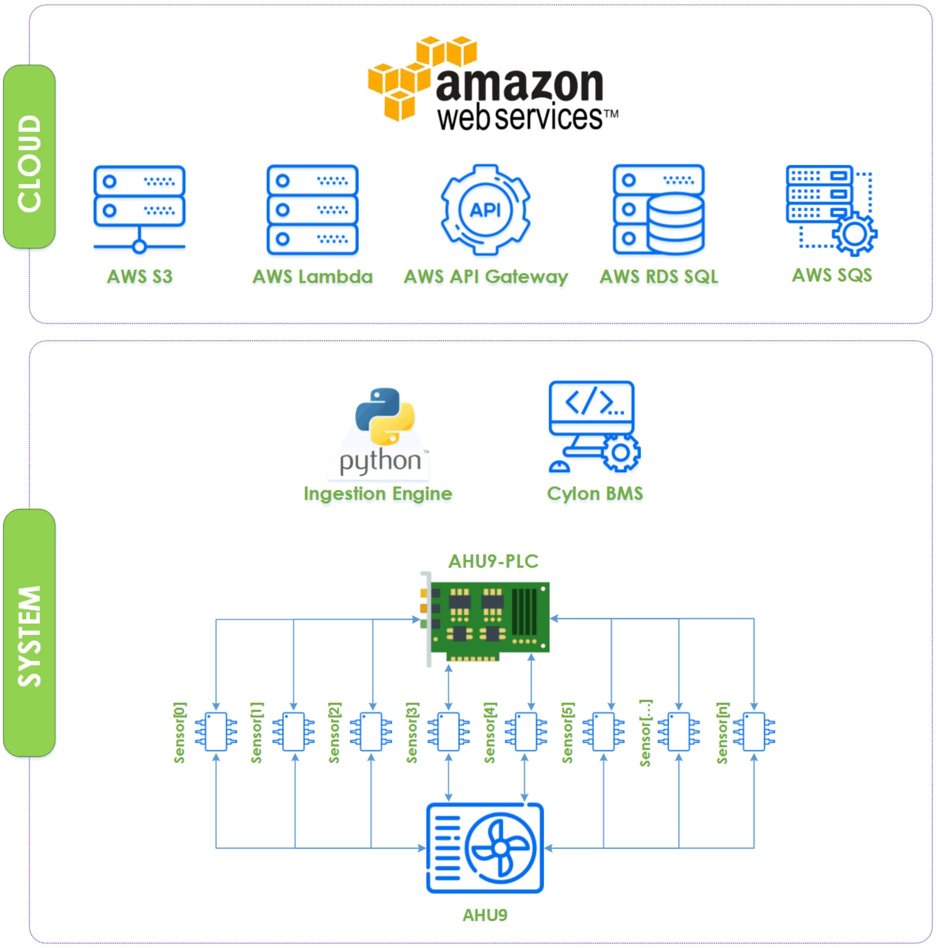



**Figure 48 Cloud computing architecture, components and technologies**

### SYSTEM LAYER

The system layer contains legacy components, systems or technologies operating in the factory, and purpose-built components that connect and stream operating data produced by these systems to the cloud. A summary of system layer components are described below;

- **Clyon BMS** periodically archived energy data every 24-hours, which included 15-minute measurements for AHU9. These archive log files were appended each day (i.e. data was not overwritten), and saved as human-readable CSV files on the local computer. Thereafter, the ingestion engine was programmed to interrogate the energy archive to achieve factory-to-cloud integration.

- **Ingestion engine** streamed archives of AHU operational data to the cloud every 24-hours. The engine was implemented using Python and AWS software development libraries, and deployed as a daily scheduled task on the Cylon BMS computer. Upon triggering the task each day, the ingestion engine read a configuration file to acquire communication endpoints (i.e. destination of data transmission), security credentials and targeted data sources. In this case, the ingestion engine was programmed to interrogate the BMS energy archive of CSV files, and transmit these logs to the cloud platform for processing.

#### *CLOUD LAYER*

The cloud layer contains managed cloud services that were configured and programmed to receive, process and store time-series operating data, which was extracted and transmitted from the building management system in the factory. A summary of cloud layer services and components are described below;

- **AWS API Gateway** functioned as the primary interface between the ingestion engine in the factory, and workflows residing on the cloud. After transmission from an ingestion agent in the factory completed, the AWS API Gateway triggered a workflow engine implemented in AWS Lambda, which initiated data cleaning and transformation workflows associated with the ingested data.

- **AWS Lambda** provided serverless compute resources to implement individual data processing workflow modules. In the context of the workflow implemented for the AHU, this involved the creation of workflow modules for (a) parsing Cylon CSV log files to produce a standard time-series format consisting of timestamp and measurement, (b) contextualisation of log files to provide meaningful names, and (c) merging of individual log files to produce a single logical dataset (e.g. combining measurements for AHU9 in a single file).

- **AWS RDS SQL** stored metadata pertaining to data processing workflows and modules, which was used by the workflow engine to establish the order of cleaning and transformation operations. The implemented relational model enabled modules to be associated with one or more workflows, and easily moved to different points in the workflow.

- **AWS S3** was used to implement a global data lake for energy archives, and stage the output from each step in a workflow (i.e. allowing the next step to take the previous output as input). The workflow processing modules (i.e. AWS Lambda) utilised AWS S3 as the primary method for persistence, which decoupled processing modules from each other, and provided some resilience against processing failures (i.e. processing modules were not chained and dependent on each other directly).

- **AWS SQS** maintained job queues for each AWS Lambda data processing module (e.g. recently uploaded Cylon data needs to be converted to a uniform format), and provided input parameters for these processing modules (e.g. the location of the AHU data file to be processed). These queues instilled resilience in data processing workflows by maintaining an ordered list of execution requests, which were only dispatched based on the availability of sufficient data processing resources.

### *CYLON/AHU WORKFLOW*

Figure 49 illustrates the processing pattern implemented for the AHU workflow, with each stage associated with an (a) *AWS SQS* message queue to receive instructions, (b) *AWS Lambda* background data processing module, and (c) *AWS S3* storage repository to persist output. After the ingestion engine transmits Cylon data to the cloud, the initial workflow engines checks to see if there are any associated workflows (i.e. cleaning and transformation for that data source). Given the availability of the AHU workflow, the engine retrieves the execution routine from *AWS RDS SQL*, and initiates the workflow by submitting a data processing job to the first queue (i.e. Stage 1). After the job has completed the output is stored in *AWS S3*, before adding a job to the queue of the next processing module (i.e. Stage 2). This feedforward pattern continues until there are no further data processing modules to execute, with the output of the last stage representing the final output of the workflow.

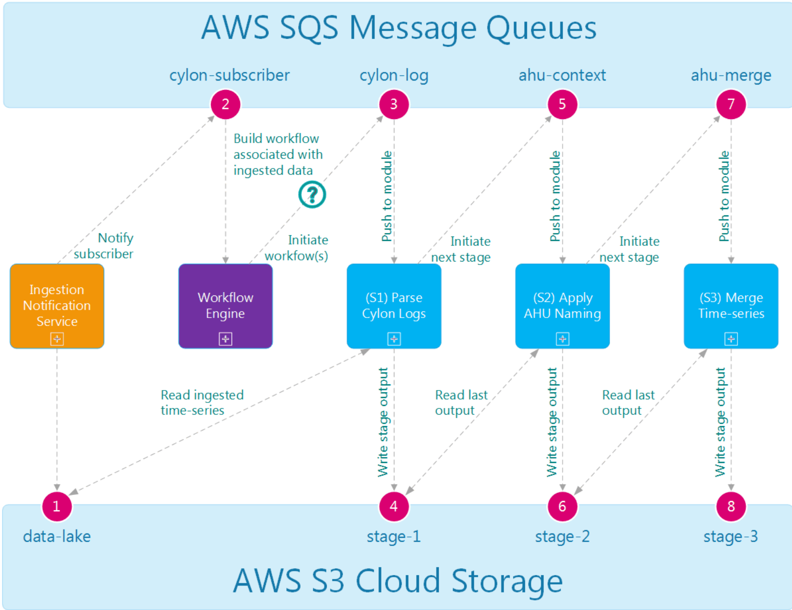



**Figure 49 Implemented AHU workflow pattern**

Figure 50 shows the native form of BMS energy data transmitted to *AWS S3*. This format is not analytics-ready given peculiarities in its structure. Firstly, header information on the first row provides metadata about the log file, such as the measurement type and interval (i.e. 900 seconds), rather than variables or features. Secondly, the structure is primarily designed to provide compact data redundancy for operational data, with each row containing data for the previous 10 days. Column A contains timestamps for the first measurement of each row, which is contained in Column C. For example, the first value on the second row has a timestamp of 14/04/12 17:30, with a corresponding measurement of 23.02. Timestamps for each measurement after Column C must be manually derived by incrementally adding 15-minute values to the first timestamp (i.e. from Column A). Column B specifies the number of measurements on each row, starting from Column C (i.e. first value). Although 1024 measurements are specified for each row in the screenshot, logging issues or outages can affect the number of measurements archived. Given a measurement frequency of 900 seconds (i.e. 15 minutes), each row of 1024 measurements should contain 10 days of data, which means adjacent rows overlap with 9 days of redundant data. Each row in the screenshot shows data logging at the same time every 24 hours (i.e. 5:30pm), but it was common for this pattern to shift without warning, which meant several control checks were needed to ensure timestamps and measurements were parsed correctly.

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | UC32net - 004 Free Cooling | UC3216 - 008 - AHU9 | TT-509-901 | 900 | | | |
| 2 | 14/04/12 17:30 | 1024 | 23.02 | 22.7 | 22.76 | 22.7 | 22.36 |
| 3 | 15/04/12 17:30 | 1024 | 22.29 | 22.33 | 22.16 | 21.63 | 21.33 |
| 4 | 16/04/12 17:30 | 1024 | 23.18 | 22.89 | 22.64 | 22.05 | 21.44 |
| 5 | 17/04/12 17:30 | 1024 | 21.82 | 21.87 | 21.78 | 21.65 | 21.71 |
| 6 | 18/04/12 17:30 | 1024 | 22.62 | 22.67 | 22.63 | 22.52 | 22.6 |
| 7 | 19/04/12 17:30 | 1024 | 23.93 | 23.83 | 23.65 | 23.48 | 23.46 |
| 8 | 20/04/12 17:30 | 1024 | 23.11 | 23.2 | 23.19 | 23.08 | 23.25 |
| 9 | 21/04/12 17:30 | 1024 | 23.28 | 23.49 | 23.41 | 23.33 | 23.35 |
| 10 | 22/04/12 17:30 | 1024 | 21.63 | 21.6 | 21.5 | 21.82 | 21.45 |
| 11 | 23/04/12 17:30 | 1024 | 21.45 | 21.52 | 21.44 | 21.81 | 21.69 |
| 12 | 24/04/12 17:30 | 1024 | 22.6 | 22.61 | 22.77 | 22.62 | 22.73 |
| 13 | 25/04/12 17:30 | 1024 | 22.35 | 22.21 | 22.33 | 22.17 | 22.05 |
| 14 | 26/04/12 17:30 | 1024 | 21.54 | 21.72 | 21.77 | 21.78 | 21.98 |
| 15 | 27/04/12 17:30 | 1024 | 22.87 | 22.75 | 22.7 | 22.73 | 22.7 |
| 16 | 28/04/12 17:30 | 1024 | 21.93 | 21.8 | 21.6 | 21.48 | 21.43 |

**Figure 50 Ingested log file for AHU return air temperature**

The first two stages in the AHU workflow transformed the Cylon BMS format to a basic time-series. Figure 51 shows the data output after cylon-log (stage 1 processing module) and ahu-points (stage 2 processing module) were applied. This shows data redundancy has been removed, with each row associated with a single observation (i.e. point-in-time), and each column representing a single measurement. The normalisation of BMS data provided subsequent processing modules with a more conventional format upon which to execute data transformations.

| | A | B |
|---|---|---|
| 1 | Timestamp | Value |
| 2 | 14/04/12 17:30 | 23.02 |
| 3 | 14/04/12 17:45 | 22.7 |
| 4 | 14/04/12 18:00 | 22.76 |
| 5 | 14/04/12 18:15 | 22.7 |
| 6 | 14/04/12 18:30 | 22.36 |
| 7 | 14/04/12 18:45 | 22.43 |
| 8 | 14/04/12 19:00 | 22.26 |
| 9 | 14/04/12 19:15 | 21.99 |
| 10 | 14/04/12 19:30 | 21.9 |

**Figure 51 AHU return air temperature after Stage 1 and 2**

Figure 52 shows the output from stage 3, where individual sensor logs for AHU9 were merged to a tidy dataset. This dataset represents a single entity (i.e. AHU9), with each row containing a single observation (i.e. point-in-time), and each column containing a single measurement (e.g. return air temperature). The availability of such formats can greatly reduce the data wrangling and pre-processing effort associated with data analytics, while a formal naming convention was borrowed from previous research to identify individual AHU measurements [1].

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | Timestamp | retT | outT | mixT | heaT | cooT | supT | zonT |
| 2 | 14/04/12 00:00 | 21.11 | 0 | 7.7 | 7.55 | 7.62 | 11.72 | 20.72 |
| 3 | 14/04/12 00:15 | 21.53 | 0 | 7.46 | 7.11 | 7.15 | 11.23 | 20.82 |
| 4 | 14/04/12 00:30 | 21.78 | 0 | 8.83 | 8.37 | 8.52 | 12 | 20.93 |
| 5 | 14/04/12 00:45 | 21.97 | 0 | 9.07 | 8.54 | 8.63 | 12.03 | 21.07 |
| 6 | 14/04/12 01:00 | 22 | 0 | 8.83 | 8.28 | 8.49 | 11.94 | 21.14 |
| 7 | 14/04/12 01:15 | 22.07 | 0 | 9.14 | 8.7 | 8.8 | 12.15 | 21.23 |
| 8 | 14/04/12 01:30 | 22.2 | 0 | 8.9 | 8.39 | 8.53 | 11.79 | 21.25 |
| 9 | 14/04/12 01:45 | 22.14 | 0 | 9.03 | 8.69 | 8.9 | 12.13 | 21.21 |
| 10 | 14/04/12 02:00 | 22.24 | 0 | 9.21 | 8.59 | 8.63 | 12.1 | 21.29 |
| 11 | 14/04/12 02:15 | 22.29 | 0 | 8.68 | 8.14 | 8.39 | 11.72 | 21.36 |
| 12 | 14/04/12 02:30 | 22.07 | 0 | 8.98 | 8.5 | 8.66 | 12.11 | 21.38 |
| 13 | 14/04/12 02:45 | 22.1 | 0 | 8.73 | 8.17 | 8.42 | 11.93 | 21.39 |
| 14 | 14/04/12 03:00 | 22.15 | 0 | 8.82 | 8.18 | 8.34 | 12.06 | 21.38 |
| 15 | 14/04/12 03:15 | 22.35 | 0 | 8.68 | 8.13 | 8.28 | 11.91 | 21.41 |
| 16 | 14/04/12 03:30 | 22.49 | 0 | 8.61 | 8.03 | 8.28 | 11.85 | 21.44 |
| 17 | 14/04/12 03:45 | 22.52 | 0 | 8.6 | 8.2 | 8.29 | 12.04 | 21.48 |
| 18 | 14/04/12 04:00 | 22.28 | 0 | 8.4 | 8.04 | 8.19 | 11.9 | 21.43 |
| 19 | 14/04/12 04:15 | 22.11 | 0 | 8.9 | 8.6 | 8.56 | 12.03 | 21.38 |
| 20 | 14/04/12 04:30 | 22.03 | 0 | 8.69 | 8.47 | 8.56 | 12.01 | 21.37 |
| 21 | 14/04/12 04:45 | 21.84 | 0 | 8.47 | 8.14 | 8.25 | 11.71 | 21.26 |
| 22 | 14/04/12 05:00 | 21.79 | 0 | 8.81 | 8.5 | 8.48 | 12.1 | 21.21 |
| 23 | 14/04/12 05:15 | 22.06 | 0 | 14.98 | 11.08 | 9.24 | 12.62 | 21.41 |
| 24 | 14/04/12 05:30 | 21.53 | 0 | 16.4 | 12.39 | 9.7 | 14.3 | 21.64 |
| 25 | 14/04/12 05:45 | 21.09 | 0 | 17.03 | 13.31 | 10.42 | 15.21 | 21.85 |
| 26 | 14/04/12 06:00 | 20.93 | 0 | 16.93 | 13.93 | 11.37 | 15.77 | 22.01 |
| 27 | 14/04/12 06:15 | 20.89 | 0 | 16.35 | 14.07 | 12.09 | 16.29 | 22.06 |

**Figure 52 AHU log file after Stage 3**

### 4.4.2 Implementation assessment

During the implementation the assessment criteria from the design methodology were used to evaluate technical decisions. The intention of the qualitative assessment criteria is to ensure important criteria are being considered, rather than providing a mechanism to measure compliance with particular criteria. However, a more quantitative evaluation of the cyber-physical system implementation shall be presented when measuring performance (section 4.5). The following sections outline the assessment criteria proposed by the design methodology, and describes how these criteria were ultimately addressed by the technical implementation.

#### INDUSTRY 4.0 ASSESSMENT

Table 26 presents the Industry 4.0 assessment criteria proposed by the design methodology, and comments summarising how these criteria were addressed within the technical implementation.

| Criteria | Comments |
|---|---|
| Interoperability | Given the implemented cyber-physical system promotes open and standard communication using HTTP, messaging interoperability between systems can be achieved without impediment (e.g. factory agent sends push notifications to other services when something changes), while also enabling components of the cyber-physical system to be easily substituted (e.g. Python factory agent can be implemented by another |

| | |
|---|---|
| | organisation using Java). In addition to messaging interoperability, incorporating PMML to persist and execute machine learning models, ensures compatibility and interoperability with a wide-range of statistical and analytics software, which enables personnel to build, deploy and import models from the proposed cyber-physical system. |
| Virtualisation | The implemented cyber-physical system enables virtualisation using an embedded software agent (i.e. Python software agent) to stream real-time AHU measurements to the fog gateway (i.e. Raspberry Pi), where an issue identification machine learning model provides self-awareness insights regarding current state. Of course, more complex and aggregated virtualisation scenarios shall arise, but the basic pattern of operation between the agent and gateway may persist. |
| Decentralisation | The architecture of fog and agent computing promotes decentralised storage, processing and intelligence, where operations are undertaken on the edge of networks, rather than centralised in a particular locale. The fog gateway (i.e. Raspberry Pi) possessed the necessary software, machine learning and communication capabilities to (a) operate autonomously across different networks, and (b) service embedded software agents monitoring components, machinery or processes. |
| Real-time | Although the real-time, scalability and resilience of the implemented cyber-physical system shall be discussed in the following sections, the decentralised and localised characteristics of fog computing encourages low-latency, reliable and resilient communication, especially when compared to cyber-physical systems depending on roundtrips between the factory and cloud. |
| Service Oriented | Messaging between the main factory, fog and cloud components depend exclusively on RESTful API services, while the utilisation of SOAP-based services were needed to acquire real-time measurements from the OPC-UA server. In addition to service interactions between components of the cyber-physical system, embedded software agents implemented push notifications to third-party systems enabling these messages to be sent to a service endpoint (i.e. other system listening for incoming messages). |
| Modularity | Given the cyber-physical system embraces decentralisation and service-orientation, modularity naturally occurs given the low-level operations and processes are hidden behind interfaces (e.g. agent sends and receives outcome of model execution without knowing the type of model or software that undertook the analysis). In the case of the cyber-physical system, service interfaces provide access to the different modules. |

**Table 26 Industry 4.0 assessment criteria**

### Stakeholder assessment

Table 27 presents the common stakeholder assessment criteria proposed by the design methodology, and comments summarising how these criteria were addressed within the technical implementation.

| Criteria | Comments |
|---|---|
| Data security | One of the common concerns associated with cyber-physical systems regards the transmission and persistence of operational data in the cyber- |

| | |
|---|---|
| | world, which has been traditionally associated with the cloud. However, the implemented cyber-physical system utilises the fog computing paradigm to refrain from transmitting or persisting operational data outside the organisations local or extended network, and thus ensuring the cyber-physical system can align with existing policies governing the operation of local networked devices (e.g. PLC's). |
| Legacy integration | Given the historic investment placed in existing control networks and technology infrastructure, facilities can resist change when it requires extensive technology replacement (e.g. installing internet-of-things smart sensors). The implemented cyber-physical system addresses this concern using embedded software agents, which may embody different protocols to interact with many different devices. In this particular implementation of the system, the embedded software agent utilised OPC to stream measurements from a legacy PLC, rather than requiring the industry partner to replace the existing technology. |
| Regulation | Many large-scale manufacturing facilities adhere to strict quality assurance and regulation, the content of which may include technologies, systems and processes governing production. Thus, some facilities may be concerned that new technology paradigms may infringe upon current regulatory compliance. While one cannot state the cyber-physical system complies with all regulatory requirements, the implementation strategy promotes the ideology of extension rather than replacement. This meant existing technologies that may already have been subject to audit or assessment, were always used instead of introducing new technologies or components (e.g. BMS archiving, PLC's for real-time measurements). |
| Performance | The performance and dependability of industrial automation and control systems has been demonstrated for decades, and therefore, extending these stable systems to incorporate emerging technologies naturally raises concerns regarding performance (e.g. guaranteed execution times, communication failure rates, concurrent throughput etc.). Given cyber-physical systems incorporate additional compute layers (e.g. operating systems, analytics engines), deterministic and consistent real-time execution shall exceed that of control networks. However, the local and embedded topology of the fog compute cyber-physical system aims to address two primary performance concerns by (a) reducing dependencies on external communication channels (i.e. broadband) that could result in control messages being dropped, and (b) promoting shorter roundtrips between the physical-world (e.g. embedded software agent) and cyber-world (e.g. Raspberry Pi gateway). |
| Disruption | Traditionally, technology development and adoption carries the risk of disruption to standard business operations. Such concerns are amplified in real-time manufacturing production scenarios, where there can be less tolerance to missed or inaccurate operation. To reduce some concerns regarding disruption, the implemented cyber-physical system separates existing control technology (i.e. sensing layer) from the cyber-physical technology (i.e. fog and cloud), and therefore does not interfere with the current operation of the control network. However, when control engineers begin to depend on real-time results and outputs from the cyber-world to influence control logic, the potential for disruption increases given tighter integration. |
| Knowledge | A practical and well-founded concern pertaining to the adoption of cyber-physical systems, relates to the current knowledge, skillset, and expertise of personnel within the organisation (e.g. machine learning, |

| | |
|---|---|
| | internet-of-things, big data). Where sufficient internal expertise does not exist, facilities may become dependent on consultants and service providers to deliver key strategic technologies and infrastructure for Industry 4.0 operations. The implemented cyber-physical system aims to reduce the knowledge burden by (a) following a systematic design methodology that prescribes the relationships and interactions, and (b) delivering a technical platform that abstracts and automates some of the technical details (e.g. uploading a PMML file can automatically make the machine learning model available to embedded software agents that may already exist in the factory). |

**Table 27 Stakeholder assessment criteria**

### FUNCTIONAL ASSESSMENT

Table 28 presents the functional assessment criteria proposed by the design methodology, and comments summarising how these criteria were addressed within the technical implementation.

| Criteria | Comments |
|---|---|
| Data integration | A core requirement of cyber-physical systems is the need to integrate data from systems and sensors, encompassing both legacy and contemporary technologies. The periodic integration of the BMS's log files was implemented to support model building, while the continuous streaming of real-time AHU operating measurements was achieved using OPC-enabled embedded software agents. As more integration scenarios are encountered, the embedded software agents may be extended to incorporate additional communication drivers and protocols, without impacting the fundamental operation of the cyber-physical system. |
| Data processing | Given the multitude of industrial technologies and information systems from which operational data can be collected, the ability to process, transform and clean data is critical to produce harmonised, consistent and homogeneous analytics-ready data. The analytics pipeline supporting model building efforts implemented such features, with AWS S3 utilised as a data lake to store inbound data from the BMS, and AWS Lambda and AWS SQS used to realise a highly-scalable (i.e. auto-scaling) and fault tolerant workflow. Of course, some data processing components developed for the Cylon BMS and AHU workflow may need to be extended when new data sources are added, while other data processing components (e.g. detect missing values) are naturally more reusable. |
| Model execution | An overlooked aspect of industrial analytics relates to the ability to integrate and execute models (e.g. machine learning models) as part of enterprise and industrial information systems, with more attention given to developing analytics pipelines, data exploration and model building using statistical software applications (e.g. R, Tableau etc.). The cyber-physical system implementation enables the transition from model building to execution using the PMML standard for encoding machine learning models, and the OpenScoring engine to provide model execution capabilities that could be enacted independent of the underlying software. |
| Real-time scoring | To affect real-time decision-making in the factory, continuous streams of operating data must be transmitted and scored by a suitable analytics |

| | |
|---|---|
| | engine. Given the availability of model execution capabilities (i.e. PMML and OpenScoring engine), an embedded software agent capable of mediating communications between the physical-world and cyber-world was developed. In the implemented cyber-physical system, the embedded software agent utilised OPC to read continuous measurements from AHU9, which was encoded as a JSON request object, and transmitted to the OpenScoring analytics engine running on the Raspberry Pi (i.e. fog gateway) to achieve real-time scoring. |
| Score validity | Ensuring analytics models are of sufficient quality (e.g. accuracy, execution speed etc.) before deployment is dependent on model building processes. Although the analytics pipeline built for the Cylon BMS and AHU operational data simplifies the process of data exploration and model building, it does not directly affect the validity of the models produced. However, this assumes processing routines applied to incoming data have not introduced data issues (i.e. misaligning timestamps etc.). Therefore, when new data processing routines and workflows are introduced, the data output needs to be sampled and compared with the original source to flag any potential issues. |
| Score propagation | Due to the interconnected nature of Industry 4.0 and cyber-physical systems, many systems may need to know of particular operational changes in the factory (e.g. machine learning model finds issue in an AHU, and notifies facilities a replacement is needed). The embedded software agent in the cyber-physical system mediates communications between the physical and cyber-worlds, and maintains an active list of service URL's to push notifications to third-party systems. Of course, the receiving system must be listening for transmissions at the specified service URL, and expect the agreed data format (e.g. JSON object currently used to define AHU issue). |

**Table 28 Functional assessment criteria**

## 4.5 Phase 4: Performance measurement

The primary objective of the performance phase was to observe the lower and upper execution boundaries (i.e. execution latency) of the technical implementation. Such system properties must be known for control and engineering scenarios that are time dependent. Although the technical implementation embodies batch and real-time operations, only those associated with cyber-physical interactions are typically subject to low-latency demands. Therefore, the cyber-physical interface connecting the physical (e.g. factory) and cyber (e.g. fog gateway) worlds is the fundamental component being measured. The following section describes the setup, configuration and execution of load testing experiments to measure the performance of the technical implementation using cloud-based and fog-based cyber-physical interfaces.

### 4.5.1 Setup

The cloud and fog cyber-physical technologies used for the experiment utilised entry-level and standard configurations. Given the contrasting differences in architecture and resources, choosing an out-of-the-box approach appeared to be the easiest way to protect from potential biases, while experiments were executed in close proximity to reduce fluctuating environmental conditions (e.g. broadband throughput, local network activity etc.) contaminating the observed measurements. Figure 53 illustrates the main components of the experimental setup. A test computer was setup to host JMeter - an open source Apache application for load testing web resources and API's. The JMeter agent was configured with experiment parameters to send, receive and measure transmissions for each cyber-physical interface. The platforms behind the cyber-physical interfaces were furnished with a simple predictive model encoded as PMML, with the OpenScoring engine installed to enable real-time scoring of the model.

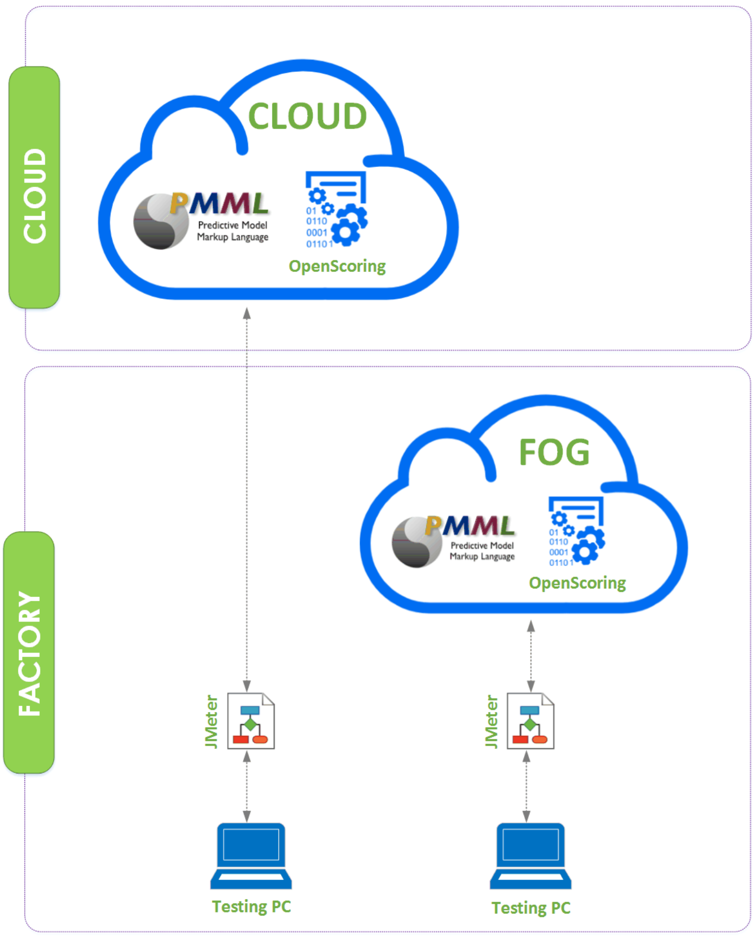



**Figure 53 Experiment setup for cloud and fog cyber-physical interfaces**

### Network environment

Although the implemented industrial analytics pipeline and cyber-physical system were deployed to the industrial partners operating environment, the load and stress testing of these implementations needed to be undertaken off-site to prevent any negative affect on existing networks or infrastructure. Therefore, to facilitate the experiment a temporary local network was setup in the university. A Huawei HG659B wireless router with 4 Ethernet ports was used to create a local new network, comprising two primary devices - (a) Test-PC hosting the JMeter agent software for logging request/response latency for cloud and fog cyber-physical interfaces, and (b) Raspberry Pi with the OpenScoring engine to facilitate the real-time scoring of machine learning models. The Test-PC specification consisted of an Intel Core i5-4380U CPU @ 2.80GHz processor, 4 GB memory, and 200 GB solid-state hard drive, running on Windows 8.1 Enterprise. Once devices were setup on the network, several broadband speed tests were undertaken to ensure consistent throughput for cyber-physical interactions with the cloud, with observed measurements ranging from 30-40Mbps download and 8-10Mbps upload.

### JMeter agent for logging

To measure cyber-physical interactions (e.g. factory to cloud analytics), the JMeter software was installed on the BMS PC to operate as a surrogate for the embedded factory agent in the cyber-physical system. This enabled load testing measurements to isolate the performance of cyber-physical interactions, which represent the extension point between legacy and emerging control paradigms. Although other factors may impede performance (e.g. control protocols used to read measurements), the objective of the performance evaluation is to demonstrate the operating boundaries and limitations associated with the embedding of advanced analytics. Therefore, the cyber-physical interface and interactions between the factory and analytics execution are the performance measurements of interest.

### Cloud interface and configuration

The cloud-based cyber-physical interface was constructed using a general-purpose Amazon Web Services (AWS) EC2 compute instance – more specifically, a t2.micro instance comprising one virtual CPU, one gigabyte of memory, and Linux-based operating system. Given the availability of the cloud compute platform, the

OpenScoring engine was downloaded from the GitHub repository, before a build process was executed using Apache Maven to create the directories and executables. Once the build process completed, a PMML encoded AHU issue identification model was copied to the default model directory (i.e. /openscoring/model) and named *AhuHeaTFault.pmml*. Finally, the Java-based server-side standalone version of the OpenScoring analytics engine was initiated, which started the RESTful web service interface to enable client applications to score the AHU model (e.g. factory agent, JMeter agent etc.). To ensure client HTTP requests could pass-through AWS security policies, the default security configuration was updated to permit HTTP requests to the OpenScoring engine on port 8080.

**FOG INTERFACE AND CONFIGURATION**

The fog-based cyber-physical interface was developed using a Raspberry Pi3 Model B to host the analytics model and execution engine. This device comprised a 64-bit ARMv8 1.2GHz processor, one gigabyte of memory, in-built Wi-Fi capabilities, and Linux-based operating system. The setup and configuration of the OpenScoring engine and analytics model for testing (i.e. AHU model) mirrored that of the cloud interface. The OpenScoring engine was downloaded and built using Apache Maven to produce the necessary directories and executables. After the build process completed, the PMML encoded model was copied to the OpenScoring engine's model directory (i.e. /openscoring/model). Finally, the Java-based server-side OpenScoring analytics engine was started to enable client applications score the AHU model (e.g. factory agent, JMeter agent etc.). Although AWS demonstrated default security policies that impeded communication on port 8080, the Raspberry Pi did not require such modifications given it was situated on a local trusted network. However, opening ports to allow communications may be necessary where local firewalls separate client applications (e.g. factory agent) and the analytics engine (e.g. OpenScoring RESTful service).

## 4.5.2 Performance analysis

After some experimentation with stress parameters, an interval of 250 milliseconds was chosen to execute test cases. This interval provided sufficient frequency to stress the system when scaled to hundreds of concurrent connections, while also providing the opportunity to illustrate workloads that can be easily managed. Table 29 describes the properties for execution and load testing scenarios (i.e. T1 to T4). The number of concurrent threads for each test determines the level of stress placed on the cyber-physical interface, with each thread tasked with executing 1,000 communication request/response loops.

| Test | Threads (concurrent) | Loops (per thread) | Total Loops |
|---|---:|---:|---:|
| T1 | 50 | 1,000 | 50,000 |
| T2 | 100 | 1,000 | 100,000 |
| T3 | 250 | 1,000 | 250,000 |
| T4 | 500 | 1,000 | 500,000 |

**Table 29 Concurrent execution scenarios**

Figure 54 shows the JMeter load testing plan comprising the aforementioned execution scenarios for cloud (AWS) and fog (Pi) cyber-physical interfaces. The screenshot shows the basic naming convention used to differentiate between cyber-physical interfaces, concurrent threads, and execution loops for scenario, while also highlighting the main JMeter controls used to measure and monitor execution (i.e. HTTP Request, View Results Table etc.).

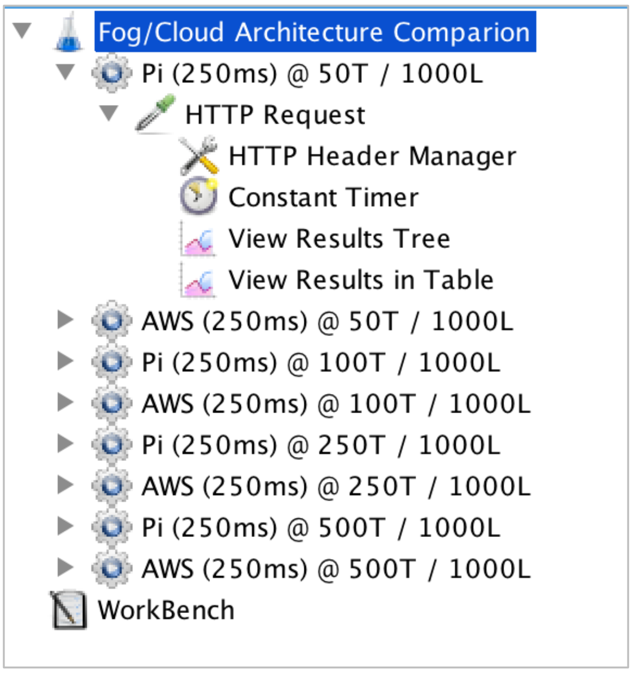



**Figure 54 JMeter load testing plan**

Figure 55 shows the HTTP request configuration employed for real-time scoring of the industrial analytics model. With the exception of the IP address and port number, these settings are the same for each execution scenario. The following points summarise the primary configuration parameters;

- **Protocol** specifies the communication protocol to use during the load testing scenarios. In this instance, HTTP was employed to align with the RESTful API exposing the OpenScoring engine for model execution.

- **IP address and port number** define the endpoint (e.g. server) of the analytics execution engine (i.e. OpenScoring), and the port upon which communication should be directed. The values displayed relate to the fog interface, which resided on the local network, and listened for inbound messages (e.g. real-time operating measurements) on port 8080. Of course, these values changed for cloud interface testing scenarios.

- **Method and path** describe the HTTP verbs that should be used to transmit measurements to the OpenScoring engine (e.g. POST, GET, PUT), and the relative path to the model that should be executed. In this instance, the POST method was used to adhere to the OpenScoring interface, with the AHU issue identification model used for each execution scenario (i.e. AhuHeaTFault).

- **Body data** contains the input parameters expected by the analytics model, with each numerical measurement set to zero. During the load testing process, JMeter packages the body data in a HTTP request using the previously defined variables (e.g. protocol, IP address etc.), and receives a HTTP response with the result of the model execution (i.e. real-time scoring result).

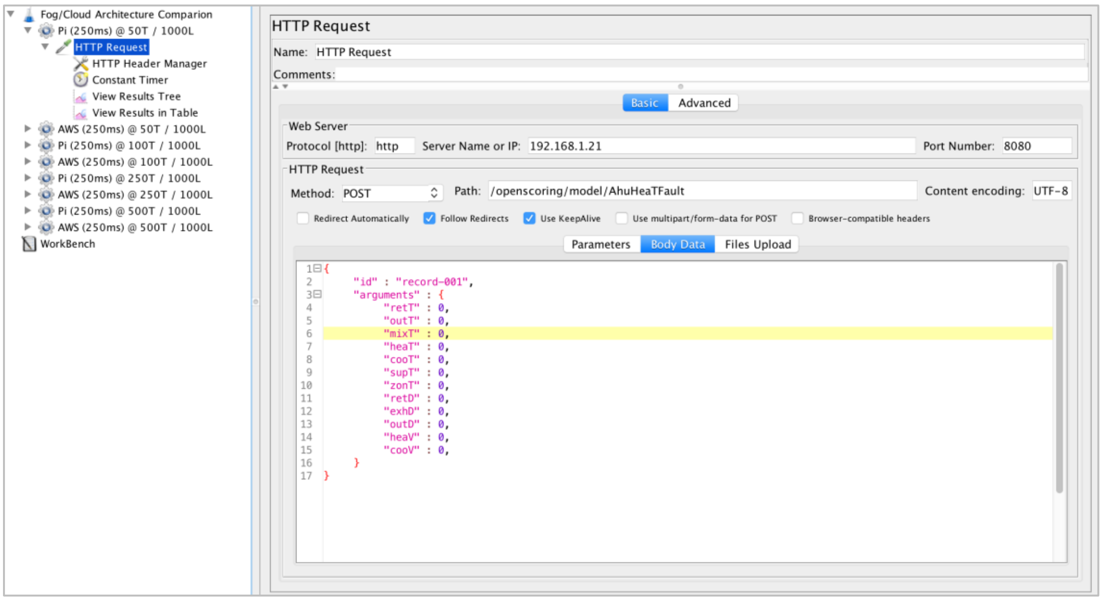


**Figure 55 HTTP request configuration for real-time scoring**

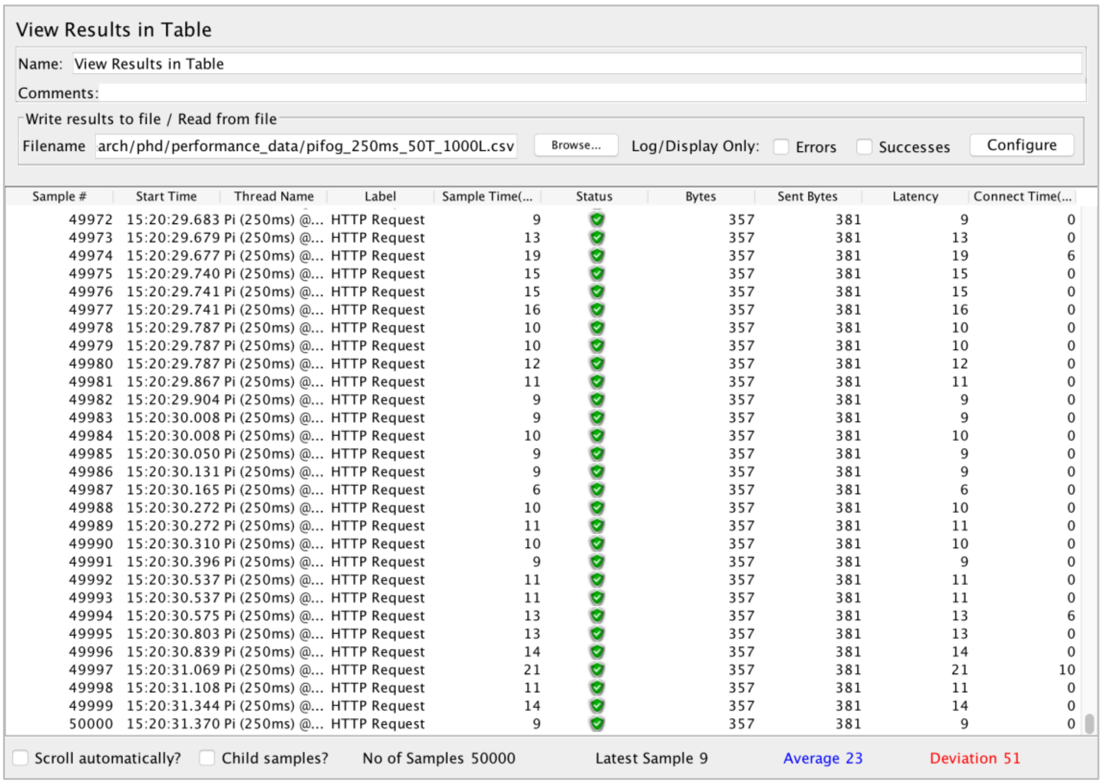
Figure 56 shows JMeter's view results component that continuously logs HTTP requests and responses during load testing. The optional filename property was set for each execution scenario, with a simple naming convention employed to label the results for each scenario. These log files provided the data necessary to interrogate and visualise the performance of fog and cloud cyber-physical interfaces. The main column headings related to communication performance are described below;

- **Sample** (milliseconds) logs the time between when the request was sent, and when the agent has received and processed the entire response.

- **Status** (success/fail) logs whether a particular request received a successful response, without raising errors or exceptions (i.e. request sent, model executed, and response returned).

- **Latency** (milliseconds) logs the time between when the request was sent, and when the agent began to receive the response (i.e. not complete processing of the response).

- **Connection time** (milliseconds) logs the time taken to establish a connection between the client (i.e. JMeter) and server (i.e. cyber-physical interface), before transmitting the load testing HTTP request.



View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename arch/phd/performance_data/pifog_250ms_50T_1000L.csv Browse... Log/Display Only: Errors Successes Configure

| Sample # | Start Time | Thread Name | Label | Sample Time(... | Status | Bytes | Sent Bytes | Latency | Connect Time(... |
|---|---|---|---|---|---|---|---|---|---|
| 49972 | 15:20:29.683 | Pi (250ms) @... | HTTP Request | 9 | | 357 | 381 | 9 | 0 |
| 49973 | 15:20:29.679 | Pi (250ms) @... | HTTP Request | 13 | | 357 | 381 | 13 | 0 |
| 49974 | 15:20:29.677 | Pi (250ms) @... | HTTP Request | 19 | | 357 | 381 | 19 | 6 |
| 49975 | 15:20:29.740 | Pi (250ms) @... | HTTP Request | 15 | | 357 | 381 | 15 | 0 |
| 49976 | 15:20:29.741 | Pi (250ms) @... | HTTP Request | 15 | | 357 | 381 | 15 | 0 |
| 49977 | 15:20:29.741 | Pi (250ms) @... | HTTP Request | 16 | | 357 | 381 | 16 | 0 |
| 49978 | 15:20:29.787 | Pi (250ms) @... | HTTP Request | 10 | | 357 | 381 | 10 | 0 |
| 49979 | 15:20:29.787 | Pi (250ms) @... | HTTP Request | 10 | | 357 | 381 | 10 | 0 |
| 49980 | 15:20:29.787 | Pi (250ms) @... | HTTP Request | 12 | | 357 | 381 | 12 | 0 |
| 49981 | 15:20:29.867 | Pi (250ms) @... | HTTP Request | 11 | | 357 | 381 | 11 | 0 |
| 49982 | 15:20:29.904 | Pi (250ms) @... | HTTP Request | 9 | | 357 | 381 | 9 | 0 |
| 49983 | 15:20:30.008 | Pi (250ms) @... | HTTP Request | 9 | | 357 | 381 | 9 | 0 |
| 49984 | 15:20:30.008 | Pi (250ms) @... | HTTP Request | 10 | | 357 | 381 | 10 | 0 |
| 49985 | 15:20:30.050 | Pi (250ms) @... | HTTP Request | 9 | | 357 | 381 | 9 | 0 |
| 49986 | 15:20:30.131 | Pi (250ms) @... | HTTP Request | 9 | | 357 | 381 | 9 | 0 |
| 49987 | 15:20:30.165 | Pi (250ms) @... | HTTP Request | 6 | | 357 | 381 | 6 | 0 |
| 49988 | 15:20:30.272 | Pi (250ms) @... | HTTP Request | 10 | | 357 | 381 | 10 | 0 |
| 49989 | 15:20:30.272 | Pi (250ms) @... | HTTP Request | 11 | | 357 | 381 | 11 | 0 |
| 49990 | 15:20:30.310 | Pi (250ms) @... | HTTP Request | 10 | | 357 | 381 | 10 | 0 |
| 49991 | 15:20:30.396 | Pi (250ms) @... | HTTP Request | 9 | | 357 | 381 | 9 | 0 |
| 49992 | 15:20:30.537 | Pi (250ms) @... | HTTP Request | 11 | | 357 | 381 | 11 | 0 |
| 49993 | 15:20:30.537 | Pi (250ms) @... | HTTP Request | 11 | | 357 | 381 | 11 | 0 |
| 49994 | 15:20:30.575 | Pi (250ms) @... | HTTP Request | 13 | | 357 | 381 | 13 | 6 |
| 49995 | 15:20:30.803 | Pi (250ms) @... | HTTP Request | 13 | | 357 | 381 | 13 | 0 |
| 49996 | 15:20:30.839 | Pi (250ms) @... | HTTP Request | 14 | | 357 | 381 | 14 | 0 |
| 49997 | 15:20:31.069 | Pi (250ms) @... | HTTP Request | 21 | | 357 | 381 | 21 | 10 |
| 49998 | 15:20:31.108 | Pi (250ms) @... | HTTP Request | 11 | | 357 | 381 | 11 | 0 |
| 49999 | 15:20:31.344 | Pi (250ms) @... | HTTP Request | 14 | | 357 | 381 | 14 | 0 |
| 50000 | 15:20:31.370 | Pi (250ms) @... | HTTP Request | 9 | | 357 | 381 | 9 | 0 |

Scroll automatically? Child samples? No of Samples 50000 Latest Sample 9 Average 23 Deviation 51

**Figure 56 JMeter logging of concurrent execution**

### 4.5.3 Threats to performance validity

The performance measurements collected during testing can be influenced by different operating environments and parameters. However, these particular variables are largely unpredictable, and representative of real-world environments (i.e. every factory shall comprise unique and changing variables). The following points summarise the primary threats considered during the testing phase;

- **Environment:** the environment used for measuring performance comprises hardware, software and services to facilitate connectivity (i.e. switches, routers, cabling, bandwidth etc.). The quality, specification and configuration of these resources can vary significantly, and are subject to random times of decreased performance (e.g. high network traffic).

- **Model:** the analytics model used may impact measured performance when the required input datasets are large (i.e. transmission overhead), or computational complexity of the model increases the runtime of the execution phase (i.e. delaying response to the factory agent).

- **Connectivity:** the measured performance of the cloud interface depends on broadband uptime, throughput and consistency, which can change dynamically depending on service provider agreements and network policies.

- **Benchmark:** given the stress testing protocol executes requests at 250 millisecond intervals, the analysis uses 250 milliseconds as the latency benchmark to demonstrate where control threads begin to overlap. This benchmark can be replaced with a time-constraint for a particular industrial engineering and control application.

## 4.6 Chapter conclusions

This chapter applied the previously presented design methodology to produce an industrial analytics architecture comprising two sub-systems. These complimentary systems were the (a) data analytics pipeline to transmit, clean and store archived data needed for data analysis and modelling, and (b) industrial cyber-physical system for delivering embedded machine learning applications using fog computing. After the implementation process, several experiments were designed and executed to evaluate and compare the performance of cloud and fog cyber-physical interfaces as a means of delivering near/real-time analytics to the factory. The results of these experiments are analysed and discussed in the next chapter.

# Chapter 5
## Results & Discussion

## 5.1 Chapter introduction

This chapter presents results from the technical implementation undertaken in Phase 3 of the methodology, with the intention of establishing performance boundaries that demonstrate the system's ability to comply with timing constraints (e.g. real-time requirements) and communication throughput (e.g. concurrent connections) for particular engineering applications, while also illustrating how the proposed cyber-physical system implementation aligns with Industry 4.0 design principles. The primary sections of this chapter include;

- **Cyber-physical performance (section 5.2)** compares latency and consistency of cloud and fog interfaces, which are responsible for connecting measurements from the factory, with advanced analytics models in the cyber world.
- **Industrial analytics capability (section 5.3)** evaluates the broader benefits realised from the real-world implementation, including alignment with Industry 4.0 design principles and stakeholder concerns.

## 5.2 Cyber-physical performance

The extension of industrial control and automation networks for Industry 4.0 depend on cyber-physical interfaces (e.g. fog and cloud). These interfaces provide gateways to advanced and complex computational resources (e.g. machine learning, big data, cloud computing etc.), which possess the technical potential to deliver self-aware and self-optimising factories. However, given the technical overheads (e.g. system software, model execution and inter-networking) associated with these resources, cyber-physical interfaces cannot deliver the same performance as embedded systems. Therefore, cyber-physical performance must be carefully measured and evaluated to determine suitability for particular control and engineering scenarios. The primary measures of performance presented and discussed in this analysis are;

- **Latency of interfaces (section 5.2.1)** relates to the exact time it takes to transmit operational data to the cyber-world, execute an industrial analytics model, and return the result to the factory to inform decision-making.

- **Consistency of interfaces (section 5.2.2)** refers to how often communications between the cyber (e.g. cloud) and physical worlds (i.e. factory) were successful, and identifies points of performance degradation.

A load testing procedure was undertaken (section 4.5) on cyber-physical interfaces supported by fog and cloud computing architectures, with the intention of comparing the performance of the novel fog architecture for delivering embedded industrial analytics implemented during this research, and more common cloud computing methods. The hardware, configurations and infrastructure employed during data collection were based on default configurations. Table 30 outlines the execution scenarios (i.e. T1 to T4) and associated parameters used for load testing fog and cloud cyber-physical interfaces.

| Results | Test | Threads (concurrent) | Loops (per thread) | Total Loops |
|---|---|---:|---:|---:|
| Figure 57 | T1 | 50 | 1,000 | 50,000 |
| Figure 58 | T2 | 100 | 1,000 | 100,000 |
| Figure 59 | T3 | 250 | 1,000 | 250,000 |
| Figure 60 | T4 | 500 | 1,000 | 500,000 |

**Table 30 Load testing concurrent execution scenarios**

Figure 57, Figure 58, Figure 59, and Figure 60 visualise the results captured from the load testing scenarios (i.e. T1 to T4), and provide summary statistics relating to each cyber-physical interfaces performance dataset. Although additional scenarios were undertaken during this research, the T1 to T4 scenarios sufficiently demonstrated the performance degradation and vulnerabilities of both cyber-physical interfaces, which made many of the other results redundant and repetitive (e.g. 200 threads told the same story as 250 threads). The incremental stressing of the interfaces can be observed by analysingT1 to T4, where typical latency in milliseconds gradually increases, and more high-latency outliers are introduced. In addition, the load testing and data collection routine was undertaken on different days, to ensure data used for analysis was void of identifiable anomalies (e.g. broadband issues). Notable observations and trends identifiable from the primary performance datasets (i.e. Figure 57, Figure 58, Figure 59, and Figure 60) include;

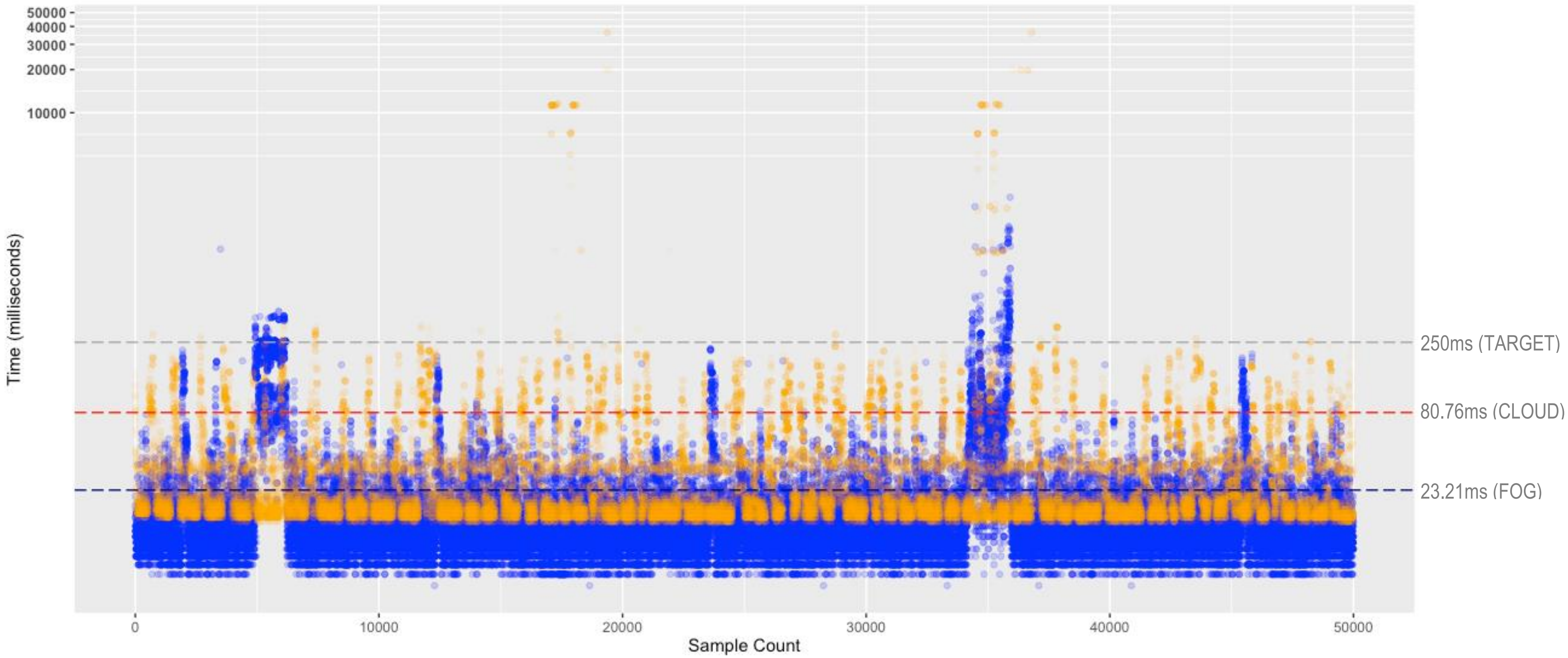
- **Average latency** recorded for the fog's cyber-physical interface outperformed the cloud interface for each load-testing scenario, with the performance gap being most evident for T2.

- **Performance outliers** for the cloud's cyber-physical interface are evident across all load-testing scenarios, with significant high-latency communications observed for T3 and T4. Although performance outliers for the fog interface are also evident, they are less common and consistent.

- **Performance consistency** of the cloud's cyber-physical interface appears more resilient to increasing numbers of concurrent connections, with normal operation (i.e. excluding outliers) demonstrating less variability and lower-latency than the fog interface. This trend can be best exemplified by T3, where the cloud's normal operation demonstrates much lower-latency.

- **Inherent capabilities** of fog and cloud cyber-physical interfaces contribute to performance degradation in different ways. The fog's local topology can deliver reliability (i.e. no observed failed communications), but with modest processing capability (e.g. maximum capacity for T3 and T4), while the cloud's dependency on external connectivity offers less reliability (e.g. outliers and communication failures from T1 to T4), but can deliver greater processing capabilities.

Although these high-level observations provide some performance insights, the following sections further investigate the performance datasets, to explore, analyse and compare the latency and consistency of the fog and cloud cyber-physical interfaces.



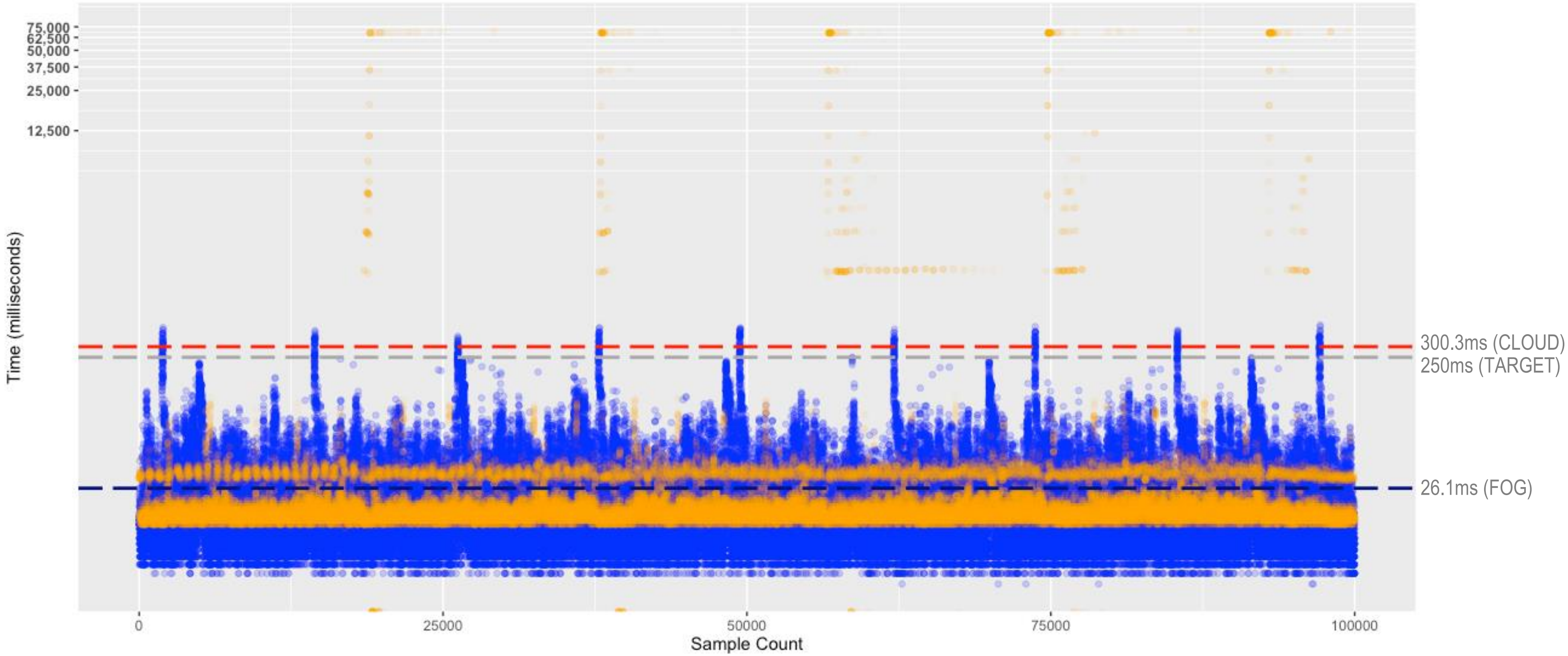

**Fog Analytics**

| Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
|---|---|---|---|---|---|
| 5.00 | 9.00 | 13.00 | 23.21 | 18.00 | 2571.00 |

**Cloud Analytics**

| Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
|---|---|---|---|---|---|
| 13.00 | 16.00 | 19.00 | 80.76 | 36.00 | 36300.00 |

**Figure 57 T1 results showing fog and cloud interface latency with 50 concurrent connections**



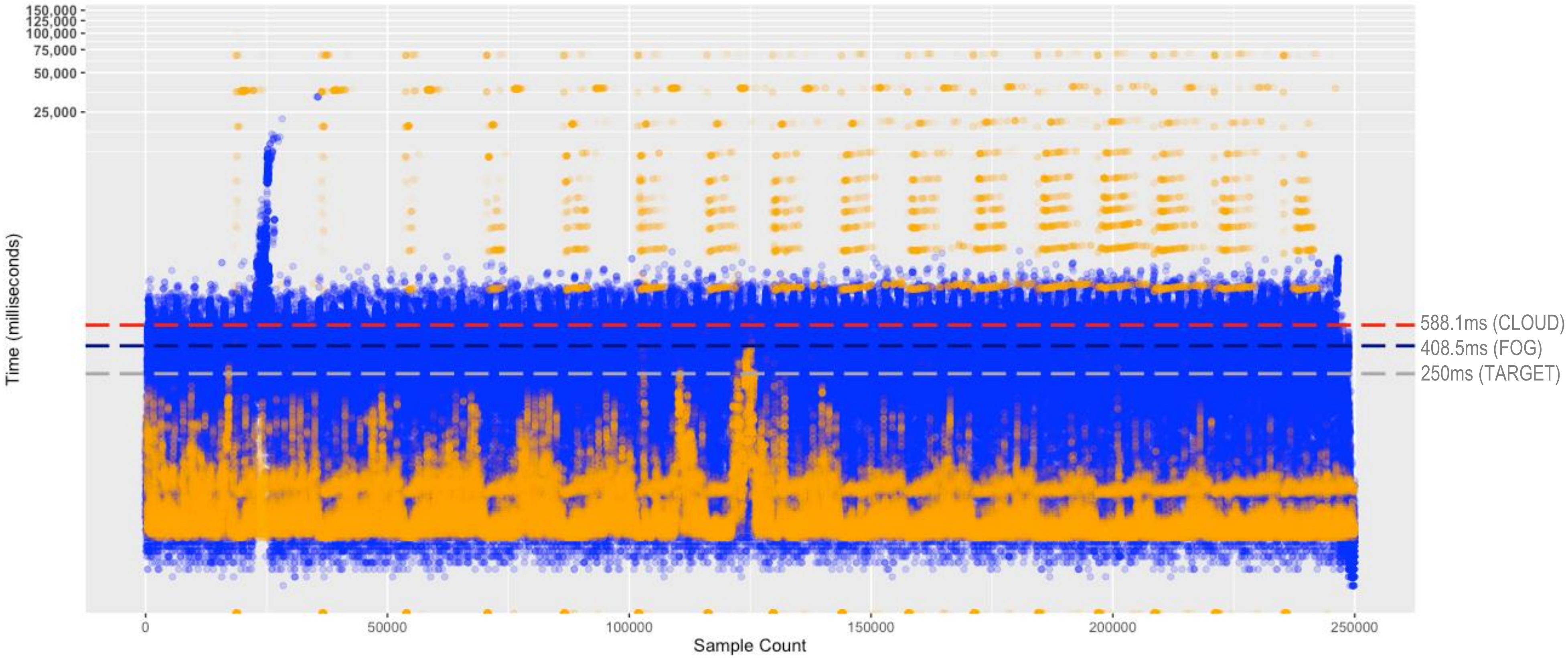

**Fog Analytics**

| Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
|---|---|---|---|---|---|
| 5.0 | 11.0 | 17.0 | 26.1 | 29.0 | 436.0 |

**Cloud Analytics**

| Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
|---|---|---|---|---|---|
| 0.0 | 16.0 | 17.0 | 300.3 | 20.0 | 70390.0 |

**Figure 58 T2 results showing comparison of fog and cloud analytics latency with 100 concurrent connections**



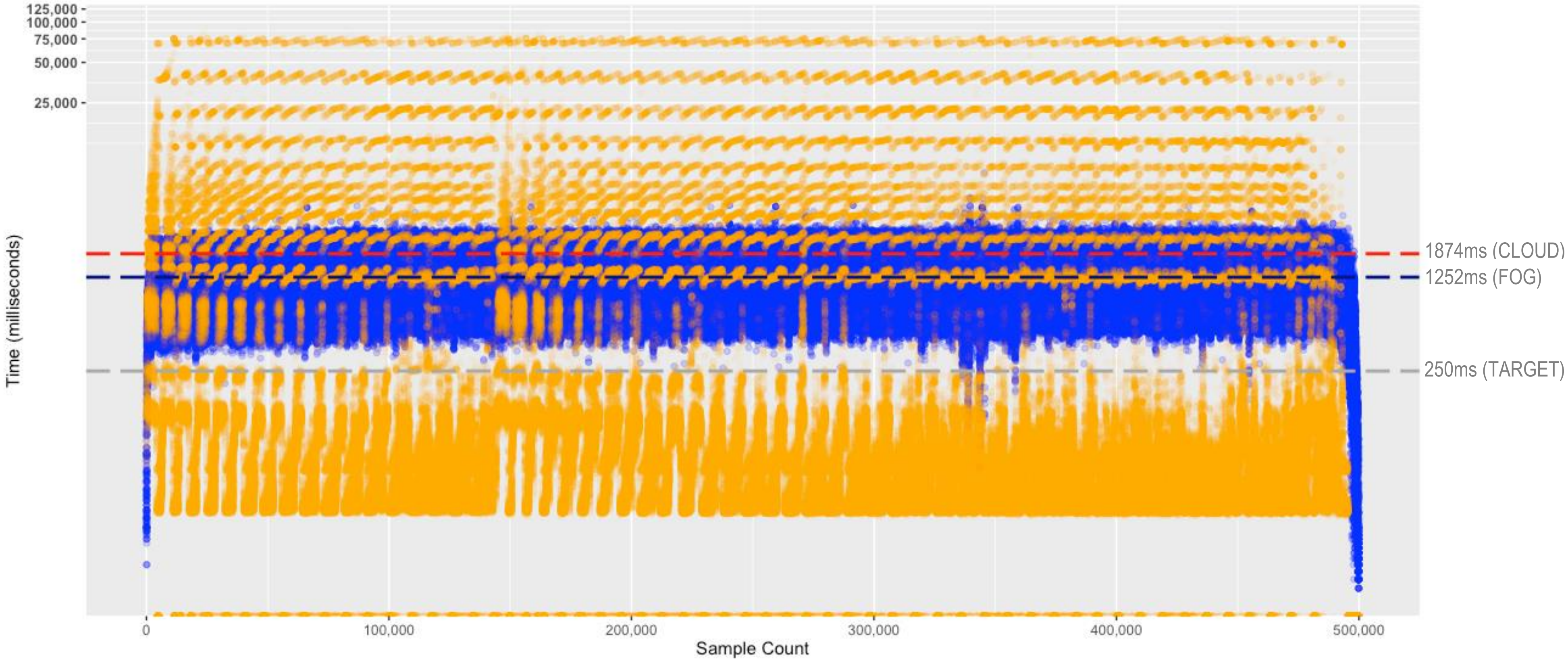

**Fog Analytics**

| Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
|---|---|---|---|---|---|
| 6.0 | 165.0 | 409.0 | 408.5 | 582.0 | 32800.0 |

**Cloud Analytics**

| Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
|---|---|---|---|---|---|
| 0.0 | 16.0 | 19.0 | 588.1 | 31.0 | 101400.0 |

**Figure 59 T3 results showing comparison of fog and cloud interface latency with 250 concurrent connections**




**Fog Analytics**

| Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
|---|---|---|---|---|---|
| 6 | 828 | 1288 | 1252 | 1602 | 7536 |

**Cloud Analytics**

| Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
|---|---|---|---|---|---|
| 0 | 26 | 55 | 1874 | 136 | 83370 |

**Figure 60 T4 results showing comparison of fog and cloud analytics latency with 500 concurrent connections**

### 5.2.1 Latency of interfaces

Given the deterministic and real-time characteristics of automation and control networks, the latency of cyber-physical interactions dictates the industrial engineering applications and scenarios the system may facilitate. This section employs descriptive statistics and visualisations to evaluate different aspects of performance latency observed from testing the fog and cloud cyber-physical interfaces.

- **Mean latency** describes the average performance observed for fog and cloud cyber-physical interfaces. Given this statistic utilises each measurements value to determine average performance, operational outliers can greatly influence the asserted latency.

- **Median latency** describes the performance measurement denoting the $50^{th}$ percentile of each dataset. Given this statistic focuses on the position of measurements rather than measured values, sporadic operational outliers do not significantly impact the asserted latency.

- **Interquartile latency range** describes the measurements observed within the middle 50% of each dataset. Given the skewed nature of the datasets (i.e. existence of outliers), the interquartile range provides an effective means of analysing the distribution and spread of measured latency, with lower interquartile ranges indicating less operating variability and stress.

Table 31 summarises the mean and median latency performance of fog and cloud interfaces for each execution scenario. The mean and median statistics for both interfaces are partitioned in two sections, with the best performing cyber-physical interface highlighted for each scenario and statistic. For example, the median difference between fog and cloud latency for the T2 scenario was 9.10 milliseconds, with the difference highlighted orange to indicate the cloud outperformed the fog. These statistics reaffirm previous assertions regarding average latency, with the fog interface outperforming the cloud across all execution scenarios. However, the cloud interface demonstrates superior performance for T2, T3 and T4 scenarios, when high-latency outliers are removed from the analysis (i.e. median measurement). In essence, this dichotomy demonstrates the inherent capabilities of both interfaces, with the fog's localised and embedded topology facilitating less high-latency measurement, and the cloud's superior compute capacity evident when sporadic high-latency performance outliers are negated.

Fog Analytics Cloud Analytics

| Scenario | Fog Mean | Cloud Mean | Mean Diff. | Fog Median | Cloud Median | Median Diff. |
|---|---|---|---|---|---|---|
| **T1 @ 50** | 23.21 | 80.76 | 57.55 | 13.00 | 19.00 | 6.00 |
| **T2 @ 100** | 17.00 | 300.30 | 283.30 | 26.10 | 17.00 | 9.10 |
| **T3 @ 250** | 409.00 | 588.10 | 179.10 | 409.00 | 19.00 | 390.00 |
| **T4 @ 500** | 1252.00 | 1874.00 | 622.00 | 1288.00 | 55.00 | 1233.00 |

**Table 31 Summary of mean and median latency performance**

Although mean and median statistics provide a single metric to gauge and compare cyber-physical interface latency, they abstract details relating to each interface's broader operating and latency patterns (e.g. upper and lower boundaries). Extrapolating a single statistic to broader operating ranges can be achieved using the interquartile range, which highlights performance measurements contained within the middle 50% of the dataset (i.e. measurements between the 25$^{th}$ and 75$^{th}$ percentile). In the context of this performance analysis, increases in the interquartile range are indicative of greater variability of each interface's middle operating range.

Table 32 presents the interquartile statistics for fog and cloud cyber-physical interfaces across each execution scenario. The table contains latency performance denoting the measurements positioned at 25% (i.e. 1Q) and 75% (i.e. 3Q), with the interquartile range calculated as difference (i.e. IQ = 3Q – 1Q). To simplify comparison of cyber-physical interfaces for each scenario, the last column (IQ Diff) shows the difference between the fog and cloud's interquartile ranges, with colour coding used to highlight the cyber-physical interface demonstrating the lowest operating range. The initial execution scenarios T1 and T2 show modest differences in operating ranges, with fog and cloud interfaces both recording the lowest interquartile range for T1 and T2 respectively. However, the cloud cyber-physical interface demonstrated greater resilience as more concurrent connections (i.e. T3 and T4) were applied, while the fog interface's interquartile range drifted significantly between 100 (T2) and 250 (T3) concurrent connections.

Fog Analytics Cloud Analytics

| Scenario | Fog (1Q) | Fog (3Q) | Fog (IQ) | Cloud (1Q) | Cloud (3Q) | Cloud (IQ) | IQ Diff. |
|---|---|---|---|---|---|---|---|
| **T1 @ 50** | 9.00 | 18.00 | 9.00 | 16.00 | 36.00 | 20.00 | 11.00 |
| **T2 @ 100** | 11.00 | 29.00 | 18.00 | 16.00 | 20.00 | 4.00 | 14.00 |
| **T3 @ 250** | 165.00 | 582.00 | 417.00 | 16.00 | 31.00 | 15.00 | 402.00 |
| **T4 @ 500** | 828.00 | 1602.00 | 774.00 | 26.00 | 136.00 | 110.00 | 664.00 |

**Table 32 Interquartile latency ranges of fog and cloud interfaces (milliseconds)**

As with the median statistic, the interquartile range utilises measurements at a position within the distribution, which naturally mitigates the effect of sporadic high-latency outliers. Therefore, the interquartile ranges can be considered beneficial for providing insights regarding potential operating boundaries, but do not indicate how consistently these operating ranges can be achieved. However, boxplot diagrams can be used to visualise both the interquartile ranges, and performance outliers. Figure 61 illustrates the segments of a boxplot, comprising a box denoting the interquartile ranges (i.e. 1Q, Median and 3Q), and upper/lower whiskers encapsulating the dataset. The boundaries for these whiskers are determined by multiplying the interquartile range by 1.5, with measurements outside this range considered operational outliers.

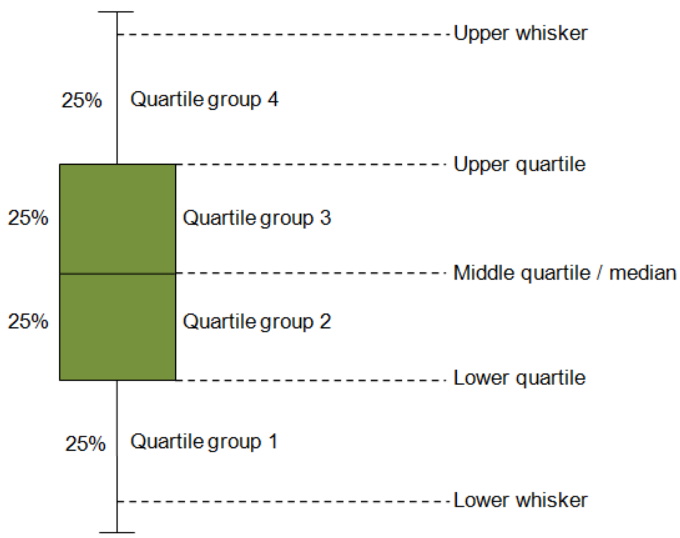



**Figure 61 Illustration of boxplot segments**

Figure 62 demonstrates changes in latency performance using the fog interface across all execution scenarios (i.e. T1 to T4) using the boxplot visualisation. This visualisation clearly shows the increase in normal operating ranges (i.e. interquartile range) from T1 to T4. To ensure these ranges are easily identifiable, the y-axis depicting latency in milliseconds uses a logarithmic scale. As previously reported in Table 32, the fog interface's interquartile operating range increases significantly from T2 to T3, but the boxplot visualisation also provides insights to operational outliers. These insights highlight operational outliers for T1 and T2 being exclusively high-latency (i.e. greater than normal operating range), but low-latency outliers materialise across T3 and T4 scenarios (i.e. less than normal operating range). This change in outlier patterns depicts the fog interface reaching capacity, with occasional high latency communications observed in T1, becoming the normal range of operation in T4.

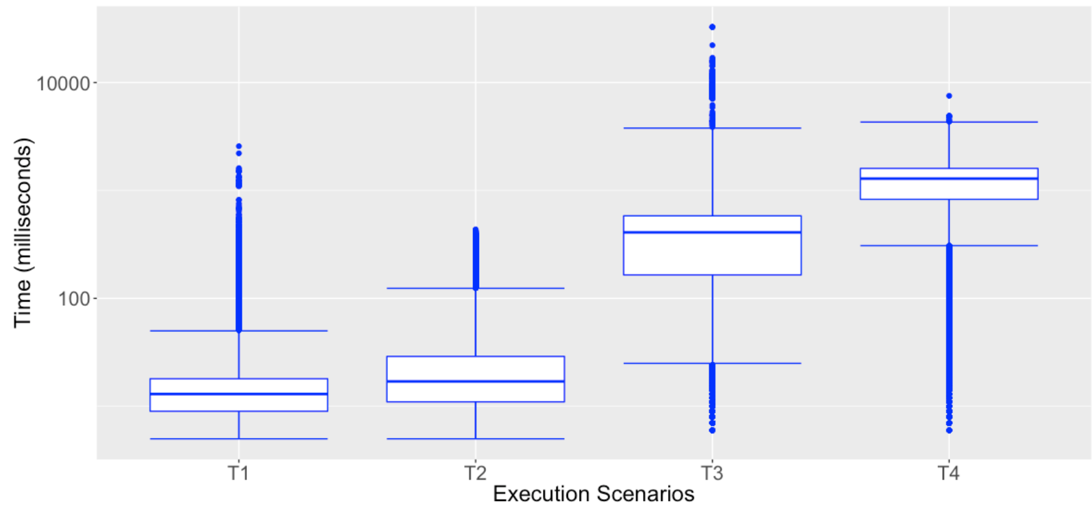



**Figure 62 Fog interface performance measurements**

Figure 63 demonstrates changes in latency performance using the cloud interface across all execution scenarios (i.e. T1 to T4) using the boxplot visualisation. As with the previous boxplot, the cloud interface latency measurements measured in milliseconds are displayed on the y-axis, which uses logarithmic scaling to clearly highlight the interquartile ranges for each scenario. The operating range consistency can be observed by the lack of spread in the interquartile range from T1 to T4, while the 1$^{st}$ quartile latency measurement remains at 16 milliseconds for T1 to T3. This demonstrates the cloud interface's lower operating boundaries (e.g. best case execution) were not significantly affected by additional concurrent connections, and while the interquartile operating range and 1$^{st}$ quartile measurements increase for T4, the deviation is modest when compared to those observed for the fog interface. However, additional outlier information highlights extreme outliers for all execution scenarios. Similar to the outlier pattern observed with the fog interface, outliers observed from T1, T2 and T3 execution scenarios become part of the normal operating range for T4. This highlights the interface's ability to respond to requests is gradually decreasing (i.e. longer requests are becoming normal), which is reinforced by the increased spread and variance in the normal interquartile operating range. Therefore, while the cloud interface demonstrated low latency execution, there are certainly concerns regarding the reliability and consistency of execution (i.e. performance drift due to outliers).

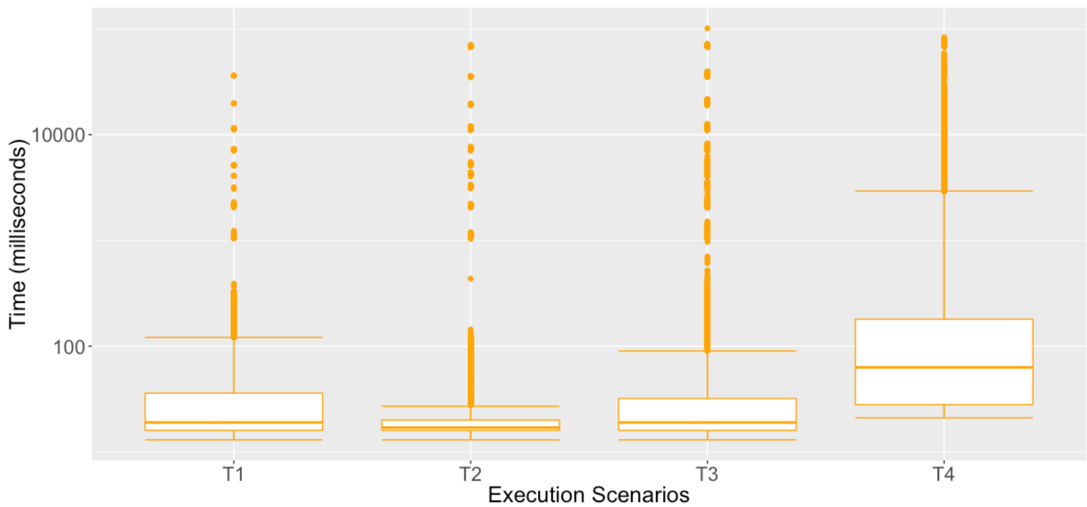



**Figure 63 Cloud interface performance measurements**

## 5.2.2 Consistency of interfaces

The consistency of interfaces focuses on determining communication reliability and resilience. While previous latency analysis evaluated normal operating ranges, there was no significant attention given to worst-performing, or failed communications. These negative attributes are important where industrial engineering or control applications possess stringent real-time constraints. To consider the consistency of each cyber-physical interface, the following sections present;

- **Maximum execution time** showing the worst-case observed request and response cycle for each execution scenario (i.e. T1 to T4).
- **Number of failed communications** recorded for each cyber-physical interface during the load testing experiments (i.e. no response).
- **Number of missed benchmarks** exceeding the 250 millisecond target utilised to simulate a time constrained Industry 4.0 scenario.

Figure 64 illustrates the maximum latency observed for fog and cloud interfaces across each execution scenario (i.e. T1 to T4), while Table 33 summarises the primary measurements, with differences highlighted to indicate the best performing interface for each scenario. Although maximum execution time may be considered a crude measure of consistency and/or reliability, observing the maximum execution time across multiple scenarios provides an indication of worst case execution. The findings show the fog interface recorded significantly lower maximum execution times compared to the cloud interface, with differences in maximum latency measured at 92.9% (T1), 99.4% (T2), 67.7% (T3) and 91.0% (T4). These significant differences in maximum latency may be attributed to the localised and embedded characteristics of the fog interface, which embodies less dependencies on network routing and external connectivity than the cloud interface.

Although not central to discussions relating to consistency, one pattern worth mentioning relates to the findings from T3, where the fog and cloud interfaces recorded higher maximum execution times at 250 concurrent connections, when compared to 500 connections (i.e. T4). Given the counterintuitive nature of these findings, one can only assume environmental factors (e.g. network traffic) impeded performance during the load testing for T3. However, fog and cloud interfaces were tested on the same day to ensure both interfaces were subject to the same environmental factors, which explains why both interfaces demonstrate increased maximum latency for T3. Such random fluctuations are difficult to avoid, and are representative of network intensive manufacturing and operational technology environments.

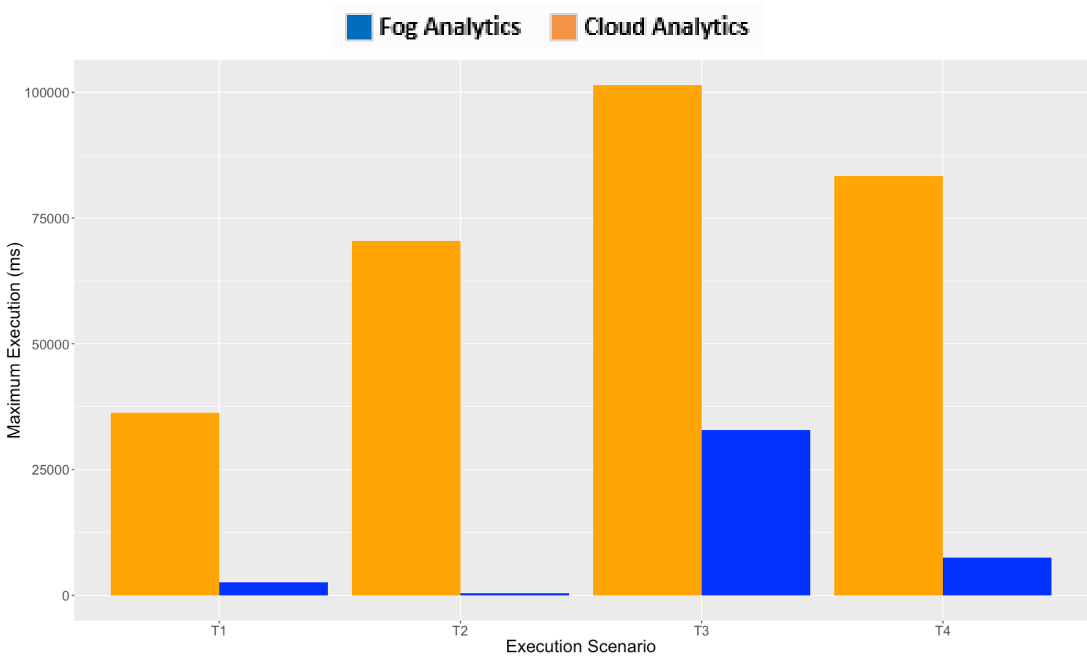



**Figure 64 Maximum execution times**


| Scenario | Fog Max. | Cloud Max. | Difference Max. | Difference Max. (%) |
|---|---|---|---|---|
| **T1 @ 50** | 2,571 | 36,034 | 33,463 | 92.9% |
| **T2 @ 100** | 436 | 70,397 | 69,961 | 99.4% |
| **T3 @ 250** | 32,798 | 101,401 | 68,603 | 67.7% |
| **T4 @ 500** | 7,536 | 83,370 | 75,834 | 91.0% |

**Table 33 Summary of maximum execution times for fog and cloud interfaces**

Figure 65 illustrates the percentage of failed communications (i.e. no response) for each execution scenario, while Table 34 summarises the number and percentage of these failures, with the best performing cyber-physical interface highlighted as the percentage difference of failure rates between the fog and cloud interface. Assuming the correct operation of technology components (i.e. hardware, software and network), these particular failure rates can be attributed timeouts from excessive loads being placed on the cyber-physical interfaces. Given these failures are recorded as zero milliseconds during load testing, failed communications can be easily identified and filtered from successful communications.

The fog interface did not record any failed communications for T1 to T4, successfully responding to 100% of requests for T1 (50,000 requests), T2 (100,000 requests), T3 (250,000 requests) and T4 (500,000 requests). In contrast, while the cloud interface demonstrated comparable failure rates for T1 (0%), modest failures were evident for T2

(0.11%) and T3 (1.42%), with more substantial failures observed for T4 (6.6%). These findings highlight the suitability of fog computing where consistent and reliable cyber-physical interactions are needed to support real-time engineering applications and scenarios (e.g. industrial control processes). Although cyber-physical cloud interfaces may facilitate industrial engineering systems and scenarios that are tolerant to occasional failures, those focused on Industry 4.0, real-time and self-optimising decision-making must naturally strive to minimise failed communications.

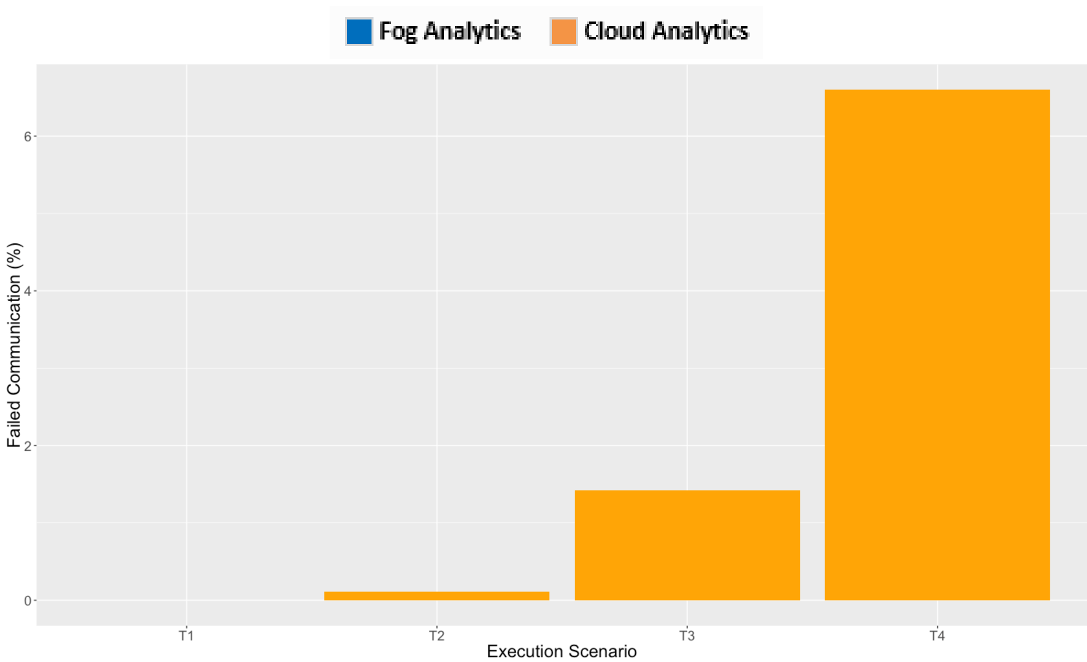



**Figure 65 Percentage of failed communications**


| Scenario | Fog (Failed) | Fog (% of All) | Cloud (Failed) | Cloud (% of All) | Failed Diff. (%) |
|---|---|---|---|---|---|
| **T1 @ 50** | 0 | 0.00% | 0 | 0.00% | 0.00% |
| **T2 @ 100** | 0 | 0.00% | 112 | 0.11% | 0.11% |
| **T3 @ 250** | 0 | 0.00% | 3,556 | 1.42% | 1.42% |
| **T4 @ 500** | 0 | 0.00% | 32,994 | 6.60% | 6.60% |

**Table 34 Summary of failed communications for fog and cloud interfaces**

Figure 66 Percentage of executions exceeding 250 milliseconds illustrates the percentage of executions that exceeded the 250 millisecond performance target, while Table 35 summarises measurements recorded for each execution scenario, with the best performing cyber-physical interface presented as the percentage difference of missed benchmark executions. Compared to previous analysis of consistency measurements (e.g. failed communications), the 250 millisecond benchmark used for this analysis is somewhat arbitrary. This particular time constraint was initially chosen to match the time interval used for testing, with the intention of highlighting when execution cycles began to overlap (i.e. another response was sent before receiving the previous request). However, the target execution time can be changed where engineering scenarios possess a particular time constraint.

The percentage of executions exceeding the 250 millisecond target for fog and cloud interfaces in T1 are practically equal, with the cloud interface exceeding the target on 380 occasions (0.76%), and the fog interface missing the target on 270 occasions (0.74%). Similarly, the differences observed for T2 are somewhat modest, with the cloud interface exceeding the target on 942 occasions (0.94%), and the fog interface missing the target on 407 occasions (0.41%). However, the additional loads applied by T3 and T4 greatly increased the number of execution cycles that exceeded the 250 millisecond target. Indeed, 66.85% of the fog interface's executions exceeded the target for T3, compared to 5.33% using the cloud interface. These trends continued for the T4 execution scenario, with 99.53% (fog) and 21.23% (cloud) of executions missing the 250 millisecond target. While previous discussions relating to consistency (e.g. failed communications) focused on worst performing measurements, the benchmark analysis demonstrates how performance consistency can also be evaluated against specific time constraints (e.g. expected response time for industrial process control).

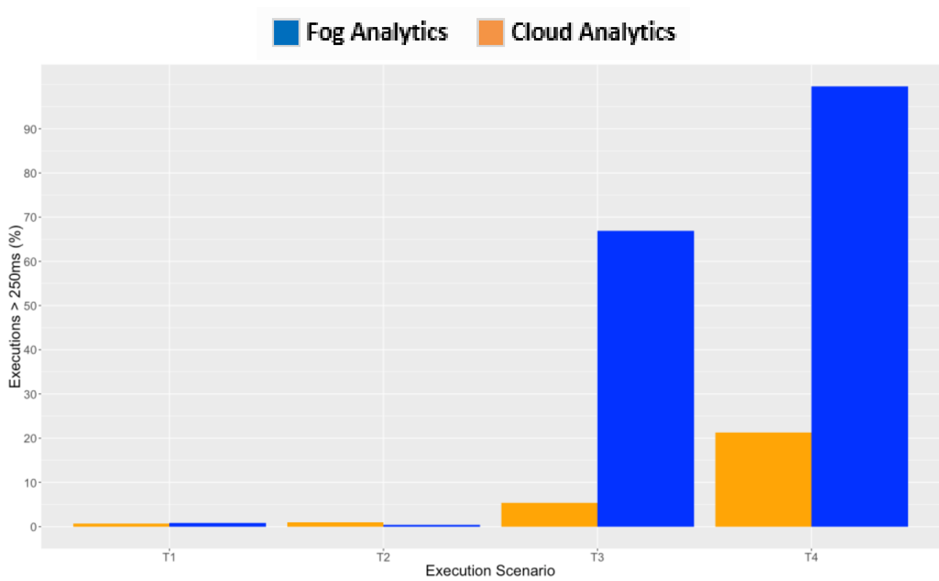



**Figure 66 Percentage of executions exceeding 250 milliseconds**

Fog Analytics Cloud Analytics

| Scenario | Fog (Missed) | Fog (% of All) | Cloud (Missed) | Cloud (% of All) | Missed Diff. (%) |
|---|---|---|---|---|---|
| **T1 @ 50** | 380 | 0.76% | 370 | 0.74% | 0.02% |
| **T2 @ 100** | 407 | 0.41% | 942 | 0.94% | 0.53% |
| **T3 @ 250** | 167,116 | 66.85% | 13,316 | 5.33% | 61.52% |
| **T4 @ 500** | 497,665 | 99.53% | 106,128 | 21.23% | 78.30% |

**Table 35 Executions exceeding 250 milliseconds for fog and cloud interfaces**

### 5.2.3 Summary of cyber-physical performance

The performance analysis of fog and cloud cyber-physical interfaces illustrates some of the strengths and weaknesses of both approaches. In particular, industrial engineering applications dependent on raw compute performance (e.g. execution of complex machine learning models) may benefit from interfacing with the cloud, while those applications demanding consistent and reliable real-time execution (e.g. minimise failed communications) may choose to interface using the fog paradigm. Of course, many engineering applications shall require a mixture of both compute latency and consistency to satisfy requirements, which may be addressed by altering the hardware and software architecture of the underlying cyber-physical platform.

Given the cloud interface utilised for load testing was based on public commercial cloud services, on-premises private cloud configurations could potentially reduce limitations relating to consistency and reliability. However, such configurations may negatively impact the computational capacity of the interface, with auto-scaling, on-demand provisioning and data accessibility offered by large datacentres being diluted by in-house alternatives. Similarly, the underlying implementation supporting the fog interface could also be augmented to address observed limitations relating to compute power and capacity;

- **Clustering** of embedded devices (e.g. Beowulf Cluster) enables processes to be distributed across multiple devices and executed in parallel, which increases the number of computational threads available to the request.

- **Load balancing** can be employed to distributed requests to different embedded devices based on particular criteria (e.g. round-robin, idle device status etc.). Although load balancing shares similarities with clustering, these requests are executed in isolation, rather than as part of one parallel task.

- **Technological advancement** shall logically increase the compute capacity of embedded and internet-of-things devices year-on-year (e.g. Moore's Law), and therefore, compute limitations associated with fog computing architectures are likely to dissipate in the medium to long-term.

Considering the Industry 4.0 design principles focused on real-time operations and decision-making, industrial cyber-physical systems supporting engineering and control applications must strike the balance between performance and consistency. Although cloud interfaces have been primarily used to support cyber-physical system implementations, there are genuine limitations regarding consistency, reliability and external risk factors (e.g. broadband downtime). While the fog interface also possesses limitations in terms of compute capacity, the extent of such limitations can be reduced through design, engineering and innovation. In contrast, the inconsistency of the cloud's real-time performance is asymptomatic of its underlying topology and architecture. Therefore, the decentralisation, flexibility and consistency offered by fog computing would appear suited to industrial cyber-physical systems supporting Industry 4.0 engineering application and scenarios.

# 5.3 Industrial analytics capability

While the previous performance analysis focused solely on communication latency and consistency, the capability analysis presented in this section measures the broader impact of the implemented industrial analytics architecture (e.g. data integration, interoperability etc.). This capability analysis employs a maturity model to measure, compare and benchmark industrial analytics capabilities before and after the system implementation. Maturity models have been used to assess readiness and operating capabilities for Industry 4.0 [149], energy management [155], risk management [156], business intelligence [157], and data quality management [158], to name a few.

The maturity model development process prescribed by De Bruin et al. [159] was used to design and build the model, which was given the name Industrial Analytics Maturity Model (IAMM). The development process consists of six sequential phases (Figure 67), with each phase characterising and shaping the model for its domain and application.

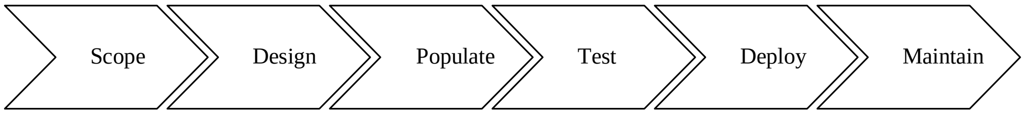



**Figure 67 Model development phases** [159]

### 5.3.1 Development of maturity model

**PHASE 1 - SCOPE**

The scope phase defines model boundaries using predefined criteria (Table 36), which includes the model's primary focus and stakeholders. A model's focus can be domain-specific or generic. Generic models are those that may be applied across different domains (e.g. quality), while domain-specific models are coupled to a particular scenario (e.g. software development). Groups of personnel or users that have a vested interest in the model's creation are known as stakeholders. These stakeholders typically inform the model's development, or benefit from its application. Common examples of stakeholder groups may include academia, practitioners, and government entities.

Table 36 shows that the IAMM's focus is domain-specific (i.e. embedded industrial analytics), with academic researchers and industry practitioners considered the primary stakeholders. These stakeholders can utilise the model to (a) illustrate current capabilities, (b) highlight areas for improvement, and (3) measure the impact of initiatives (i.e. positive/negative changes).

| **Criteria** | **Options** | **Selection** |
|---|---|---|
| Focus of Model | Domain Specific | ☑ |
| | General | |
| Stakeholders | Academia | ☑ |
| | Practitioners | ☑ |
| | Government | |
| | Combination | ☑ |

**Table 36 Scope criteria selection for IAMM**

**PHASE 2 – DESIGN**

The design phase defines the model architecture and application using predefined criteria (Table 37). These criteria provide a deeper understanding of (1) who will use the model, (2) why they need the model, and (3) how they can apply the model. A key aspect of the design phase was to manage the trade-off between domain accuracy and model simplicity. Although simple models may abstract particular nuances of the domain, complex models can create user adoption challenges (e.g. time-consuming assessment process).

The IAMM's audience was classified as internal executives and management, given these personnel are typically invested in developing better insights, and improving efficiencies using industrial analytics. A self-assessment method controlled by staff members was chosen to measure capabilities, which would be driven by internal requirements, roadmaps and objectives (e.g. Industry 4.0), while incorporating multiple applications, perspectives and dimensions (e.g. engineering, automation, technology etc.) to evaluate maturity.

| **Criteria** | **Options** | **Selection** |
|---|---|---|
| Audience | Internal Executives and Management | ☑ |
| | External Auditors and Partners | |
| Method | Self-Assessment | ☑ |
| | Third Party Associated | |
| | Certified Practitioner | |
| Driver | Internal Requirement | ☑ |
| | External Requirement | |
| Respondents | Management | |
| | Staff | ☑ |
| | Business Partners | |
| Application | Single Entity / Single Region | |
| | Multiple Entities / Single Region | |
| | Multiple Entities / Multiple Regions | ☑ |

**Table 37 Design criteria selection of IAMM**

A maturity model structure and application can take two forms. First, models may employ a multi-level approach. Such models adhere to the continuous maturity principle, where dimensions of the model may assert different levels of maturity (e.g. engineering could be Level 5, while technology could be Level 3). This approach is useful when modelling multifaceted domains, and identifying operational strengths and weaknesses. Second, models may also employ a single-level approach. These models adhere to the staged maturity principle, which employs a single label to classify overall maturity (e.g. manufacturing facility Level 4). This approach suits scenarios where natural linear progressions exist (e.g. beginner to advanced), and no details of composite capabilities are needed.

Given multiple disciplines and dimensions that exist for industrial cyber-physical systems and industrial analytics (Table 38), the IAMM's architecture utilises the dimensions of the industrial analytics lifecycle (Figure 42) to outline a multi-level approach. By using the primary dimensions of the industrial analytics lifecycle, the assessment process can compartmentalise criteria that directly affect different aspects of industrial analytics capabilities (e.g. industrial data integration).

| **Dimension** | **Levels** | **Rationale** |
|---|---|---|
| Open Standards | 10 | Standards-based technologies and protocols are needed to promote interoperability between different stages in the industrial analytics lifecycle. |
| Operation Technology | 10 | Operation Technology must support the systems and processes that facilitate the acquisition of industrial data in the factory. |
| Information Technology | 10 | Information Technology must provide the infrastructure and technologies needed to support the transmission and processing of data between different areas of the industrial analytics lifecycle. |
| Data Analytics | 10 | Data Analytics must possess the knowledge and skills necessary to model engineering problems that can be deployed in factory operations. |
| Embedded Analytics | 10 | Embedded Analytics must facilitate the deployment of data-driven models in the factory to affect real-time decision-making across operations. |

**Table 38. IAMM architecture and dimensions**

### PHASE 3 – POPULATE

The populate phase defines model components and subcomponents, which relate to different aspects of the domain's capability being assessed. Such components may be identified using numerous formal methods, such as literature reviews, stakeholder interviews, surveys, and case studies, to name a few. Given multi-dimensional maturity models for measuring industrial analytics capabilities do not exist in literature, the IAMM was populated (Figure 68) using knowledge derived through interactions with a large-scale industrial partner (i.e. DePuy Ireland). The IAMM contains dimension components - (1) Open Standards, (2) Operation Technology, (3) Information Technology, (4) Data Analytics, and (5) Embedded Analytics, and capability subcomponents (highlighted blue) that describe processes and technologies considered important to the maturity of each dimension.

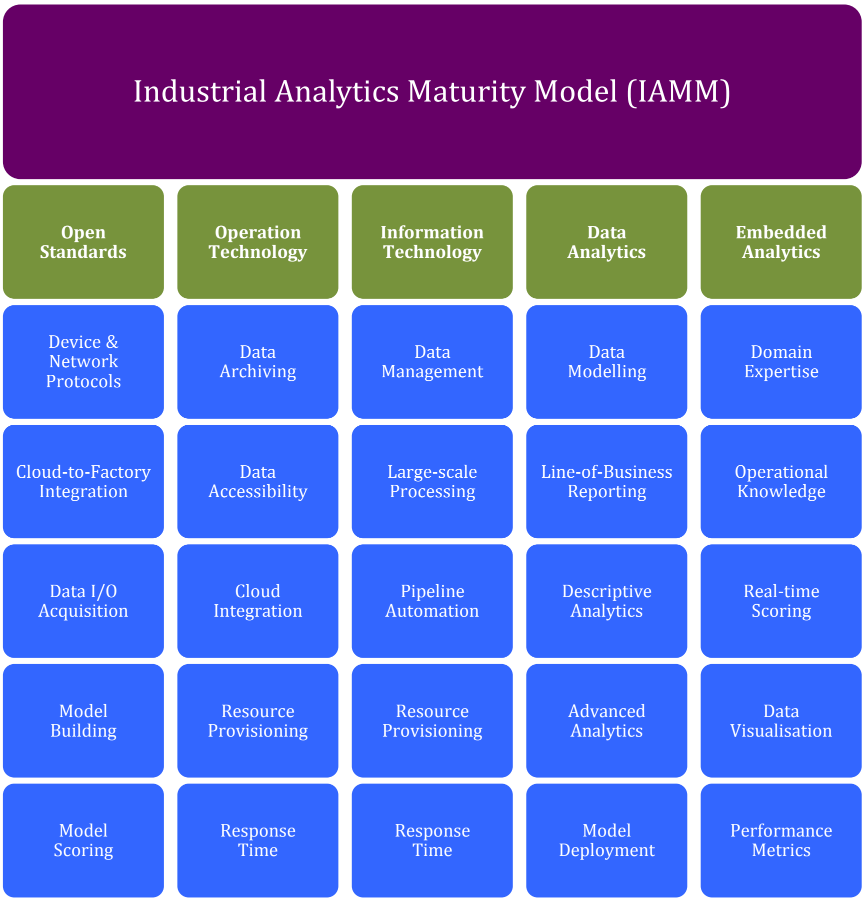



**Figure 68. Industrial analytics maturity model architecture**

To evaluate the maturity of each dimension's subcomponents, hypothesis statements (Table 39) are used to determine approximate truth. These types of statements enable practitioners to approximate maturity using an agreement scale (Yes=2, Partially=1, or No=0), which provides a simple method for scoring and measuring. In turn, the maturity level for each dimension can be derived by calculating the average score of its subcomponents.

| Code | Component | Hypothesis Statement |
|---|---|---|
| D1.1 | Devices & Network Protocols | Devices and instrumentation in the factory are accessed using open technology standards. |
| D1.2 | Cloud-to-Factory Integration | The factory floor is connected with cloud platforms using open technology standards. |
| D1.3 | Data I/O Acquisition | Archived operational data can be queried using standard I/O interfaces. |
| D1.4 | Model Building | Data-driven models are interoperable with other software, platforms and engines. |
| D1.5 | Model Scoring | Production-ready data-driven models are accessed and scored using standard protocols. |
| D2.1 | Data Archiving | All data points and measurements in the factory are archived in a central location. |

| | | |
|---|---|---|
| D2.2 | Data Accessibility | Archived data is labelled, catalogued, identifiable, and directly accessible. |
| D2.3 | Cloud Integration | Real-time operations utilize cloud computing for large-scale data storage, processing or analysis. |
| D2.4 | Resource Provisioning | New compute or technical resources are provisioned to support analytics efforts. |
| D2.5 | Response Time | Basic provisioning and support requests relating are fulfilled in 24 to 48 hours. |
| D3.1 | Data Management | Governance policies exist for cataloguing, storing, processing, and identifying data sources. |
| D3.2 | Large-scale Processing | Scalable and robust architectures exist to support exponential increases in data throughput. |
| D3.3 | Pipeline Automation | Manually data processing and cleaning routines have been automated using workflow pipelines. |
| D3.4 | Resource Provisioning | New compute or technical resources are provisioned to support analytics efforts. |
| D3.5 | Response Time | Basic provisioning and support requests relating are fulfilled in 24 to 48 hours. |
| D4.1 | Data Modelling | Data transformation, wrangling and preparation activities are undertaken using our own statistical tools and libraries. |
| D4.2 | Line-of-Business Reporting | Performance reporting and analysis is undertaken using productivity tools such as MS Excel. |
| D4.3 | Descriptive Analytics | Basic data relationships and patterns are identified in each month using statistical software packages. |
| D4.4 | Advanced Analytics | Predictive data-driven models are regularly built to inform decision-making. |
| D4.5 | Model Deployment | Accurate data-driven models are always deployed to provide end-users with access to the new knowledge. |
| D5.1 | Domain Expertise | Subject matter experts guide analytics investigations and questions relating to factory operations. |
| D5.2 | Operational Knowledge | Subject matter experts informing analytics efforts always possess an intimate knowledge of the process being investigated. |
| D5.3 | Real-time Scoring | Production-ready models are always deployed in the factory to positively impact real-time operations and decision-making. |
| D5.4 | Data Visualisation | Knowledge contained in models is presented to end-users in a manner that simplifies decision-making. |
| D5.5 | Performance Metrics | Top-line metrics are used extensively in embedded analytics applications throughout the factory. |

**Table 39 Industrial analytics maturity model assessment**

### PHASES 4 TO 6 - TEST, DEPLOY AND MAINTAIN

These phases relate to stakeholder feedback and model improvement protocols. The test phase engages stakeholders to determine if the model's architecture sufficiently represents the domain, while deploy and maintain phases utilises stakeholder feedback to continuously refine the model. Given the IAMM's design, structure and completeness originated from real-world requirements and analysis activities described during the implementation, further testing the model's alignment with the domain was not necessary. Therefore, these phases are somewhat superfluous to this analysis, but relevant to future work that should be undertaken by the industry partner.

### 5.3.2 Threats to model validity

The potential threats to the IAMM's validity can be classified as (1) those generally associated with maturity models, and (2) those stemming from model-specific design decisions. Both types of threats are described in Table 40.

| Threat | Discussion |
|---|---|
| Accuracy | Given IAMM focuses on approximating industrial analytics capabilities for comparison and benchmarking, accuracy was not considered a major issue. A greater threat relates to maintaining assessment consistency across longitudinal analysis. However, this challenge may be addressed by tightening assessment guidelines to ensure consistency between assessors, while also developing in-house quality policies to validate the integrity of assessments. |
| Scoring | There is a natural trade-off between model granularity and usability. High-level models may omit details to simplify assessment, while low-level models may come with significant overheads that impede usability. IAMM adopts somewhat of a hybrid perspective, whereby a complete architecture guides assessment, but simplified scoring facilitates adoption. However, these trade-offs may be adjusted in future iterations of the model based on real-world feedback. |
| Bias | All maturity models are subject to design bias. This cannot be avoided given the level of interpretation involved in the initial model construction. To reduce the potential for researcher design bias, the IAMM architecture was formed using multiple operational perspectives acquired through industry partner engagement. In addition, iterative refinement and practitioner feedback should also facilitate the dilution of any design bias overtime. |
| Coverage | Given the potential complexity of modelling an entire domain, maturity models typically address specific maturity characteristics. IAMM focuses on operational dimensions related to cyber-physical systems and Industry 4.0, including technology convergences and industrial analytics capabilities. However, the criteria proposed to measure these capabilities during the design phase may not provide appropriate coverage of the domain. Although such gaps are largely unavoidable for new maturity models, gaps in domain coverage can be addressed when refining the model. |

**Table 40 Summary of model validity threats**

### 5.3.3 Application of maturity model

This section describes the deployment and application of the IAMM to measure the impact of the cyber-physical system on the industry partner's industrial analytics capabilities. The impact was determined by undertaking capability assessments before and after the cyber-physical system implementation. In addition to highlighting changes strengths and weaknesses relating to industrial analytics capabilities, the application of the IAMM also demonstrated the usefulness of the model for benchmarking capabilities across departments and facilities.

### Assessment protocol

Figure 69 illustrates the assessment protocol used to measure industrial analytics capabilities. The diagram depicts assessment actions undertaken by three separate assessors in the outer section (i.e. score, reason etc.), before these actions were synthesised to determine capability scores for each dimension. The assessors comprised two academic researchers, and one automation engineer employed by the industry partner, which represent the intended domains of the IAMM (i.e. academia and industry). Table 41 summarises each step in this assessment protocol.

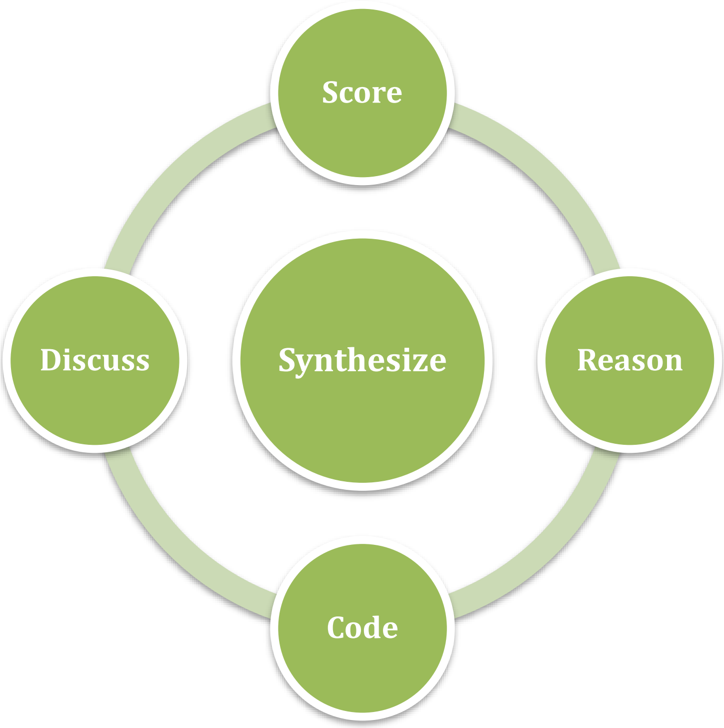



**Figure 69 Capability assessment protocol**

| Step | Description |
|---|---|
| Score | Each assessor evaluated and scored the hypothesis statements (Table 4) before and after the implementation of the cyber-physical system. |
| Reason | For each assigned score, assessors were required to provide a textual description rationalising and justifying their decision. |
| Code | In addition to a description, assessors were required to explicitly label the industrial analytics lifecycle model to highlight where the capability improvement was realised. |
| Discuss | After scoring, reasoning and coding components of the IAMM, assessors presented their assertions, and the validity of these assertions were evaluated by the group. |
| Synthesise | Finally, individual assessments were synthesised during group discussions to form the final capability levels (i.e. before and after implementation). These unified capability results are presented and discussed in the following sections. |

**Table 41 Capability assessment protocol**

Figure 70 illustrates the changes in industrial analytics capabilities, before and after the implementation of the industrial cyber-physical system. Although the facility's traditional engineering, control and automation systems were state-of-the-art, the capability assessment highlighted gaps relating to the development, management and deployment of industrial analytics models, and the adoption of Industry 4.0 design principles (e.g. open standards). These gaps are discussed in the following sections.

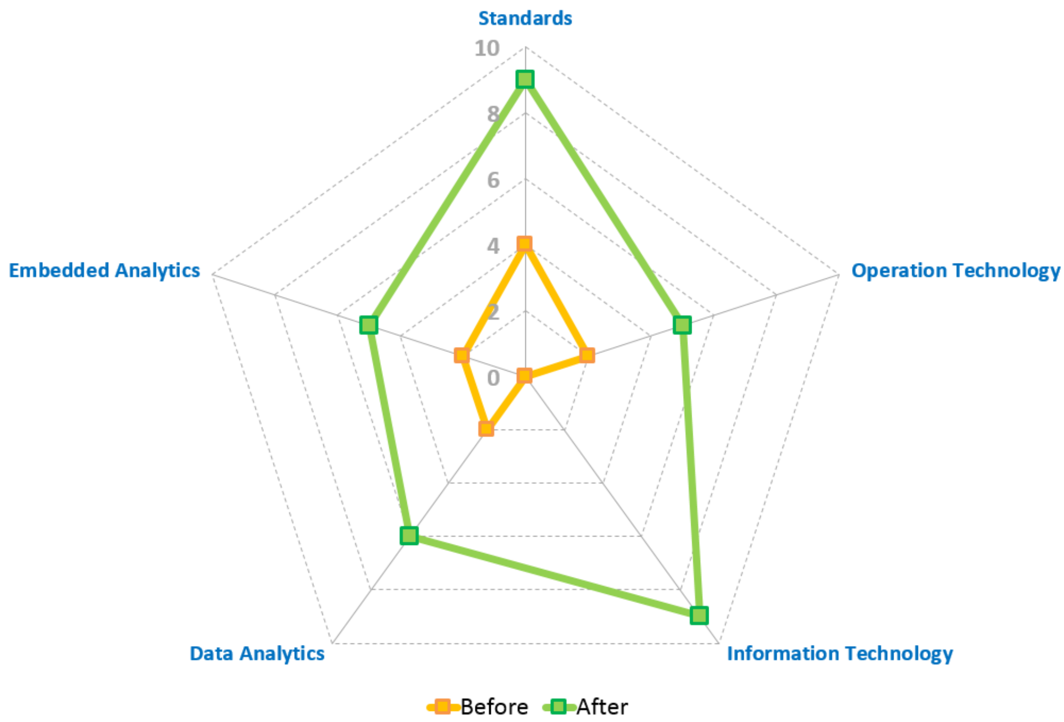



**Figure 70 Comparison of industrial analytics capabilities**

### INDUSTRIAL ANALYTICS LIFECYCLE

Figure 71 illustrates the industrial analytics lifecycle, with codes from Table 39 used to identify where particular improvements were observed. This lifecycle model depicts closed-loop data flows between multidisciplinary teams, which can be used to establish clear boundaries and responsibilities, and illustrate the primary data streams of importance to industrial analytics (i.e. batch and real-time streams). Batch streams are responsible for acquiring, cleaning and serving operational data to facilitate model creation and development, while real-time streams are concerned with embedding these models in real-world operations to inform timely decision-making (e.g. self-configuring industrial systems).

As previously mentioned, the codes overlaid (e.g. D1.2) on the industrial analytics lifecycle correspond to the IAMM's hypothesis statements (Table 39). These codes were attached during the assessment protocol, where assessors were required to explicitly highlight and rationalise assertions. The final codes presented indicate capability improvements were evident throughout most of the industrial analytics lifecycle.

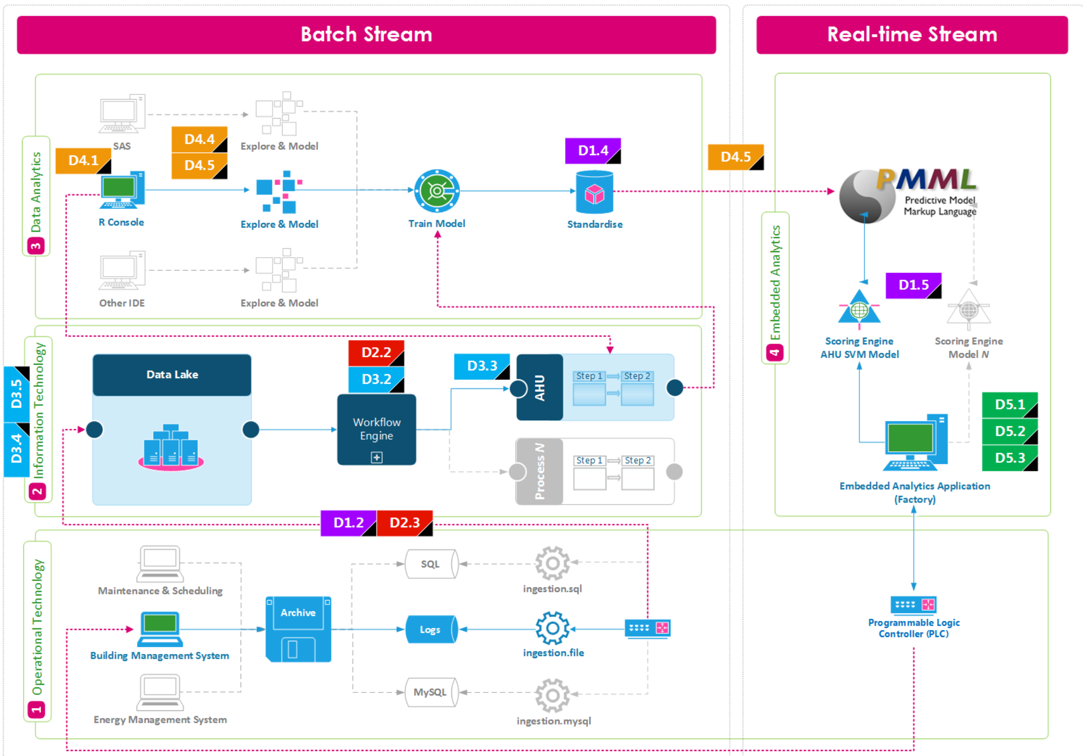



**Figure 71 Coded analysis of industrial analytics lifecycle**

### Open standards

Apart from operation technology (Figure 72), positive changes relating to standards were observed throughout the industrial analytics lifecycle. Although some standards were evident (e.g. OLE Process Control) for automation and controls, data and cloud integration was supported by proprietary commercial offerings, resulting in capability improvements for D1.2, D1.4 and D1.5. These particular improvements are discussed further in Table 42.

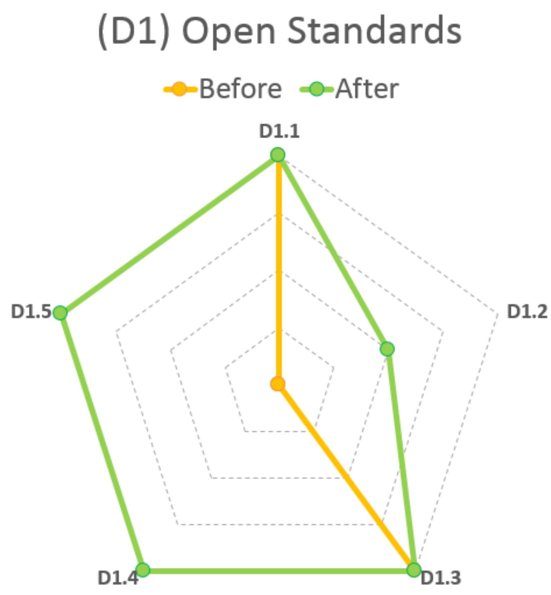



**Figure 72 Open standards comparison**

| Component | Rationale |
|---|---|
| D1.1<br>Devices & Network Protocols | Some open standards (e.g. OLE Process Control) are currently used for building automation and control, but the cyber-physical system implementation does not target improvements at this level. Therefore, no capability changes were expected or recorded. |
| D1.2<br>Cloud-to-Factory Integration | The industrial analytics lifecycle (Figure 71) shows Hypertext Transfer Protocol (HTTP) supporting factory-to-cloud integration. However, a cloud-based proprietary software library was used to support aspects of integration, and therefore the improved capability assigned was only deemed 'partial'. |
| D1.3<br>Data I/O Acquisition | OLEDB, ODBC and standard I/O streams could be used to access data repositories on the network. Similar to device standards, the cyber-physical implementation being assessed does not target improvements in factory-level I/O, and therefore, no capability changes were expected or recorded. |
| D1.4<br>Model Building | Most of the analytics observed were part of commercial offerings, and did not appear to adhere to any open standards or practices. Indeed, there was no obvious means of accessing the data or model outputs from these systems or applications. Given the cyber-physical implementation employs the Predictive Modelling Markup Language (PMML) to encode, share and deploy industrial analytics models, full agreement with the assessment hypothesis statement was deemed appropriate. |
| D1.5 | There was no evidence of industrial analytics models being deployed or used in |

| | |
|---|---|
| Model Scoring | real-time, and therefore no standards existed to support these operations. In contrast, the cyber-physical implementation utilises open web services to score PMML analytics models in real-time. These services are initiated using standard HTTP requests, while data exchanges are facilitated using machine-readable JavaScript Object Notation (JSON). Full agreement with the hypothesis statement was deemed appropriate given the comprehensive use of standards to support real-time model scoring. |

**Table 42 Open standards assessment**

### OPERATION TECHNOLOGY

The positive changes observed for operation technology largely stemmed from increases in data accessibility and integration with cloud computing technologies (Figure 73). Although these particular capability improvements were most evident for D2.2 and D2.3, descriptions for all assessment criteria are provided in Table 43.

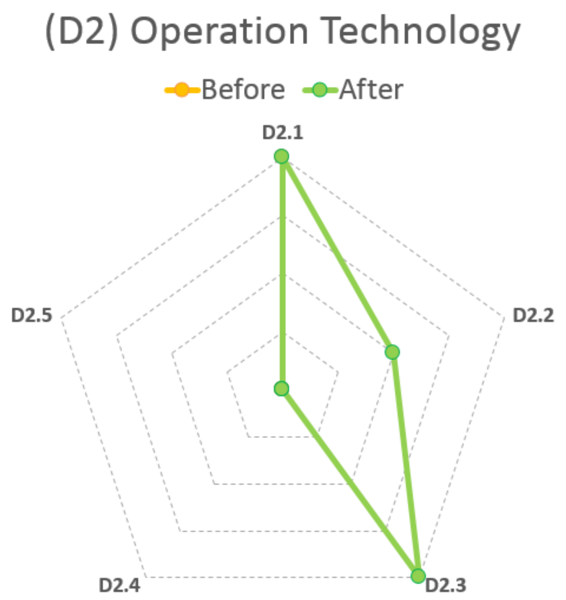



**Figure 73 Operation technology comparison**

| Component | Rationale |
|---|---|
| D2.1<br>Data Archiving | Given the cyber-physical implementation only interacted with the Building Management System (BMS), and these data points were already setup and configured to log every 24 hours, there were no obvious capability changes or improvements. |
| D2.2<br>Data Accessibility | The current data repositories exhibited arbitrary naming conventions, and were largely inaccessible (i.e. resided on isolated PC). Therefore, the cyber-physical implementation improved accessibility by using a workflow to contextualise operating data, and making analytics-ready data centrally available to end-users. |
| D2.3<br>Cloud Integration | There was no prior evidence of cloud integration, but the cyber-physical implementation integrated with cloud services to provide large-scale data processing capabilities. |
| D2.4<br>Resource Provisioning | No policies or processes existed to provision tools or technologies for industrial analytics activities. Given the internal and corporate nature of such |

| | |
|---|---|
| | capabilities, the cyber-physical implementation could not influence changes. |
| D2.5<br>Response Time | More general policies (e.g. technology requisition) for provisioning resources did not provide prompt turnaround times. Given the technical nature of the industrial cyber-physical system implementation, such capabilities were not directly addressed or affected. |

**Table 43. Operation technology assessment**

### INFORMATION TECHNOLOGY

Given few convergences existed between operation and information technology before the cyber-physical system implementation, many positive capability changes were observed (Figure 74). These improvements are discussed in Table 44.

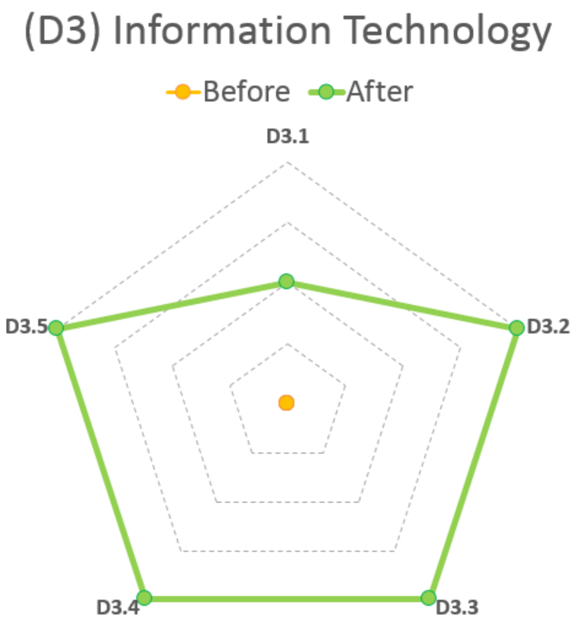



**Figure 74 Information technology comparison**

| Component | Rationale |
|---|---|
| D3.1<br>Data Management | While factory-level data repositories used arbitrary naming for data points, the implemented cloud repository (i.e. data lake) comprised tags describing the origin and application of the data. These tags formed a catalogue to identify data sources for mapping, cleaning and end-user lookups. |
| D3.2<br>Large-scale Processing | Due to the cyber-physical system's auto-scaling cloud configuration, data ingestion and workflow processes can scale to manage large datasets and interoperate with big data tools. |
| D3.3<br>Pipeline Automation | A formal workflow processes facilitated the turnkey cleaning and transformation of operational data, resulting in the automatic production of analytics-ready data for both end-users and third party systems. |
| D3.4<br>Resource Provisioning | The ability of cloud computing to deliver on-demand provisioning of compute and storage resources, meant the cyber-physical system was capable of realising seamless provisioning of services, which previously required internal personnel to undertake manual setup and configuration. |
| D3.5 | The on-demand and automatic provisioning enabled by factory-to-cloud |

| | |
|---|---|
| Response Time | integration facilitated instant response times. |

**Table 44 Information technology assessment**

### DATA ANALYTICS

The positive capability changes for data analytics were largely realised by automating the preparation of analytics-ready operational data, and facilitating the utilisation of different statistical software (Figure 75). These changes demonstrated capability improvements for D4.1, D4.3, D4.4 and D4.5. A broader discussion of these capability changes are presented in Table 45.

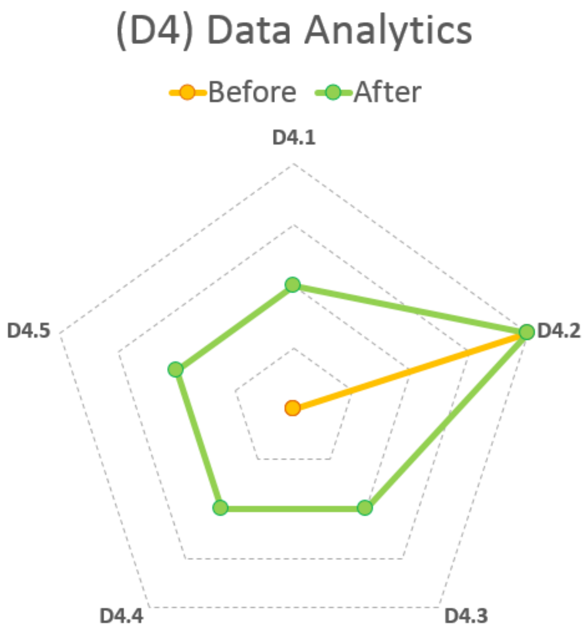



**Figure 75 Data analytics comparison**

| Component | Rationale |
|---|---|
| D4.1<br>Data Modelling | Existing information systems were used to monitor and report operational data, but there was no evidence of advanced analytics models (e.g. machine learning) being used to inform decision-making. However, the cyber-physical system facilitated model building using R Studio and open source libraries to build a simple issue identification scenario, which formed the basis for testing the system's underlying performance. |
| D4.2<br>Line-of-Business Reporting | The current infrastructure employed ad hoc analysis using MS Excel and MS SQL to build historic reports (e.g. last week/month etc.). Given the cyber-physical system did not target or affect line-of-business reporting, these capabilities remained unchanged.. |
| D4.3<br>Descriptive Analytics | The cyber-physical system provided access to analytics-ready operational data, which could be consumed using statistical software capable of reading CSV formats. Therefore, these capabilities were largely improved by the availability and accessibility of operational data from the workflow engine. |
| D4.4<br>Advanced Analytics | The cyber-physical system demonstrated advanced analytics capabilities by building, deploying and executing an issue identification model, using both |

| | |
|---|---|
| | cloud and fog computing architectures. |
| D4.5<br>Model Deployment | The cyber-physical system demonstrated the ability to deploy PMML encoded industrial analytics models to facilitate real-time scoring and operational decision-making. |

**Table 45 Data analytics assessment**

### Embedded analytics

The capability changes in embedded analytics stemmed from the ability to operationalise analytics models in the factory, enabling the real-time scoring of PMML encoded analytics models (Figure 76). This resulted in capability improvements relating to D5.1, D5.2, and D5.3. A broader discussion and rationale for these capability improvements are presented in Table 46.

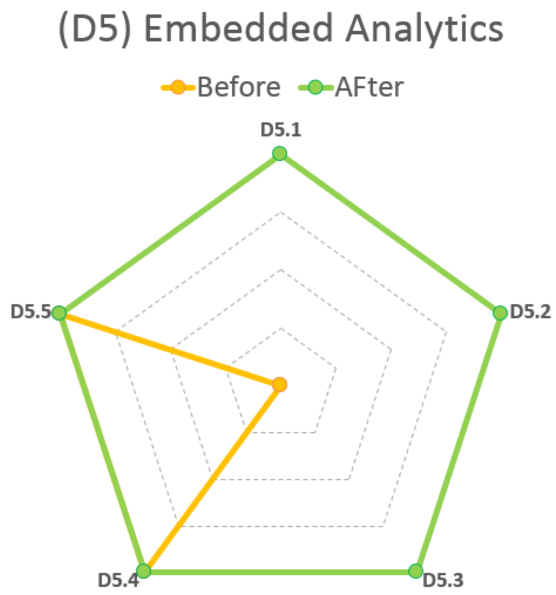



**Figure 76 Embedded analytics comparison**

| Component | Rationale |
|---|---|
| D5.1<br>Domain Expertise | The ability to incorporate subject matter expertise was facilitated by the industrial analytics lifecycle, where engineering knowledge relating AHU diagnostics was used to construct and encode an analytics model with PMML, which was later utilised to measure cyber-physical system performance. |
| D5.2<br>Operational Knowledge | The observed analytics teams were comprised of individuals with backgrounds in statistics, maths, and physics, with an under appreciation for engineering and domain knowledge pertaining to factory operations. Given the multidisciplinary design methodology and implementation strategy underpinning the cyber-physical system, there is a natural focus on the integration of operational and domain knowledge with technology. |
| D5.3<br>Real-time Scoring | The operationalisation and real-time scoring of industrial analytics models did not form part of existing operations. After the cyber-physical system implementation, software agents embedded in the factory possessed the ability to obtain continuous data streams from PLC's, and score PMML encoded analytics models in real-time. |

| | |
|---|---|
| D5.4<br>Data Visualisation | A couple of information dashboards and software systems in the factory reported historical operational data (e.g. energy usage). However, the cyber-physical system did not extend these capabilities, and therefore, data visualisation capabilities were unaffected. |
| D5.5<br>Key Performance Metrics | Existing operational teams employed metrics to gauge performance and efficiencies (e.g. production output to energy utilisation). Given the cyber-physical system did not enhance or inform these metrics, there was no observed capability change. |

**Table 46 Embedded analytics assessment**

### 5.3.4 Summary of capability assessment

There are many challenges associated with developing and improving industrial analytics capabilities. Common challenges include managing heterogeneous technologies and platforms, forming multidisciplinary teams, and defining prescriptive approaches, to name a few. These challenges are exacerbated further where no methods exist to evaluate current capabilities, or strategically identify areas for improvement (e.g. technical roadmap). Thus, the IAAM maturity model was designed to measure the impact of the cyber-physical system on industrial analytics capabilities. The results presented in this chapter showed positive improvements across several maturity dimensions, which should be expected given the lack of analytics operations or infrastructure before implementation. The measured improvements for industrial analytics capabilities are outlined below;

- **Open standards** were previously neglected in favour of proprietary commercial technologies, which reduced the interoperability and accessibility needed to support Industry 4.0 scenarios. In contrast, the demonstrated implementation introduced open technology standards to support data encoding (e.g. JSON) and transmissions (e.g. HTTP), meaning communication and integration with other systems and services could be realised more efficiently.

- **Factory-to-cloud integration** was not evident during the initial capability assessment, with corporate virtual machines chosen to provide additional compute capacity. However, this strategy suffered from slow provisioning and approval processes, while scaling and altering configurations were manual endeavours. The demonstrated implementation leveraged the public cloud to overcome these challenges, and deliver a more seamless analytics pipeline to support data preparation, exploration and model building.

- **Real-time industrial analytics** using advanced predictive models (e.g. machine learning) were not used to inform timely decision-making, but some commercial software systems (e.g. BMS and MES) provided summary statistics and metrics to relay operating states. The implementation provided an automated analytics pipeline to build predictive models, and delivered the infrastructure enabling model deployment to the factory. This provided the facility with an open, efficient and streamlined process for embedding real-time industrial analytics.

The value of capability assessments becomes apparent over time. Therefore, capability assessments should not be considered isolated events, but rather as something that belongs to a longitudinal process that continuously monitors, improves and compares capability levels. In turn, this process naturally produces benchmarks for comparing capabilities across departments and facilities, while also demonstrating improvements and advancements to management. Although the IAMM provides a foundational framework to support capability assessment, facilities should extend and customise the model's architecture as needed. The proposed model does not proclaim to capture every dimension and aspect of industrial analytics, and therefore, refinements and extensions are encouraged to enhance representation of the domain.

## 5.4 Chapter conclusions

This chapter presented results pertaining to the real-time performance of cyber-physical interfaces using fog and cloud computing approaches, and an evaluation of the broader industrial analytics capabilities (e.g. interoperability, integration etc.) from the cyber-physical system implementation. These results served to highlight many positive changes relating to open standards, data processing, and real-time scoring, while also discussing the merits of fog computing as an implementation approach for industrial cyber-physical systems.

# Chapter 6
# Thesis Conclusions

## 6.1 Chapter introduction

This chapter presents the concluding remarks and insights from this research. The following sections contemplate how the characteristics and operations of the industrial partner informed the researcher's perspective of real-world smart manufacturing adoption efforts, and how this research identified and addressed theoretical and technical gaps that were observed within this real-world environment.

This thesis contributes to knowledge through the presentation of theoretical and technical artefacts that support the design and implementation of industrial analytics solutions in the context of Industry 4.0. The individual contributions presented include the (1) *unified design methodology* to provide multidisciplinary teams with a formal and consistent design process, (2) *industrial data pipeline* to automate industrial data integration and cleaning to reduce the effort and complexity associated with data preparation, and (3) *industrial cyber-physical system based on fog computing architecture* to deploy and embed machine learning models within real-time factory operations.

## 6.2 Insights from industrial collaboration

This research commenced at a time when awareness and overall interest in smart manufacturing and Industry 4.0 was modest compared to present day. Due to the limited body of knowledge at that time, many of the industrial partner's smart manufacturing initiatives appeared to follow technology-first approaches, without thoroughly considering domain requirements, formal methods, or industry standards that may support Industry 4.0 adoption. Thus, many aspects of this thesis were guided by the idea that Industry 4.0 efforts should embrace guidelines and requirements, which directly influence technology development, selection and deployment using quantitative and qualitative measurements. Table 47 describes some of the main internal perspectives and approaches observed through industry engagement, with research notes/commentary that consider how some of these perspectives may conflict with Industry 4.0 principles and operations.

| Dimension | Internal perspective and assumptions | Researcher's perspective and observations |
|---|---|---|
| Big Data | • Equipment throughout the factory consistently logs data<br>• Integrating with operational systems and data sources is not complex<br>• Operating insights can be derived by placing many datasets in a single location | • Data logging and integration tasks were generally overlooked, but consistently presented the greatest impediment to delivering internal analytics and reporting projects.<br>• Oversimplifying the process of deriving operational insights may be attributed to the background of the personnel that formed the analytics teams.<br>• Given most of the industrial data sources encountered onsite were less than a couple of Gigabytes in size, the pressing need for big data systems and pipelines seems premature. |
| Cloud Computing | • Public cloud computing providers will not be permitted by corporate governance<br>• Private cloud computing platforms are equivalent to public offerings | • Internal technology personnel were not well-informed about the public cloud (e.g. AWS)<br>• Choosing to ignore the auto-scaling, managed services and technical support provided by public cloud providers could greatly increase the cost of producing Industry 4.0 infrastructures and applications. |
| Centralised Database | • Time-series data generated by equipment and processes from across all of the organisation's manufacturing facilities should be stored in a single centralised relational database<br>• Technology personnel promoted the idea that existing relational databases in the facility were both infinitely scalable and real-time capable<br>• Existing information and operation technology teams did not believe any additional skills or knowledge were required to facilitate smart manufacturing operations | • Choosing to employ a centralised database to support smart manufacturing operations across geographically distributed sites would appear to violate Industry 4.0 design principles relating to decentralisation, while also introducing many practical problems related to reliable real or near-time communications.<br>• Critical questions relating to upper operating capacity, fault tolerance and scaling strategies were largely ignored, which is likely to result in the accumulation of technical debt.<br>• Discussions relating to data persistence, querying and processing did not employ definitive use cases and operating scenarios to inform decision-making. |
| Data Analytics | • Current factory operations employ advanced data analytics<br>• Existing industrial systems include the analytics needed for smart manufacturing<br>• Any future gaps in analytics capabilities can be overcome by off-the-shelf software<br>• Confusion between visualisation, summary statistics, and predictive analytics | • A shared and common understanding of data analytics does not exist between teams, which can create some friction when agreeing technical roadmaps, software adoption etc.<br>• Recruitment of analytics personnel appeared to be biased towards those with maths and physics backgrounds, while overlooking those with engineering and computing subject matter expertise. The potential issues with such policies would appear to be (a) finding problems to solutions that already exist (e.g. computer vision for identifying defect parts), and (b) not possessing the skills/knowledge to operationalise the analytics models. |
| Embedded Analytics | • Not currently focused on applying analytics models to real-time operations | • Overlooking the importance of embedded analytics shall naturally reduce the return-on-investment of data analytics efforts.<br>• By not ensuring data and embedded analytics efforts can converge, the development and accumulation of knowledge from data analytics has poor utility in the context of smart manufacturing operations. |
| Openness & Integration | • Technical and analytics efforts by different teams will converge naturally<br>• Current technical and analytics outputs are 'easily' extended/changed<br>• Challenges that arise in the future can be solved through software purchases<br>• Industrial operations are based on open and consistent standards | • Technical and analytics initiatives relating to smart manufacturing adoption were generally oversimplified, with many important design and implementation decisions overlooked in favour of reported progress.<br>• Although some standards were evident, there was no strong commitment to encourage openness and standards across technical or analytics projects.<br>• Proprietary commercial software and systems would appear to dominate the facilities decision-making. |

**Table 47 Primary research notes and observations that influenced research efforts**

The initial research direction set by the industry partner focused on the use of centralised cloud and big data platforms to store, process and analyse data for smart manufacturing operations. Although centralising operational data may be beneficial for many scenarios (e.g. data warehousing, reporting etc.), pursuing an entirely centralised approach for smart manufacturing appeared to conflict with several design principles associated with Industry 4.0 (e.g. decentralised operations and real-time decision-making). However, such conflicts were not identified by internal teams given these initiatives were not subject to feedback or acceptance testing aligned with these principles, which naturally promotes poor technology alignment and/or subjective success criteria. While one could not say with absolute certainty why internal teams resisted testing assumptions or measuring feedback using relevant methods, these approaches could be driven by concerns or uncertainty relating to interdisciplinary topics, and/or inadequate understanding of emerging Industry 4.0 design principles or requirements.

In the context of the operating teams encountered during this research, there was no significant agreement or universal understanding of the technologies relevant to smart manufacturing, or how these technologies will ultimately be applied to real-time factory operations. Although some consensus (e.g. technology preferences) was evident between different teams, individual disciplinary bias invariably contributed to debate around roadmaps, milestones and prioritisation. Indeed, insufficient interdisciplinary expertise and collaboration could prove to be one of the most significant barriers to smart manufacturing adoption within the facility. Being aware of such potential barriers and conflicts provided this research with the opportunity to define formal methods that may unify some of these disparate perspectives, and demonstrate an approach for developing and deploying predictive machine learning applications under the umbrella of an industrial cyber-physical system, while incorporating relevant Industry 4.0 design guidelines and best practices.

Some of the internal perspectives promoting the notion that plug-and-play solutions will eventually be used to enable smart manufacturing operations would appear to be unrealistic. As the field matures there will invariably be many engineering and technology building blocks, but facilities must also appreciate that much of the promise of smart manufacturing relates to the discovery of new knowledge and methods, which in many instances will be derived from their own proprietary processes. Hence, to achieve these particular competitive advantages facilities must still commit to managing their growing operational data, training and upskilling personnel, and implementing appropriate infrastructures, technologies and communications for Industry 4.0.

## 6.3 Impact of primary research contributions

Given the aforementioned state of Industry 4.0 initiatives centred on engineering informatics, the primary high-level objectives of this research centred on producing interdisciplinary guidelines to support the design and development of industrial analytics systems to enable industrial machine learning applications. Figure 77 highlights the design and implementation dimensions for industrial analytics and cyber-physical systems that were impacted by the contributions from this thesis. The following sections summarise and conclude how these contributions facilitated changes to design and implementation approaches.

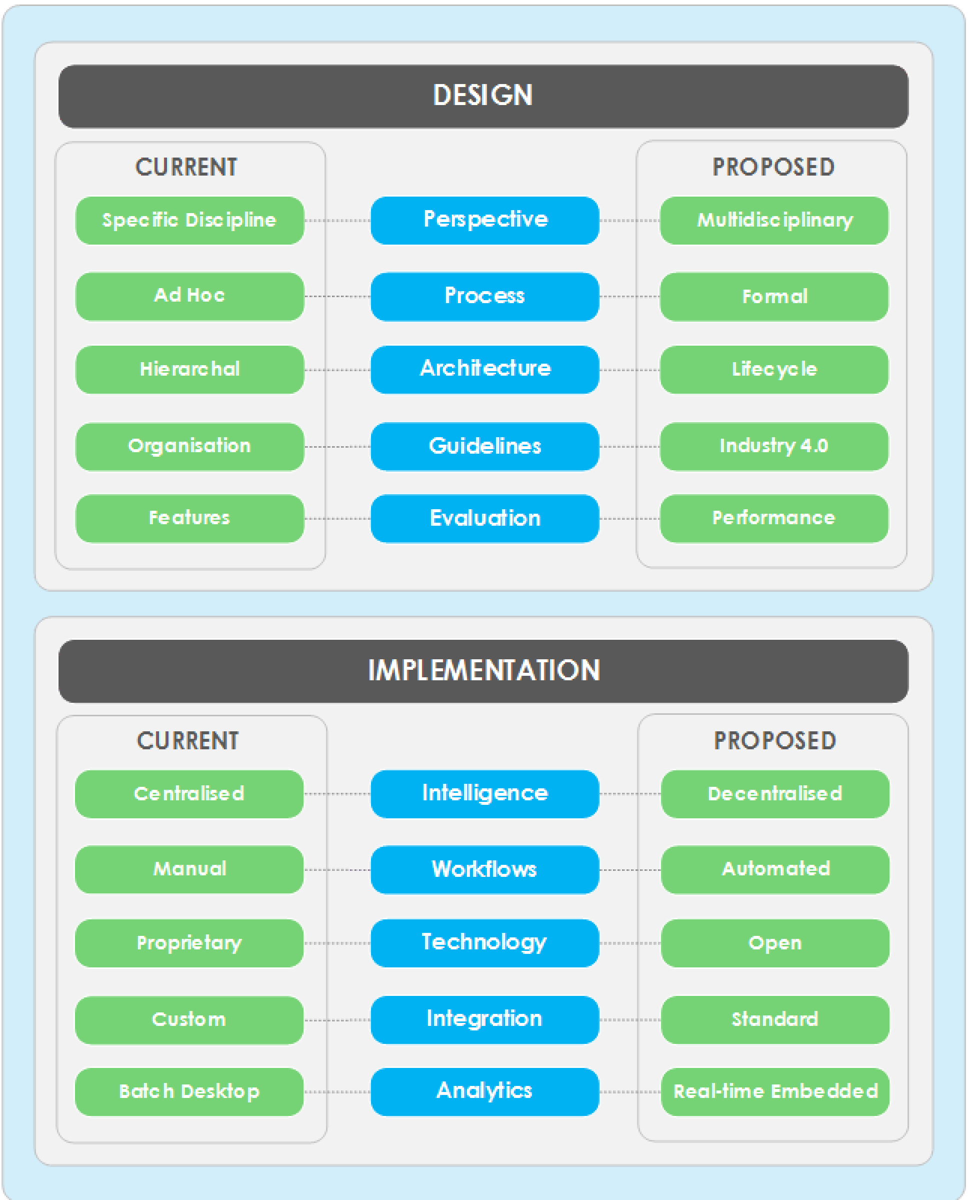



**Figure 77 Comparison of current and proposed industrial analytics approach**

### 6.3.1 Industrial analytics design methodology

A unified design methodology to promote the development of industrial analytics capabilities was presented as one component of this thesis. This design methodology promoted a bottom-up approach for *designing* information systems and architectures to support data-driven Industry 4.0 applications, and provided engineering and technology teams with a shared perspective of the industrial analytics lifecycle. The proposed lifecycle created an awareness of the distinct roles and data flows pertaining to factory operations, while also highlighting pertinent acceptance criteria in the form of Industry 4.0 design principles, corporate stakeholder concerns and functional requirements.

### PERSPECTIVE

The perspective dimension refers to design concerns of those developing the industrial analytics infrastructure, with current approaches embodying design perspectives from specific disciplines (e.g. technology, process engineering etc.). Given the teams focused on Industry 4.0 initiatives were comprised of qualified and experienced personnel, the overriding perspectives of each team tended to closely follow the discipline-specific perspective held by the majority. This type of dynamic proved largely counterproductive when trying to establish facility-wide viewpoints with regard to progressing Industry 4.0 and industrial analytics capabilities. However, the development and application of the unified design methodology presented in this research provided internal teams with a shared multidisciplinary perspective for implementing industrial analytics and cyber-physical systems. After creating, presenting and applying the unified design methodology, existing operational teams were introduced to the conceptual industrial analytics lifecycle model to promote multidisciplinary perspectives, which served to;

- **Classify primary teams and disciplines.** Although teams were aware of each other's existence, there were no prominent methods for information sharing or collaboration. In addition, some teams possessed the freedom to redefine themselves in response to trends, which invariably resulted in unintended overlaps and duplication between these teams (e.g. multiple 'big data' teams emerged during the first 12 months of this research). To prevent these overlaps from occurring in the future, the lifecycle model classified the primary teams and disciplines needed to deliver industrial analytics in the factory, which provided each team with an opportunity to commit to fulfilling a particular role based on their domain expertise, and encouraged teams to collaborate rather than address every problem (e.g. engineering, big data, machine learning etc.).

- **Highlight logical connections between teams.** In addition to classifying teams and disciplines, the lifecycle model highlights logical relationships and connections between systems and processes managed by each team. These connections depict how data flows throughout the factory, while also identifying technical components needed to bridge gaps in current connectivity, integration, contextualisation and interoperability. This enabled each team to be more aware of the teams they should be collaborating with to realise a more cohesive and unified approach to industrial analytics, and emphasised to some extent how changes to existing systems and processes may impact other teams.

- **Outline primary responsibilities for each team.** Although most personnel were aware of which teams were responsible for traditional systems and processes (e.g. energy management system), much ambiguity and disagreement was evident when discussing contemporary information systems and processes (e.g. big data processing). Thus, many teams appeared to be overly ambitious regarding the breadth of technology and analytics challenges they were addressing, which one could argue negatively impacted the demonstrated and quantified outputs from these teams. The lifecycle model provided some boundaries regarding the responsibilities of each team, with the intention of encouraging cross-team collaboration, and providing guidelines to determine which teams should take ownership of contemporary systems and processes as they emerge in the factory.

#### PROCESS

The process dimension relates to the underlying methods and approaches informing the design of systems delivering industrial analytics for Industry 4.0, with current approaches adopting ad hoc practices and procedures, while the proposed approach promoted the idea of formal and systematic methods. Based on interactions with the industry partner's internal teams, there were few instances of formal or systematic approaches supporting the design or implementation of Industry 4.0 information systems. The formal processes adopted by some teams (e.g. Lean or SCRUM) were general-purpose process models, and therefore did not possess specific details to prescribe, guide or evaluate the design of Industry 4.0 systems. Hence, the design methodology presented in this research provided greater prescription to guide design and implementation efforts, which served to;

- **Connect conceptual, software and implementation models.** The design decisions for Industry 4.0 and related systems were modelled internally at different levels of abstraction, with corporate/management producing high-level conceptual models, and engineering/computing focused on lower-level design and implementation models. In addition to differing abstractions, these modelling approaches also possessed unique vocabularies and symbols, which served to obfuscate requirements and ideas. Although enforcing a single modelling language or process was considered internally, the natural resistance from staff and potential negative impacts on operations were deemed too great to enact change. Thus, the unified design methodology provided an alternative approach to connect, extract, and integrate design details from conceptual, software and implementation models, without forcing disparate internal stakeholders to move from their existing modelling activities.

- **Prescribe fundamental acceptance criteria for Industry 4.0 initiatives.** The internal teams responsible for developing information systems, emerging technologies, and data analytics to support Industry 4.0 operations, were largely focused on the discovery and exploration of technologies, with less attention given to the general requirements and concerns for Industry 4.0, which should naturally inform and govern technology selection. Therefore, to increase awareness and encourage teams to consider technologies and systems in the context of Industry 4.0, the unified design methodology prescribed and highlighted acceptance criteria observed through interactions with stakeholders in the facility, and those evident in the literature.

#### ARCHITECTURE

The architecture dimension references the technical components comprising the proposed industrial analytics infrastructure, with current approaches commonly utilising hierarchal technology layers (e.g. ISA-95), while the proposed approach considers how these technical components can form a natural closed-loop lifecycle. Although many arguments can be made relating to architectural decisions, the lifecycle architecture was proposed to provide designers and stakeholders with some context regarding the positioning of systems and technologies, while also demonstrating the importance of making analytics and intelligence accessible to factory operations. A more definitive and prescribed lifecycle architecture was deemed necessary after observing numerous discussions between internal teams failing to produce a consensus regarding the position of information and control systems in the ISA-95 reference model. Thus, the unified design methodology from this research proposed the notion of an industrial analytics lifecycle, which served to;

- **Classify data flows throughout the factory.** The industrial information systems observed within the facility (i.e. building management system and related controllers) demonstrated loose data management and governance practices, which resulted in significant amounts of time being used to discover and access data repositories. To demystify some of the intertwined and complex relationships impeding data discovery, the unified design methodology classified (a) the ownership of data sources and systems using the identified operating teams, and (b) the communication latency of each source and system. These classifications provided stakeholders with a common understanding of data ownership and information flows throughout the factory, while also giving some context to batch and real-time processes.

- **Highlight the lack of embedded intelligence.** Before the introduction of the unified design methodology, the perceived view of architecture followed one depicting hierarchal technology layers, where data moved from the factory floor at the bottom layer, to enterprise and reporting systems residing in the top layers. Although there is nothing inherently wrong with such hierarchical perspectives, in this instance the intelligence and insights derived at the top layers were disconnected from factory operations. Hence, the unified design methodology proposed the idea that architecture could be designed around the lifecycle of data flowing through the factory, and data-driven intelligence incorporated in real-time operations (e.g. energy).

**GUIDELINES**

The guideline dimension refers to the development of high-level design requirements and ideologies for the industrial analytics infrastructure, with current approaches influenced by internal organisation-level policies, guides and personnel, while the proposed approach encourages the use of Industry 4.0 design principles. In most instances, the observed Industry 4.0 initiatives utilised existing policies and procedures governing the application of technology (e.g. use of public cloud providers was not permitted). Of course, these guideline pre-dated Industry 4.0 and smart manufacturing endeavours, which naturally meant specific requirements or characteristics associated with Industry 4.0 could be easily overlooked. An investigation of these emerging requirements within the literature, highlighted some conflicts with internal policies and guidelines, including the inability of systems to (a) automatically scale up or down based on-demand, (b) embed advanced analytics in factory operations, and (c) deliver systems that are open and non-proprietary, to name a few. Thus, the design methodology addressed these conflicts by describing and encouraging guidelines aligned with core Industry 4.0 design principles and concerns, which served to;

- **Created an awareness of Industry 4.0 design principles.** Before the unified design methodology was proposed, internal teams were not aware of the emerging design principles pertaining to Industry 4.0. These design principles focused and provided context to team-level discussions around industrial analytics, systems and processes, which were being considered as part of the partners overall technical roadmap.

- **Highlight potential shortcomings of existing guidelines.** The emerging design principles and requirements for Industry 4.0 enabled internal teams to reflect on previous and on-going projects, and highlight where certain shortcoming in existing guidelines potentially conflicted with Industry 4.0 alignment. One notable guideline that was reversed during these discussions was the use of the AWS public cloud platform, which could provide significantly more on-demand compute capacity than the corporate private cloud.

#### EVALUATION

The evaluation dimension relates to the procedures used to assess industrial analytics systems and capabilities, with current approaches focused on commercial technology acquisition and feature availability, while the proposed approach encourages the use of performance metrics and assessments. There were no formal evaluation methods governing the acceptance of Industry 4.0 technologies and systems, and therefore internal evaluation methods were somewhat subjective and ad hoc. These evaluations focused on comparing planned and implemented features, with successful evaluations implying the implemented systems were delivered as planned. Although such evaluations denote successful implementation, they overlook the general real/near-time performance requirements for Industry 4.0. Thus, the unified design methodology introduced the notion that industrial cyber-physical systems (and similar) supporting Industry 4.0 operations should be subject to real/near-time performance analysis and evaluation, which may be undertaken pre-implementation using simulation, or post-implementation using real-world stress testing. The definition of performance and acceptance criteria to evaluate systems supporting Industry 4.0 served to;

- **Introduce multidimensional performance for Industry 4.0.** This evaluation and acceptance criteria ensured fundamental concerns were considered during the design process, which incorporated Industry 4.0, corporate stakeholder, and functional concerns. These concerns provided some context upon which to evaluate successful designs and implementations, while also encouraging participants to develop a greater appreciation for concerns outside their core discipline and/or perspectives.

- **Quantify assessment of industrial analytics capabilities.** The introduction of the industrial analytics maturity assessment arose from the inability of internal teams to objectively measure current capabilities, or describe the impact of particular design/implementation changes. Hence, the maturity model presented in this provided a tool to measure and compare capabilities, such as openness, factory-to-cloud integration, and real-time analytics. Although there are many directions and applications for the tool, the maturity model has been used by internal teams to investigate the value of projects that are not bound to hard engineering applications (e.g. frameworks, architectures etc.).

### 6.3.2 Implementation of industrial analytics architecture

A fog computing architecture was used to implement the industrial cyber-physical system and enable real-time embedded machine learning model execution, while a more common cloud computing architecture was used to develop an industrial data pipeline to support batch data integration, processing and exploration. The performance analysis of fog and cloud cyber-physical interfaces illustrated some of the strengths and weaknesses of both approaches. In particular, industrial engineering applications dependent on raw compute performance (e.g. execution of complex machine learning models) may benefit from interfacing with the cloud, while those applications demanding consistent and reliable real-time execution (e.g. minimise failed communications) may choose to interface using the fog paradigm. Of course, many engineering applications shall require a mixture of both compute latency and consistency to satisfy requirements, which may be addressed by altering the hardware and software architecture of the underlying cyber-physical platform.

### INTELLIGENCE

The intelligence dimension refers to where the primary computation and decision-making is undertaken in the factory, with current approaches favouring central intelligence and processing (e.g. service residing on cloud server), while the proposed approach encourages decentralised intelligence and processing (e.g. services embedded on the edge). In the context of this research, the data processing, reporting and manufacturing intelligence produced by the building management systems was performed on a centralised server, with informed decision-making regarding energy performance and optimisation dependent on access to the server. In addition, other industrial information systems encountered during the general investigative stages of this research (e.g. manufacturing execution system) exhibited the same intelligence characteristics.

Although this research does not claim centralised intelligence should be considered a negative trait, such computation does not naturally align with Industry 4.0 design principles and requirements pertaining to decentralised and autonomous decision-making, where consistent, secure and reliable connections to central servers and resources may not be feasible due to the increasing number of disparate devices, sensors and technologies distributed across networks-of-networks. Thus, the industrial cyber-physical system implemented during this research investigated the potential use of fog/edge processing to embed predictive machine learning models in real-time factory operations.

### WORKFLOWS

The workflow dimension relates to the data processing activities needed to support the development and deployment of industrial analytics models, with current approaches depending heavily on manual activities, while the proposed approach encourages automated and scalable data processing workflows. Although somewhat trivial when observing from a high-level, the time and cost associated with industrial data acquisition and cleaning represented significant portions of project effort. These substantial efforts were required to (a) identify owners of each data source, (b) verify data was actively logging on a day-to-day basis, (c) implement the necessary tools and/or protocols to acquire the necessary data, and (d) transform the acquired data to a consistent and analytics-ready state.

Based on observations during this research, these individual workflow tasks required much discussion, investigation and implementation to resolve. While these tasks may be unavoidable due to different technology permutations, or the need for human engagement on-site (e.g. data owners), the manual approach employed by engineering and analytics teams meant efforts were typically duplicated across projects. Thus, the industrial data pipeline supporting factory-to-cloud integration automated the ingestion, processing and serving of analytics-ready data.

#### TECHNOLOGY

Technology describes behaviours towards technology adoption, with current approaches utilising commercial and proprietary solutions, while the proposed approach adopts the ideology of open and standards-driven technology. No policies regarding technology openness were evident during the research, with those technologies and systems encountered typically appearing closed or proprietary, which naturally impedes opportunities for integration and interoperability with third-party systems and data consumers. However, classifying systems using different levels of openness can be obfuscated where on-site administrators possess insufficient system knowledge, which may result in open and/or standard interfaces not being presented.

In the case of the building management system used in this research, the commercial software vendor did not expose an open or standard interface for third-party applications, which meant energy insights (e.g. excessive use of compressed air) could not be seamlessly integrated with other systems and processes relevant to optimising and maintaining energy operations in the factory. Such closed systems conflict with Industry 4.0 principles encouraging the use of openness and standards, and therefore can impede facilities from developing a technology ecosystem that supports smart manufacturing operations. Therefore, the industrial cyber-physical system implemented during this research sought to demonstrate how openness and/or standards can be considered at different layers, from industrial protocols (i.e. OPC-UA) for data acquisition, to encoding embedded machine learning models (i.e. PMML) for real-time predictive analytics.

#### INTEGRATION

Integration refers to the methods underpinning system interconnectivity and interoperability, with current approaches depending on custom and ad hoc integration routines, while the proposed approach considers the application of standard programmatic interfaces. The data integration approaches used across analytics and Industry 4.0 projects were largely inconsistent, and employed different tools, protocols and processes. Given the lack of uniformity relating to data integration, these methods were classified as ad hoc and custom integrations. In addition to the risk of duplicating data integration routines, the sporadic adoption of different integration tools introduced additional overhead in terms of (a) upskilling and training of staff, and (b) creating dependencies on particular integration tools (i.e. changing data integration tools may stop another application from working correctly). Thus, the industrial data pipeline and cyber-physical system implementations presented in this research employed web service interfaces to standardise data requests for energy data.

**ANALYTICS**

The analytics dimension relates to the delivery and usage scenarios for industrial analytics models, with current approaches applying analytics on batch (i.e. historic) operational data from standalone computers, while the proposed approach promotes the use of embedded industrial analytics in the factory to positively affect real-time decision-making and operations. The internal analytics teams engaged during this research used statistical software applications (e.g. R or SAS) for data exploration and predictive modelling, and business intelligence applications (e.g. Tableau or Qlikview) for data summary and visualisation. One of the main issues identified with these standard analytics operations related to the inadequate throughput of operational insights to the factory, which must be facilitated to enable (a) timely human-assisted decision-making, and (b) automated machine-to-machine decision-making. Although developing data-driven insights using desktop applications forms an important part of the analytics process, for Industry 4.0 scenarios these insights must be accessible to real-time factory operators and systems to deliver results (e.g. energy savings). Thus, the overall industrial analytics architecture and energy fault detection scenario demonstrated how traditional analytics processes for model building (i.e. data pipeline), could be integrated with embedded real-time analytics (i.e. cyber-physical system) to deliver insights to factory-level operations.

## 6.4 Scope and limitations

Given the pervasiveness of both smart manufacturing and industrial cyber-physical systems, the contributions and implementations presented in this research were aligned with the resources and timeframe available. The following points highlight some of the more prominent limitations of this research;

- **Review methods and process:** given the inconsistent research themes and multidisciplinary influences relating to industrial analytics and cyber-physical systems for Industry 4.0, the broad review methods and processes were adopted with the intention of creating a well-rounded understanding of the factors that may influence design and implementation. Although systematic mapping and review processes are appropriate for underdeveloped fields, there is an inherit trade-off in the depth that can be delivered on one particular topic.

- **Predictive engineering applications:** although cyber-physical systems may incorporate different forms of analytics and simulation models, this research focused exclusively on predictive engineering applications that employ machine learning methods, and how these applications could be integrated with industrial cyber-physical systems and processes.

- **Machine learning implementations:** the perspective that industrial predictive analytics resides within the machine learning domain was adopted by this research. However, the focus on machine learning was primarily driven by industry interest relating to the technology, and its adoption does not aim to discount the usefulness or applicability of traditional statistical methods, or future artificial intelligence approaches. In addition, such scope refinement was also important to facilitate the development and deployment of a real-world industrial cyber-physical system.

- **Industrial energy operations:** the deployment and testing of the industrial cyber-physical system utilised energy operations in the factory, and more specifically the identification of operational issues within an AHU. This area of operations was chosen for two reasons. Firstly, energy information systems and databases were of good quality and maintained regularly. Secondly, internal governance policies did not significantly restrict access to these energy information systems and databases.

- **Communication interfaces:** given the reasonable quality of energy information systems and databases, the implemented industrial cyber-physical system only needed to support a subset of potential device and file-based data sources (i.e. SQL and OPC-UA). However, the need to manage large and obscure libraries of communication interfaces shall continue to decrease as more high-level protocols and interfaces are adopted, with OPC-UA receiving significant industry support for ISA-95 and RAMI 4.0 specifications.

## 6.5 Future work

The knowledge and insights generated from this research have formed the basis of an Enterprise Ireland commercialisation project, which will extend the fog computing industrial cyber-physical system concept as a commercial platform to share and deploy production-ready machine learning models across industry. The primary advances from this future research include;

- **Interchangeable cloud and fog interfaces:** implementation of expert rules to determine when the industrial cyber-physical system should use (a) fog interfaces, or (b) cloud interfaces to execute machine learning models (e.g. use the cloud for more intensive computation when latency constraints are loose).
- **Industrial analytics marketplace:** development of an industrial analytics marketplace that enables 'one-click' deployment of PMML-encoded machine learning models, whereby researchers and third-party developers can license production-ready models to large-scale manufacturing facilities.

# References


[1] K. Bruton, P. Raftery, P. O'Donovan, N. Aughney, M. M. Keane, and D. T. J. O'Sullivan, "Development and alpha testing of a cloud based automated fault detection and diagnosis tool for Air Handling Units," *Autom. Constr.*, vol. 39, no. 0, pp. 70–83, Apr. 2014.

[2] ISO, "ISO 14644-1:2015 Cleanrooms and associated controlled environments -- Part 1: Classification of air cleanliness by particle concentration," 2015. [Online]. Available: https://www.iso.org/standard/53394.html. [Accessed: 21-Dec-2017].

[3] M. A. Piette, S. K. Kinney, and P. Haves, "Analysis of an information monitoring and diagnostic system to improve building operations," *Energy Build.*, vol. 33, no. 8, pp. 783–791, Oct. 2001.

[4] International Energy Agency, "Computer Aided Evaluation of HVAC System Performance, 342002."

[5] Michael D. Myers, "Qualitative Research in Information Systems," *Assoc. Inf. Syst.*, no. March, p. Not applicable, 1997.

[6] M. R. De Villiers, "Three approaches as pillars for interpretive Information Systems research: development research, action research and grounded theory," *South African Inst. Comput. Sci. Inf. Technol.*, pp. 111–120, 2005.

[7] C. Potter, "An Analysis of Action Research Projects," UNT / Denton ISD PDS, 2014.

[8] M. A. Abdel-Fattah, "Grounded theory and action research as pillars for interpretive information systems research: A comparative study," *Egypt. Informatics J.*, vol. 16, no. 3, pp. 309–327, 2015.

[9] R. L. Baskerville, "Investigating Information Systems with Action Research," *Commun. Assoc. Inf. Syst.*, vol. 2, no. 3, pp. 1–32, 1999.

[10] E. DePoy and L. Gitlin, *Introduction to Research, 5th Edition*, 5th Editio. 2015.

[11] D. Schon, *Educating the reflective practitioner: toward a new design for teaching and learning in the professions*, 1st Edition. 1987.

[12] P. O'Donovan, K. Leahy, K. Bruton, and D. T. J. O'Sullivan, "Big data in manufacturing: a systematic mapping study," *J. Big Data*, vol. 2, no. 1, p. 20, 2015.

[13] P. O'Donovan, K. Leahy, K. Bruton, and D. T. J. O'Sullivan, "An industrial big data pipeline for data-driven analytics maintenance applications in large-scale smart manufacturing facilities," *J. Big Data*, vol. 2, no. 1, p. 25, 2015.

[14] P. O'Donovan, K. Bruton, and D. T. J. O'Sullivan, "Case study: the implementation of a data-driven industrial analytics methodology and platform for smart manufacturing," *Int. J. Progn. Heal. Manag.*, 2016.

[15] P. O'Donovan, K. Bruton, and D. T. J. O'Sullivan, "IAMM: A maturity model for measuring industrial analytics capabilities in large-scale manufacturing facilities," *Int. J. Progn. Heal. Manag.*, vol. 7, 2016.

[16] P. O'Donovan, C. Gallagher, K. Bruton, and D. T. J. O'Sullivan, "A fog computing industrial cyber-physical system for embedded low-latency machine learning Industry 4.0 applications," *Manuf. Lett.*, 2018.

[17] P. O. Donovan *et al.*, "A data pipeline for PHM data-driven analytics in large-scale smart manufacturing facilities," *Annu. Conf. Progn. Heal. Manag. Soc.*, no. September, pp. 367–376, 2015.

[18] P. O'Donovan, C. Gallagher, K. Leahy, S. Blake, K. Bruton, and D. T. J. O'Sullivan, "A systematic mapping of industrial cyber-physical systems research for Industry 4.0," in *International Manufacturing Conference*, 2017.

[19] K. Bruton, B. P. Walsh, D. Ó. Cusack, P. O'Donovan, and D. T. J. O'Sullivan, "Enabling Effective Operational Decision Making on a Combined Heat and Power System Using the 5C Architecture," *Procedia CIRP*, vol. 55, pp. 296–301, 2016.

[20] K. Bruton, P. O. Donovan, A. Mcgregor, and D. T. J. O. Sullivan, "Design and development of a software tool to assist ISO 50001 implementation in the manufacturing sector," 2016.

[21] K. Leahy, C. Gallagher, K. Bruton, P. O'Donovan, and D. T. J. O'Dullivan, "Automatically Identifying and Predicting Unplanned Wind Turbine Stoppages Using SCADA and Alarms System Data: Case Study and Results," *J. Phys. Conf. Ser.*, vol. 926, no. 1, 2017.

[22] C. V Gallagher, K. Leahy, P. O'Donovan, and K. Bruton, "Development and application of a machine learning supported methodology for measurement and verification (M&V) 2.0," *Energy Build.*, pp. 1–17, 2018.

[23] B. Dworschak and H. Zaiser, "Competences for cyber-physical systems in manufacturing – first findings and scenarios," *Procedia CIRP*, vol. 25, pp. 345–350, 2014.

[24] B. Bagheri, S. Yang, H. Kao, and J. Lee, "Cyber-physical Systems Architecture for Self-Aware Machines in Industry 4 . 0 Environment," *IFAC Pap.*, pp. 1622–1627, 2015.

[25] M. Riedl, H. Zipper, M. Meier, and C. Diedrich, "Cyber-physical systems alter automation architectures," *Annu. Rev. Control*, vol. 38, no. 1, pp. 123–133, 2014.

[26] R. F. Babiceanu and R. Seker, "Big Data and virtualization for manufacturing cyber-physical systems: A survey of the current status and future outlook," *Comput. Ind.*, 2016.

[27] J. Davis, T. Edgar, J. Porter, J. Bernaden, and M. Sarli, "Smart manufacturing, manufacturing intelligence and demand-dynamic performance," *Comput. Chem. Eng.*, vol. 47, pp. 145–156, Dec. 2012.

[28] J. Lee, E. Lapira, B. Bagheri, and H. Kao, "Recent advances and trends in predictive manufacturing systems in big data environment," *Manuf. Lett.*, vol. 1, pp. 38–41, Oct. 2013.

[29] J. Lee, "Recent Advances and Transformation Direction of PHM," *NIST*, 2014. [Online]. Available: http://www.nist.gov/el/isd/upload/Keynote_Lee_IMS-University_of_Cincinnati_updated.pdf. [Accessed: 03-Aug-2015].

[30] P. Wright, "Cyber-physical product manufacturing," *Manuf. Lett.*, vol. 2, no. 2, pp. 49–53, 2014.

[31] S. Manufacturing *et al.*, "Implementing 21st Century Smart Manufacturing," 2011.

[32] Smart Process Manufacturing Engineering Virtual Organization Steering Committee, "Smart Process Manufacturing an Operations and Technology Roadmap," p. 61, 2009.

[33] F. Meziane, S. Vadera, K. Kobbacy, and N. Proudlove, "Intelligent systems in manufacturing: current developments and future prospects," *Integr. Manuf. Syst.*, vol. 11, no. 4, pp. 218–238, 2000.

[34] P. Sharma and M. Sharma, "Artificial Intelligence in Advance Manufacturing Technology-A Review Paper on Current Application," *Int. J. Eng. Manag. Sci.*, vol. 1, no. 1, pp. 4–7, 2014.

[35] J. D. Ucla, M. Mills, and J. Ottino, "Smart Manufacturing , Manufacturing Intelligence and Demand-Dynamic Performance Smart Manufacturing Leadership Coalition ( SMLC ) Jan 30 Wall Street Journal : The Coming Tech Led Boom," 2011.

[36] D. Zuehlke, "SmartFactory—Towards a factory-of-things," *Annu. Rev. Control*, vol. 34, no. 1, pp. 129–138, Apr. 2010.

[37] J. Lee, H.-A. A. Kao, and S. Yang, "Service Innovation and Smart Analytics for Industry 4.0 and Big Data Environment," *Procedia CIRP*, vol. 16, pp. 3–8, 2014.

[38] S. Chand and J. Davis, "What is Smart Manufacturing?," *Time Magazine*, 2010.

[39] S. Heng, "Industry 4.0: Huge potential for value creation waiting to be tapped. Deutsche Bank Research," 2014. [Online]. Available: http://www.dbresearch.com/servlet/reweb2.ReWEB?rwsite=DBR_INTERNET_EN-PROD&rwobj=ReDisplay.Start.class&document=PROD0000000000335628.

[40] Y. Lu, K. C. Morris, and S. Frechette, *Current Standards Landscape for Smart Manufacturing Systems*. 2016.

[41] D. Emerson, H. Kawamura, and W. Matthews, "Plant-to-business interoperability using the ISA-95 standard," 2007.

[42] D. Schulz, "FDI and the Industrial Internet of Things: Protection of Investment for Industrie 4.0," *IEEE Int. Conf. Emerg. Technol. Fact. Autom. ETFA*, 2015.

[43] B. Scholten, "Integrating ISA-88 and ISA-95," in *ISA Expo*, 2007, pp. 2–4.

[44] M. Discrete and M. Group, "A Discrete Manufacturing Reference Architecture (DIRA) Framework Whitepaper,", 2011.

[45] B. Scholten, "Integrating ISA-88 and ISA-95", 2007.

[46] R. Baheti and H. Gill, "Cyber-physical Systems," *Impact Control Technol.*, no. 1, pp. 161--166, 2011.

[47] Horizon 2020, "Smart Cyber Physical Systems." [Online]. Available: https://ec.europa.eu/programmes/horizon2020/en/h2020-section/smart-cyber-physical-systems.

[48] NIST, "Cyber Physical Systems." [Online]. Available: http://www.nist.gov/cps/.

[49] Cyber Physical Systems Engineering Lab, "Cyber Physical Systems." [Online]. Available: http://www.cpse-labs.eu/.

[50] National Science Foundation, "Cyber Physical Systems." [Online]. Available: https://www.nsf.gov/funding/pgm_summ.jsp?pims_id=503286.

[51] J. Lee, H. D. Ardakani, S. Yang, and B. Bagheri, "Industrial Big Data Analytics and Cyber-physical Systems for Future Maintenance & Service Innovation," *Procedia CIRP*, vol. 38, pp. 3–7, 2015.

[52] M. Hermann, T. Pentek, and B. Otto, "Design Principles for Industrie 4.0 Scenarios: A Literature Review," *Tech. Univ. Dortmund*, vol. 1, no. 1, pp. 4–16, 2015.

[53] D. Swink, "Smart Manufacturing Leadership Coalition ( SMLC ) The infusion of intelligence that transforms the way Industries conceptualize , design , and operate the manufacturing enterprise ", 2014.

[54] Smart Manufacturing Leadership Coalition, "SMLC Forum: Priorities, Infrastructure, and Collaboration for Implementation of Smart Manufacturing,", pp. 1–27, 2012.

[55] P. Basanta-Val, M. García-Valls, and I. Estévez-Ayres, "Towards a cyber-physical architecture for industrial systems via real-time Java technology," *Proc. - 10th IEEE Int. Conf. Comput. Inf. Technol. CIT-2010, 7th IEEE Int. Conf. Embed. Softw. Syst. ICESS-2010,* pp. 2341–2346, 2010.

[56] G. Philipp, P. Cato, and M. Amberg, "Data processing requirments of Industry 4.0 - use cases for big data applications," in *23rd European Conference on Information Systems (ECIS)*, 2015, pp. 1–13.

[57] A. Buda, K. Främling, J. Borgman, M. Madhikermi, and S. Kubler, "Data supply chain in Industrial Internet," *11th IEEE World Conf. Fact. Commun. Syst. (WFCS 2015)*, 2015.

[58] D. Broman, E. A. Lee, S. Tripakis, and M. Törngren, "Viewpoints, formalisms, languages, and tools for cyber-physical systems," *Proc. 6th Int. Work. Multi-Paradigm Model. - MPM '12*, pp. 49–54, 2012.

[59] C. Yang, W. Shen, and X. Wang, "A formulation for IoT-enabled dynamic service selection across multiple manufacturing clouds," in *IEEE International Conference on Computer Supported Cooperative Work in Design*, 2016.

[60] C. Wang, H. Vo, and P. Ni, "An IoT Application for Fault Diagnosis and Prediction," *Proc. - 2015 IEEE Int. Conf. Data Sci. Data Intensive Syst. 8th IEEE Int. Conf. Cyber, Phys. Soc. Comput. 11th IEEE Int. Conf. Green Comput. Commun. 8th IEEE Inte*, pp. 726–731, 2016.

[61] I. Ungurean, N. C. Gaitan, and V. G. Gaitan, "An IoT architecture for things from industrial environment," *IEEE Int. Conf. Commun.*, 2014.

[62] A. Marini and D. Bianchini, "Big data as a service for monitoring cyber-physical production systems," *Proc. - 30th Eur. Conf. Model. Simulation, ECMS 2016*, pp. 579–586, 2016.

[63] F. Tao, Y. Cheng, L. Da Xu, L. Zhang, and B. H. Li, "CCIoT-CMfg: Cloud computing and internet of things-based cloud manufacturing service system," *IEEE Trans. Ind. Informatics*, vol. 10, no. 2, pp. 1435–1442, 2014.

[64] M. Xia, T. Li, Y. Zhang, and C. W. de Silva, "Closed-loop design evolution of engineering system using condition monitoring through internet of things and cloud computing," *Comput. Networks*, vol. 101, pp. 5–18, 2015.

[65] X. Yue, H. Cai, H. Yan, C. Zou, and K. Zhou, "Cloud-assisted industrial cyber-physical systems: An insight," *Microprocess. Microsyst.*, vol. 39, no. 8, pp. 1262–1270, 2015.

[66] C. Yu, X. Xu, and Y. Lu, "Computer-Integrated Manufacturing, Cyber-Physical Systems and Cloud Manufacturing - Concepts and relationships," *Manuf. Lett.*, vol. 6, pp. 5–9, 2015.

[67] A. Brusaferri, A. Ballarino, F. A. Cavadini, D. Manzocchi, and M. Mazzolini, "CPS-based hierarchical and self-similar automation architecture for the control and verification of reconfigurable manufacturing systems," *19th IEEE Int. Conf. Emerg. Technol. Fact. Autom. ETFA 2014*, 2014.

[68] N. Jazdi, "Cyber physical systems in the context of Industry 4.0," *Autom. Qual. Test.*, pp. 2–4, 2014.

[69] B. Bagheri, S. Yang, H. A. Kao, and J. Lee, "Cyber-physical systems architecture for self-aware machines in industry 4.0 environment," *IFAC Proc. Vol.*, vol. 48, no. 3, pp. 1622–1627, 2015.

[70] C. Klotzer and A. Pflaum, "Cyber-physical systems as the technical foundation for problem solutions in manufacturing, logistics and supply chain management," in *International Conference on the Internet of Things*, 2015, pp. 12–19.

[71] B. S. Iarovyi, W. M. Mohammed, A. Lobov, B. R. Ferrer, and J. L. M. Lastra, "Cyber–Physical Systems for Open-Knowledge-Driven Manufacturing Execution Systems," *Proc. IEEE*, vol. 104, no. 5, pp. 1142–1154, 2016.

[72] Z. Song, Y. Sun, J. Wan, and P. Liang, "Data quality management for service-oriented manufacturing cyber-physical systems," *Comput. Electr. Eng.*, pp. 1–11, 2016.

[73] C. Zhai, Z. Zou, Q. Chen, L. Xu, L.-R. Zheng, and H. Tenhunen, "Delay-aware and reliability-aware contention-free MF–TDMA protocol for automated RFID monitoring in industrial IoT," *J. Ind. Inf. Integr.*, vol. 3, pp. 8–19, 2016.

[74] Y. Deng, H. Zhu, G. Zhang, and H. Yin, "Design and implementation of an 'Internet of Things' based quality control system," *2013 Int. Conf. Comput. Control. Informatics Its Appl.*, pp. 141–146, 2013.

[75] G. Butala, P.,Vrabič; and R. Oosthuizen, "Distributed Manufacturing Systems and the Internet of Things: a Case Study," *SAIIE25 Proc.*, pp. 1–8, 2013.

[76] D. Kerpen, M. Lohrer, M. Saggiomo, M. Kemper, J. Lemm, and Y. S. Gloy, "Effects of cyber-physical production systems on human factors in a weaving mill: Implementation of digital working environments based on augmented reality," *Proc. IEEE Int. Conf. Ind. Technol.*, pp. 2094–2098, 2016.

[77] P. Leit *et al.*, "Industrial Automation based on Cyber - Physical Systems Technologies : Prototype Implementations and Challenges," pp. 1–25, 2015.

[78] J. Lee, H. D. Ardakani, S. Yang, and B. Bagheri, "Industrial Big Data Analytics and Cyber-physical Systems

for Future Maintenance & Service Innovation," *Procedia CIRP*, vol. 38, pp. 3–7, 2015.

[79] P. J. Mosterman and J. Zander, "Industry 4.0 as a Cyber-Physical System study," *Softw. Syst. Model.*, vol. 15, pp. 17–29, 2016.

[80] N. Shariatzadeh, T. Lundholm, L. Lindberg, and G. Sivard, "Integration of Digital Factory with Smart Factory Based on Internet of Things," *Procedia CIRP*, vol. 50, pp. 512–517, 2016.

[81] M. Liu, J. Ma, L. Lin, M. Ge, Q. Wang, and C. Liu, "Intelligent assembly system for mechanical products and key technology based on internet of things," *J. Intell. Manuf.*, 2014.

[82] F. Tao, Y. Wang, Y. Zuo, H. Yang, and M. Zhang, "Internet of Things in product life-cycle energy management," *J. Ind. Inf. Integr.*, pp. 1–14, 2016.

[83] T. Qu, S. Lei, Z. Wang, D. Nie, X. Chen, and G. Huang, "IoT-based real-time production logistics synchronization system under smart cloud manufacturing," *Int. J. Adv. Manuf. Technol.*, vol. 84, no. 1–4, pp. 147–164, 2016.

[84] C. Yang, W. Shen, T. Lin, and X. Wang, "IoT-enabled dynamic service selection across multiple manufacturing clouds," *Manuf. Lett.*, vol. 7, pp. 22–25, 2016.

[85] M. H. Hartmann, B. Halecker, M. Hartmann, and B. Halecker, "Management of Innovation in the Industrial Internet of Things Things,", 2015.

[86] K. Kang, T. Qu, D. Nie, T. Zhang, Z. Wang, and G. Huang, "Production system multi-stage synchronization based on collaborative optimization under the Internet-of-Things environment," *ICNSC 2016 - 13th IEEE Int. Conf. Networking, Sens. Control*, 2016.

[87] F. Pramudianto, J. Simon, M. Eisenhauer, H. Khaleel, C. Pastrone, and M. Spirito, "Prototyping the Internet of Things for the future factory using a SOA-based middleware and reliable WSNs," *IEEE Int. Conf. Emerg. Technol. Fact. Autom. ETFA*, 2013.

[88] Y. Zhang, "Real-time Data Driven Monitoring and Optimization Method for loT -based Sensible Production Process," pp. 486–490, 2013.

[89] J. Lee, B. Bagheri, and H. Kao, "Recent Advances and Trends of Cyber-Physical Systems and Big Data Analytics in Industrial Informatics," *Int. Conf. Ind. Informatics*, 2014.

[90] Q. Gao, F. Li, and C. Chen, "Research of Internet of Things applied to manufacturing execution system," pp. 661–665, 2015.

[91] M. Qiang, D. Yu-feng, X. Ting, and W. Shun-Li, "Research of Visualization Monitoring Technology Based on Internet of Things in Discrete Manufacturing Process," *2015 2nd Int. Symp. Dependable Comput. Internet Things*, pp. 128–133, 2015.

[92] C. Lee, C. Yeung, and M. Cheng, "Research on IoT Based Cyber Physical System for Industrial Big Data Analytics," pp. 1855–1859, 2015.

[93] J. Liu and J. Yu, "Research on the framework of internet of things in manufacturing for aircraft large components assembly site," *Proc. - 2013 IEEE Int. Conf. Green Comput. Commun. IEEE Internet Things IEEE Cyber, Phys.*, pp. 1192–1196, 2013.

[94] M. Garetti, L. Fumagalli, and E. Negri, "Role of ontologies for cps implementation in manufacturing," *Manag. Prod. Eng. Rev.*, vol. 6, no. 4, pp. 26–32, 2015.

[95] K. Fischer and J. Gesner, "Security architecture elements for IoT enabled automation networks," *IEEE Int. Conf. Emerg. Technol. Fact. Autom. ETFA*, 2012.

[96] J. Wan *et al.*, "Software-Defined Industrial Internet of Things in the Context of Industry 4.0," *IEEE Sens. J.*, vol. 16, no. 20, 2016.

[97] K. Ahmed, J. O. Blech, M. A. Gregory, and H. Schmidt, "Software Defined Networking for Communication and Control of Cyber-Physical Systems," *2015 IEEE 21st Int. Conf. Parallel Distrib. Syst.*, pp. 803–808, 2015.

[98] X. Ji, G. He, J. Xu, and Y. Guo, "Study on the mode of intelligent chemical industry based on cyber-physical system and its implementation," *Adv. Eng. Softw.*, vol. 99, pp. 18–26, 2016.

[99] B. Yan and G. Huang, "Supply chain information transmission based on RFID and internet of things," *Comput. Commun. Control. Manag.*, vol. 4, pp. 166–169, 2009.

[100] I. Dumitrache and S. I. Caramihai, "*The enterprise of future as a cyber-physical system*, ", vol. 46, no. 9. IFAC, 2013.

[101] L. Yuan, Y. Guo, J. Jiang, and L. Nian, "The Research on Monitoring of Discrete Manufacturing Process Based on Internet of Things," *IEEE Internet Things IEEE Cyber, Phys.*, pp. 1186–1191, 2013.

[102] D. Segura Velandia, N. Kaur, W. Whittow, P. Conway, and A. West, "Towards industrial internet of things: Crankshaft monitoring, traceability and tracking using RFID," *Robot. Comput. Integr. Manuf.*, vol. 41, pp. 66–77, 2016.

[103] T. Qu, S. Lei, Y. Chen, Z. Wang, H. Luo, and G. Q. Huang, "Internet-of-Things enabled smart production

logistics execution system based on cloud manufacturing," pp. 1–7, 2014.

[104] F. Pérez, E. Irisarri, D. Orive, M. Marcos, and E. Estevez, "A CPPS Architecture approach for Industry 4.0," in *IEEE International Conference on Emerging Technologies and Factory Automation*, 2015.

[105] J. Lee, B. Bagheri, and H.-A. Kao, "A cyber-physical systems architecture for Industry 4.0-based manufacturing systems," *Manuf. Lett.*, vol. 3, pp. 18–23, 2015.

[106] Z. Bi, G. Wang, and L. Da Xu, "A visualization platform for internet of things in manufacturing applications," *Internet Res.*, 2016.

[107] L. Wang, R. Gao, and I. Ragai, "An Integrated Cyber-Physical System for Cloud Manufacturing," *Mater. Micro Nano Technol. Prop. Appl. Syst. Sustain. Manuf.*, vol. 1, 2014.

[108] R.-Y. Chen, "An intelligent value stream-based approach to collaboration of food traceability cyber physical system by fog computing," *Food Control*, vol. 71, pp. 124–136, 2017.

[109] Y. Lu and J. Cecil, "An Internet of Things (IoT)-based collaborative framework for advanced manufacturing," *Int. J. Adv. Manuf. Technol.*, vol. 84, no. 5–8, pp. 1141–1152, 2016.

[110] P. Brizzi *et al.*, "Bringing the Internet of Things along the manufacturing line: A case study in controlling industrial robot and monitoring energy consumption remotely," *IEEE Int. Conf. Emerg. Technol. Fact. Autom. ETFA*, 2013.

[111] I. Gräßler, A. Pöhler, and J. Pottebaum, "Creation of a Learning Factory for Cyber Physical Production Systems," *Procedia CIRP*, vol. 54, pp. 107–112, 2016.

[112] K.-F. Seitz and P. Nyhuis, "Cyber-Physical Production Systems Combined with Logistic Models – A Learning Factory Concept for an Improved Production Planning and Control," *Procedia CIRP*, vol. 32, pp. 92–97, 2015.

[113] J. Michniewicz and G. Reinhart, "Cyber-physical Robotics – Automated Analysis, Programming and Configuration of Robot Cells based on Cyber-physical-systems," *Procedia Technol.*, vol. 15, pp. 566–575, 2014.

[114] J. Schuhmacher and V. Hummel, "Decentralized Control of Logistic Processes in Cyber-physical Production Systems at the Example of Logistics Learning Factory," *Procedia CIRP*, vol. 54, pp. 19–24, 2016.

[115] N. Gronau, "Determinants of an Appropriate Degree of Autonomy in a Cyber-physical Production System," *Procedia CIRP*, vol. 52, pp. 1–5, 2016.

[116] F. Shrouf and G. Miragliotta, "Energy management based on Internet of Things: practices and framework for adoption in production management," *J. Clean. Prod.*, vol. 100, pp. 235–246, 2015.

[117] B. C. Pirvu, C. B. Zamfirescu, and D. Gorecky, "Engineering insights from an anthropocentric cyber-physical system: A case study for an assembly station," *Mechatronics*, vol. 34, pp. 147–159, 2016.

[118] S. Thiede, M. Juraschek, and C. Herrmann, "Implementing Cyber-physical Production Systems in Learning Factories," *Procedia CIRP*, vol. 54, pp. 7–12, 2016.

[119] D. Mourtzis, E. Vlachou, and N. Milas, "Industrial Big Data as a Result of IoT Adoption in Manufacturing," *Procedia CIRP*, vol. 55, pp. 290–295, 2016.

[120] T. Bangemann, M. Riedl, M. Thron, and C. Diedrich, "Integration of Classical Components into Industrial Cyber-Physical Systems," *Proc. IEEE*, vol. 104, no. 5, pp. 947–959, 2016.

[121] Y. Zhang, W. Wang, N. Wu, and C. Qian, "IoT-Enabled Real-Time Production Performance Analysis and Exception Diagnosis Model," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 3, pp. 1318–1332, 2016.

[122] G. Schuh, M. Pitsch, S. Rudolf, W. Karmann, and M. Sommer, "Modular sensor platform for service-oriented cyber-physical systems in the european tool making industry," *Procedia CIRP*, vol. 17, pp. 374–379, 2014.

[123] G. Schuh, T. Potente, C. Thomas, and T. Hempel, "Short-term Cyber-physical Production Management," *Procedia CIRP*, vol. 25, pp. 154–160, 2014.

[124] P. Leitao, J. Barbosa, M. Papadopoulou, and I. Venieris, "Standardization in cyber-physical systems: The ARUM case," *2015 IEEE Int. Conf. Ind. Technol. ICIT 2015*, pp. 2988–2993, 2015.

[125] M. Herterich, F. Uebernickel, and W. Brenner, "The Impact of Cyber-physical Systems on Industrial Services in Manufacturing," *Procedia CIRP*, vol. 30, pp. 323–328, 2015.

[126] A. Rana and G. Lehal, "Smart Computing Prototype for Industry 4.0 Revolution with IOT and Bigdata Implementation Model," *Indian J. Sci. Technol.*, vol. 8, pp. 1–7, 2015.

[127] S. Shafiq, C. Sanin, E. Szczerbicki, and C. Toro, "Virtual engineering object/virtual engineering process: A specialized form of cyber physical system for industrie 4.0," *Procedia Comput. Sci.*, vol. 60, pp. 1146–1155, 2015.

[128] J. Ruan, W. Yu, Y. Yang, and J. Hu, "Design and Realize of Tire Production Process Monitoring System Based on Cyber-Physical Systems," *Int. Conf. Comput. Sci. Mech. Autom. CSMA 2015*, pp. 175–179, 2016.

[129] A. Astarloa, N. Moreira, U. Bidarte, M. Urbina, and D. Modrono, "FPGA based nodes for sub-microsecond synchronization of cyber-physical production systems on high availability ring networks," *Int. Conf. ReConFigurable Comput.*, pp. 1–6, 2016.

[130] C. Alexakos and A. Kalogeras, "Internet of Things integration to a multi agent system based manufacturing environment," *2015 IEEE 20th Conf. ETFA*, pp. 1–8, 2015.

[131] C. Wang, S. Member, Z. Bi, S. Member, L. Da Xu, and S. Member, "IoT and Cloud Computing in Automation of Assembly Modeling Systems," vol. 10, no. 2, pp. 1426–1434, 2014.

[132] S. Paulo and S. Feldmann, "Model-Driven Engineering and Semantic Technologies for the Design of Cyber-Physical Systems," vol. 2, no. 3, pp. 210–215, 2013.

[133] K. Thramboulidis and F. Christoulakis, "UML4IoT: A UML-based approach to exploit IoT in cyber-physical manufacturing systems," *Comput. Ind.*, vol. 82, pp. 259–272, 2016.

[134] M. Blackburn and P. Denno, "Virtual design and verification of cyber-physical systems: Industrial process plant design," *Procedia Comput. Sci.*, vol. 28, pp. 883–890, 2014.

[135] J. Otto, S. Henning, and O. Niggemann, "Why Cyber-physical Production Systems Need a Descriptive Engineering Approach – A Case Study in Plug & Produce," *Procedia Technol.*, vol. 15, pp. 295–302, 2014.

[136] A. Bonci, M. Pirani, and S. Longhi, "A database-centric approach for the modeling, simulation and control of cyber-physical systems in the factory of the future.," *IFAC-PapersOnLine*, vol. 49, no. 12, pp. 249–254, 2016.

[137] M. Wang, R. Y. Zhong, Q. Dai, and G. Q. Huang, "A MPN-based scheduling model for IoT-enabled hybrid flow shop manufacturing," *Adv. Eng. Informatics*, vol. 30, no. 4, pp. 728–736, 2016.

[138] L. Cheng, T. Wang, H. Xiao, Z. Wang, J. Wang, and J. Liu, "A study on the architecture of manufacturing internet of things," *Int. J. Model. Identif. Control*, vol. 23, pp. 8–23, 2015.

[139] S. Setty *et al.*, "A Unified Framework for the Design of Distributed Cyber-Physical Systems – Industrial Automation Example," pp. 996–1002, 2015.

[140] J. Chen *et al.*, "CPS Modeling of CNC Machine Tool Work Processes Using an Instruction-Domain Based Approach," *Engineering*, pp. 247–260, 2015.

[141] C. Fang, X. Liu, J. Pei, W. Fan, and P. Pardalos, "Optimal production planning in a hybrid manufacturing and recovering system based on the internet of things with closed loop supply chains," *Oper. Res.*, vol. 16, no. 3, pp. 543–577, 2016.

[142] C. Fang, X. Liu, P. M. Pardalos, and J. Pei, "Optimization for a three-stage production system in the Internet of Things: procurement, production and product recovery, and acquisition," *Int. J. Adv. Manuf. Technol.*, vol. 83, no. 5–8, pp. 689–710, 2016.

[143] G. Lanza, B. Haefner, and A. Kraemer, "Optimization of selective assembly and adaptive manufacturing by means of cyber-physical system based matching," *CIRP Ann. Manuf. Technol.*, vol. 64, no. 1, pp. 399–402, 2015.

[144] X. Sun, X. Wang, J. Wu, and Y. Liu, "Prediction-based manufacturing center self-adaptive demand side energy optimization in cyber physical systems," *Chinese J. Mech. Eng.*, vol. 27, no. 3, pp. 488–495, 2014.

[145] A. Sangiovanni-Vincentelli, W. Damm, and R. Passerone, "Taming Dr. Frankenstein: Contract-based design for cyber-physical systems," *Eur. J. Control*, vol. 18, no. 3, pp. 217–238, 2012.

[146] M. Saggiomo, M. Kemper and Y. Gloy, "Weaving machine as cyber-physical production system: Multi-objective self-optimization of the weaving process," *Int. Conf. on Industrial Technology*,2008.

[147] Z. Wen, X. Liu, Y. Xu, and J. Zou, "A RESTful framework for Internet of things based on software defined network in modern manufacturing," *Int. J. Adv. Manuf. Technol.*, vol. 84, no. 1–4, pp. 361–369, 2016.

[148] P. Hu, "A System Architecture for Software-Defined Industrial Internet of Things," *2015 IEEE Int. Conf. Ubiquitous Wirel. Broadband*, 2015.

[149] A. Schumacher, S. Erol, and W. Sihn, "A Maturity Model for Assessing Industry 4.0 Readiness and Maturity of Manufacturing Enterprises," *Procedia CIRP*, 2016, vol. 52.

[150] E. Francalanza, J. Borg, and C. Constantinescu, "A knowledge-based tool for designing cyber physical production systems," *Comput. Ind.*, 2016.

[151] J. Lee, B. Bagheri, and H.-A. Kao, "A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems," *Manuf. Lett.*, vol. 3, pp. 18–23, Jan. 2015.

[152] Z. Xiang-li, Y. Jin, Y. Kun, and L. Jian, "A remote manufacturing monitoring system based on the Internet of Things," *Int. Conf. Comput. Sci. Netw. Technol.*, pp. 221–224, 2012.

[153] L. Wang, R. Gao, and I. Ragai, "An Integrated Cyber-Physical System for Cloud Manufacturing," *Vol. 1 Mater. Micro Nano Technol. Prop. Appl. Syst. Sustain. Manuf.*, 2014.

[154] P. Leitão, A. W. Colombo, and S. Karnouskos, "Industrial automation based on cyber-physical systems

technologies: Prototype implementations and challenges," *Comput. Ind.*, vol. 81, pp. 11–25, 2016.

[155] P. Antunes, P. Carreira, and M. Da Silva, "Towards an energy management maturity model," *Energy Policy*, vol. 73, pp. 803–814, 2014.

[156] F. Oliva, "A maturity model for enterprise risk management," *Int. J. Prod. Econ.*, vol. 173, pp. 66–79, 2016.

[157] G. Lahrmann, F. Marx, R. Winter, and F. Wortmann, "Business Intelligence Maturity Models : An Overview," *44th Hawaii Int. Conf. Syst. Sci.*, pp. 1–12, 2011.

[158] M. Ofner, B. Otto, and H. Österle, "A Maturity Model for Enterprise Data Quality Management," *Enterp. Model. Inf. Systesms Archit.*, vol. 8, 2015.

[159] T. De Bruin, R. Freeze, U. Kaulkarni, and M. Rosemann, "Understanding the Main Phases of Developing a Maturity Assessment Model," *Australas. Conf. Inf. Syst.*, pp. 8–19, 2005.

# Appendix A – Activity Diagrams

## Collect data activity diagram

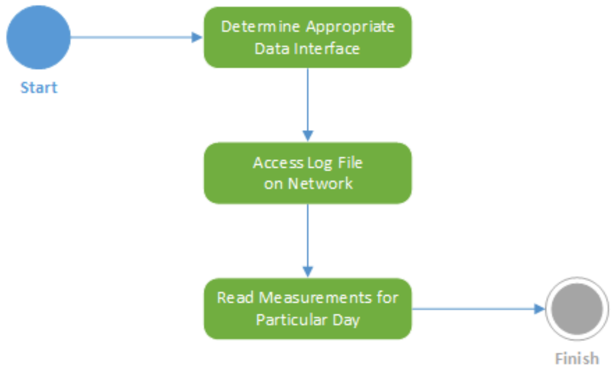



**Figure 78 Activity diagram for collect data use case**

## Transmit data activity diagram

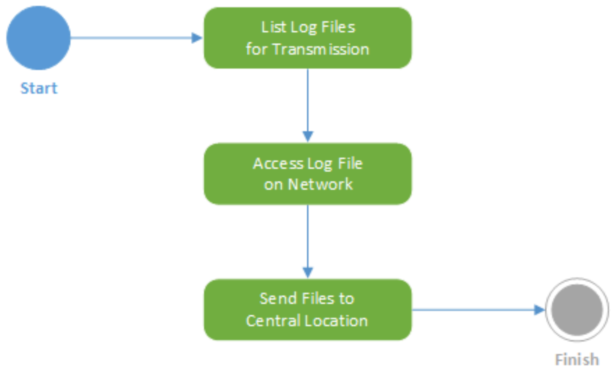



**Figure 79 Activity diagram for transmit data use case**

## Store data activity diagram

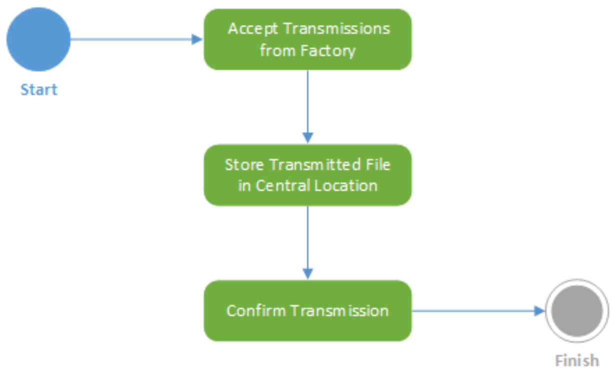



**Figure 80 Activity diagram for store data use case**

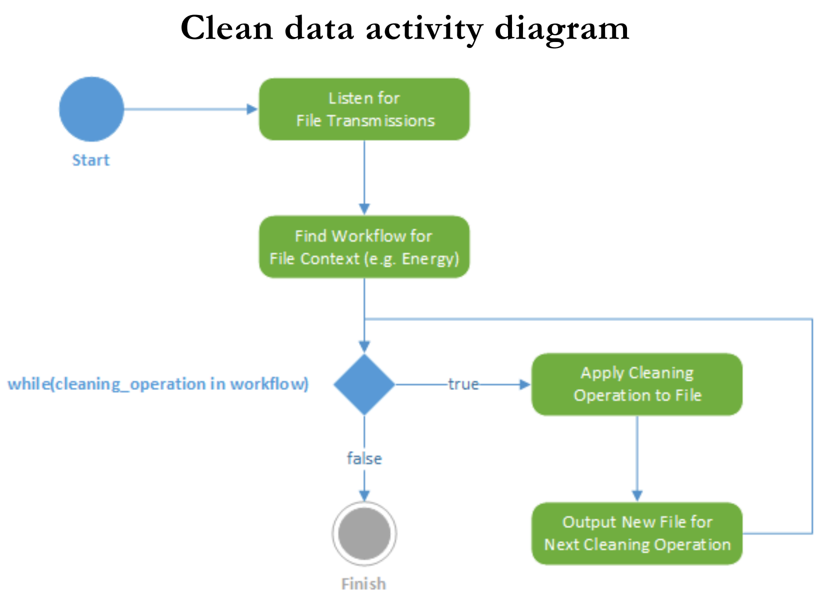



Figure 81 Activity diagram for clean data use case

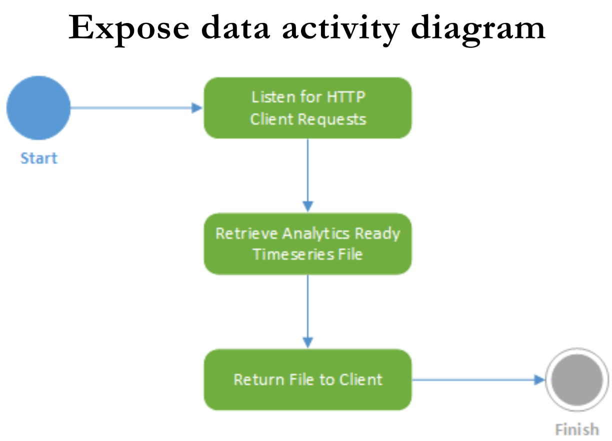



Figure 82 Activity diagram for expose data use case

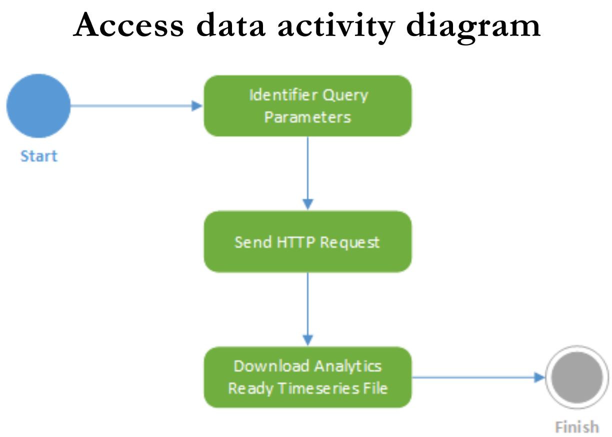



Figure 83 Activity diagram for access data use case

## Build model activity diagram

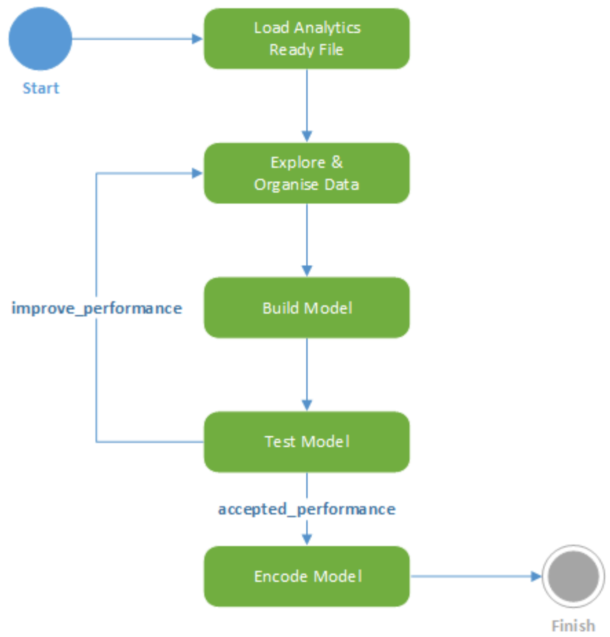



**Figure 84 Activity diagram for build model use case**

## Deploy model activity diagram

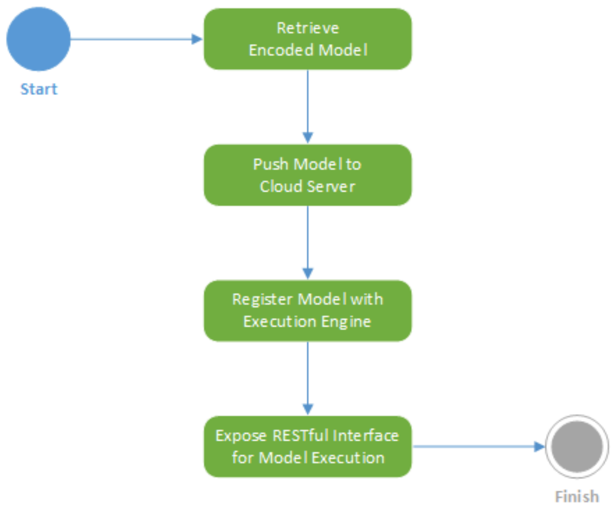



**Figure 85 Activity diagram for deploy model use case**

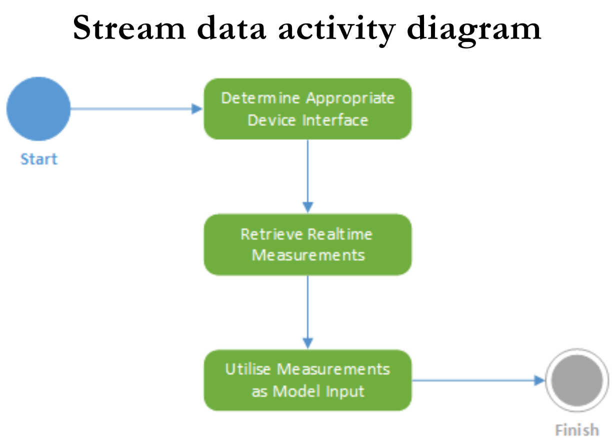



Figure 86 Activity diagram for stream data use case

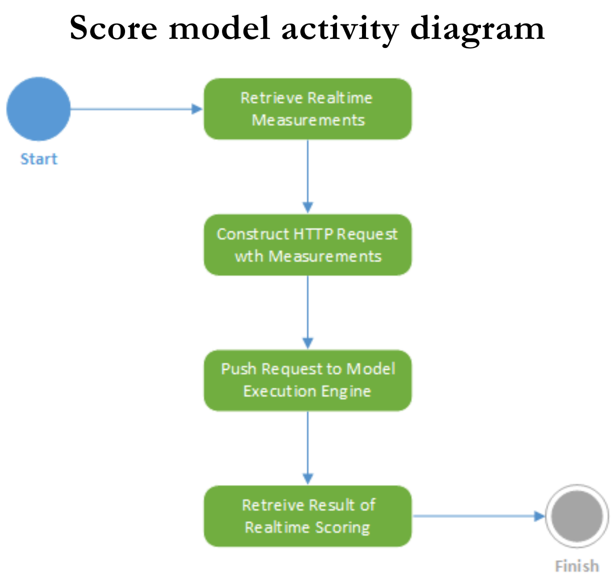



Figure 87 Activity diagram for score model use case

## Relay score activity diagram

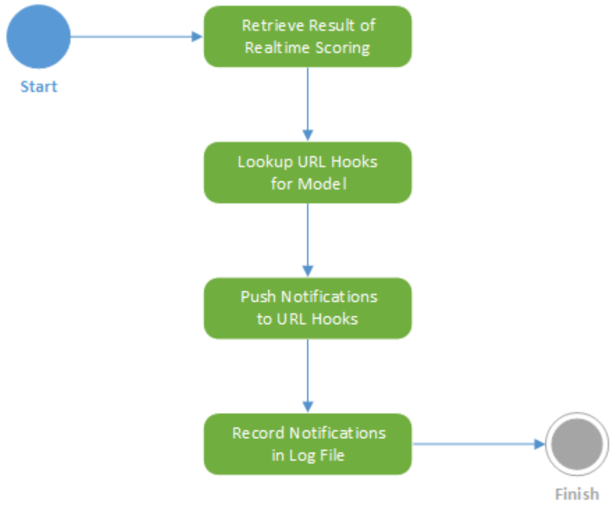



Figure 88 Activity diagram for relay score use case

# Appendix B – Sequence Diagrams

## Collect data sequence diagram

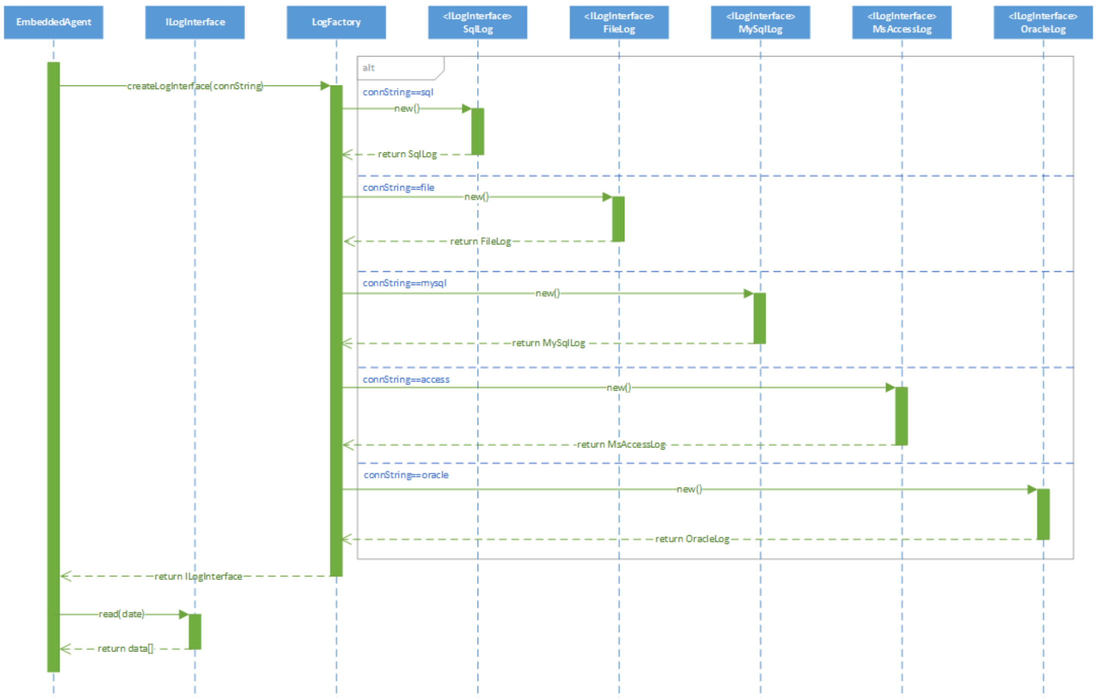



Figure 89 Sequence diagram for collect data use case

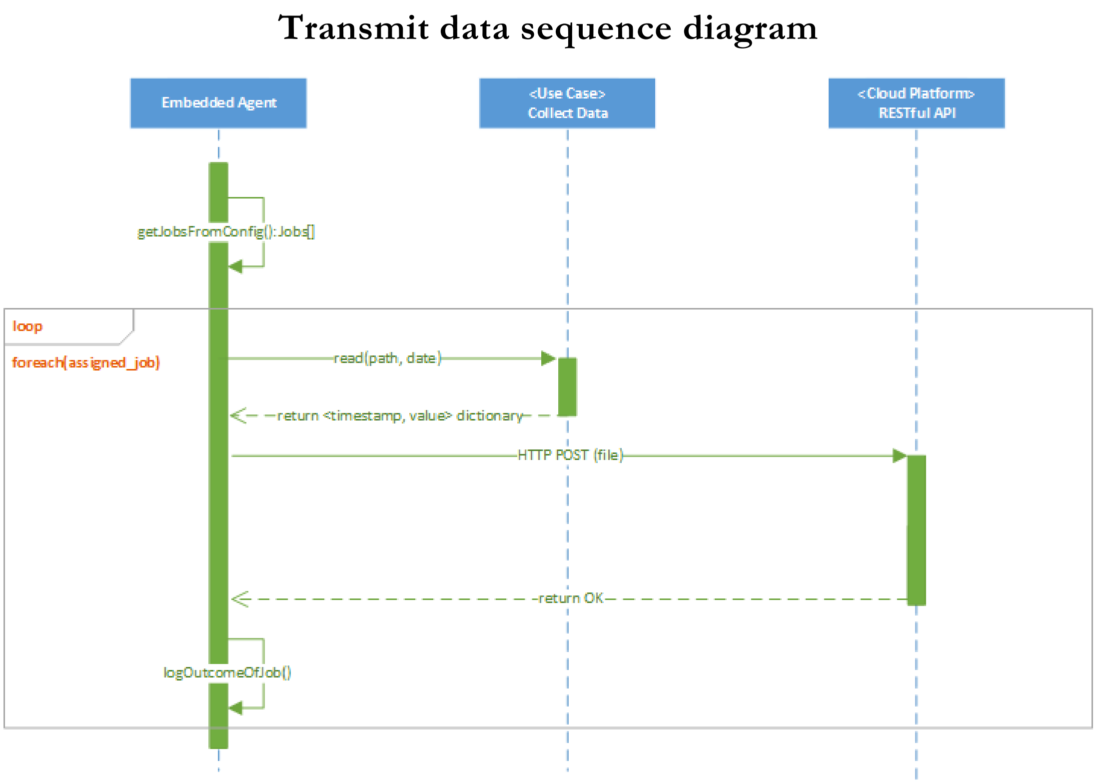



**Figure 90 Sequence diagram for transmit data use case**

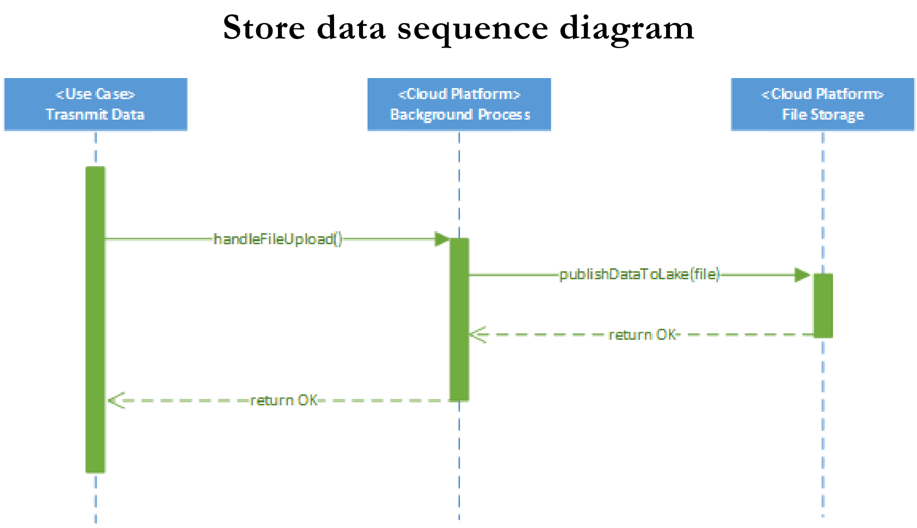



**Figure 91 Sequence diagram for store data use case**

## Clean data sequence diagram

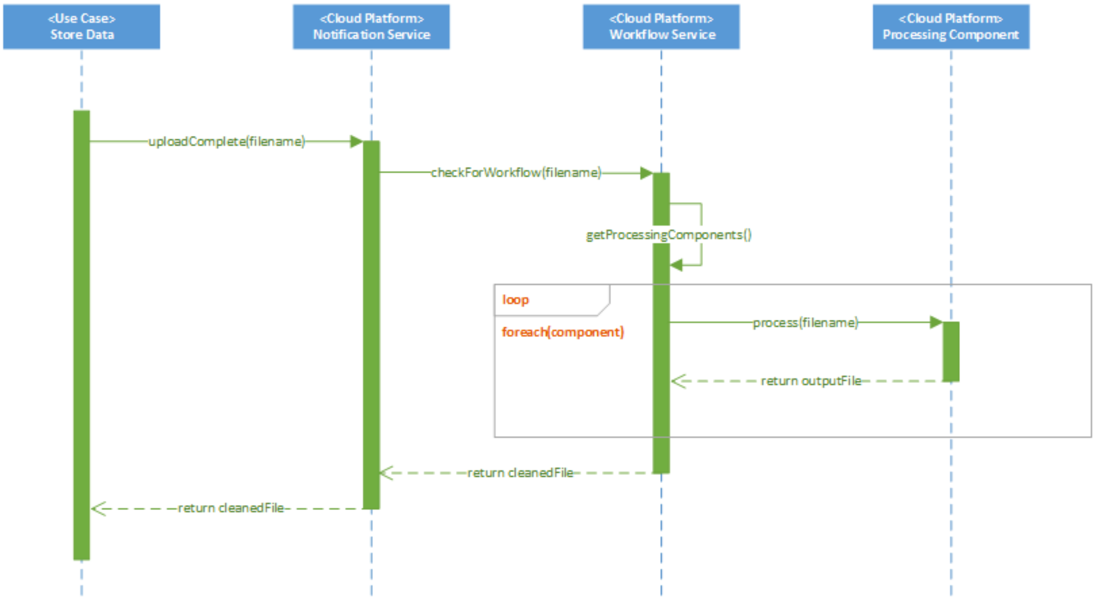



**Figure 92 Sequence diagram for clean data use case**

## Expose data sequence diagram

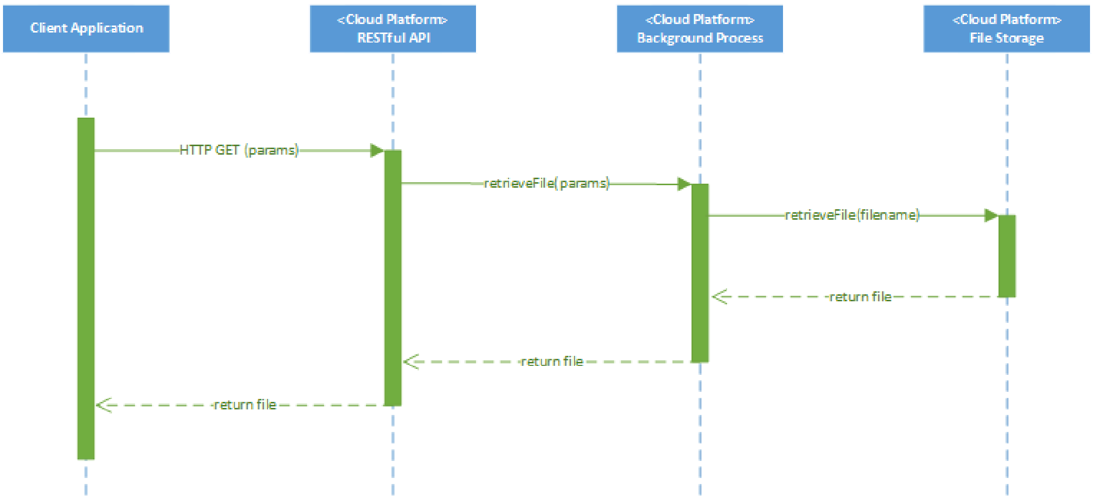



**Figure 93 Sequence diagram for expose data use case**

## Access data sequence diagram

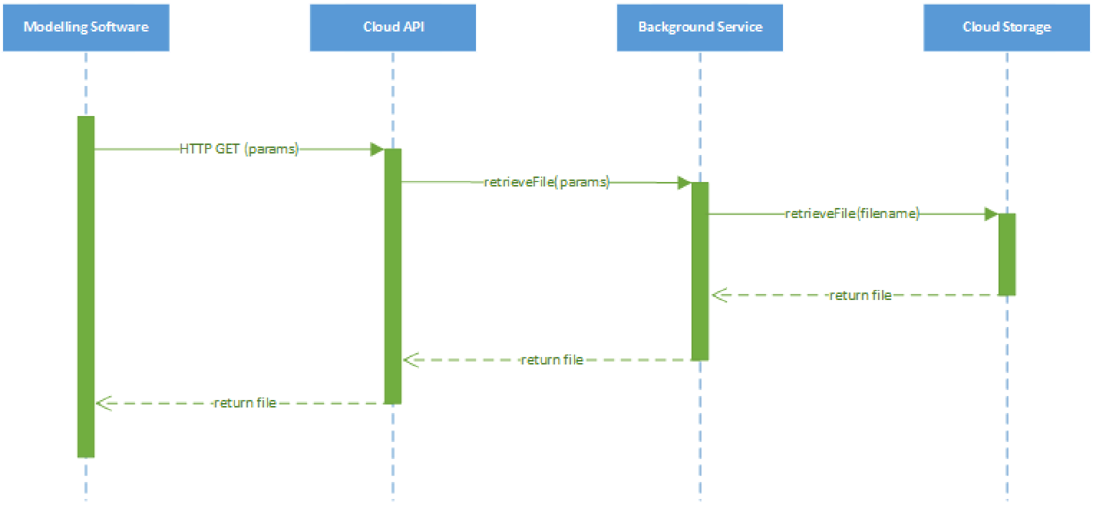



**Figure 94 Sequence diagram for access data use case**

## Build model sequence diagram

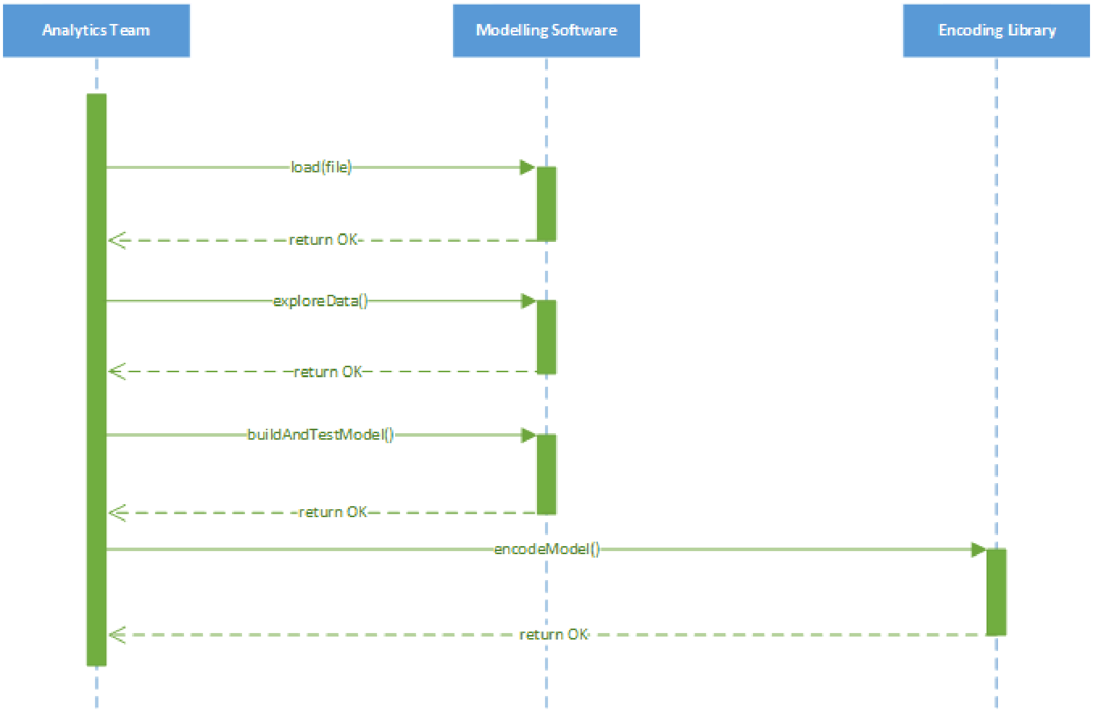



**Figure 95 Sequence diagram for build model use case**

## Deploy model sequence diagram

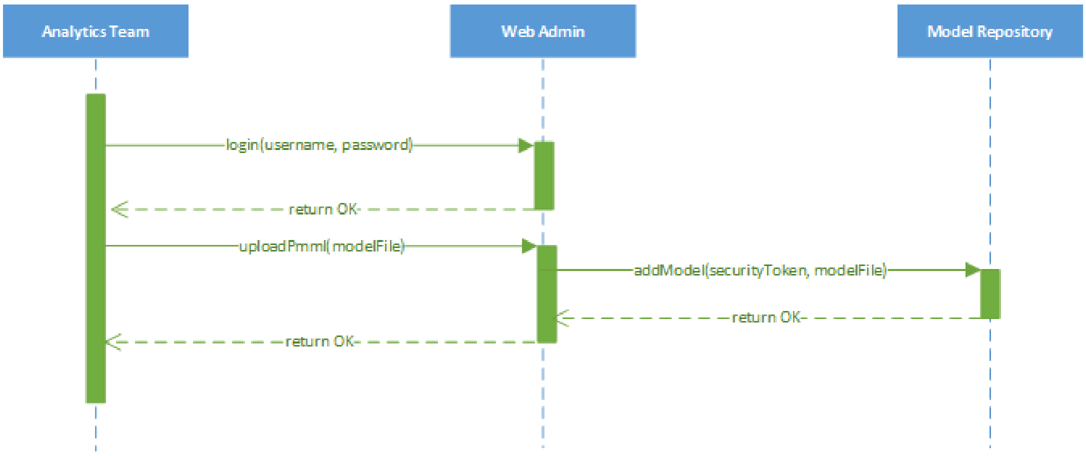



**Figure 96 Sequence diagram for deploy model use case**

## Stream data sequence diagram

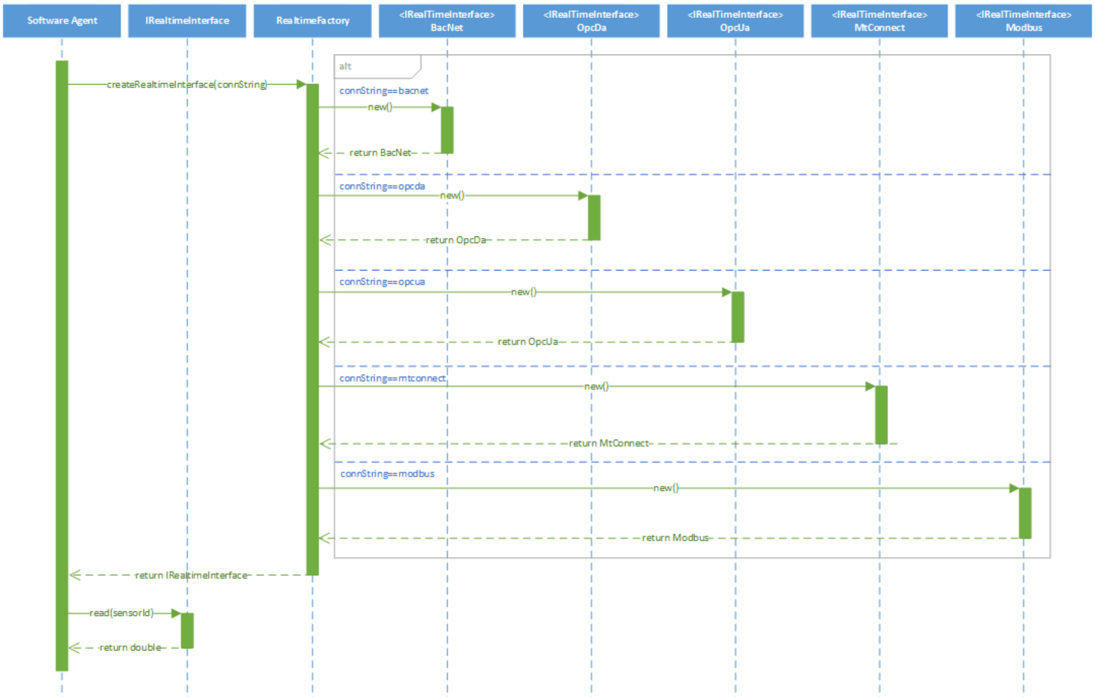



**Figure 97 Sequence diagram for stream data use case**

# Score model sequence diagram

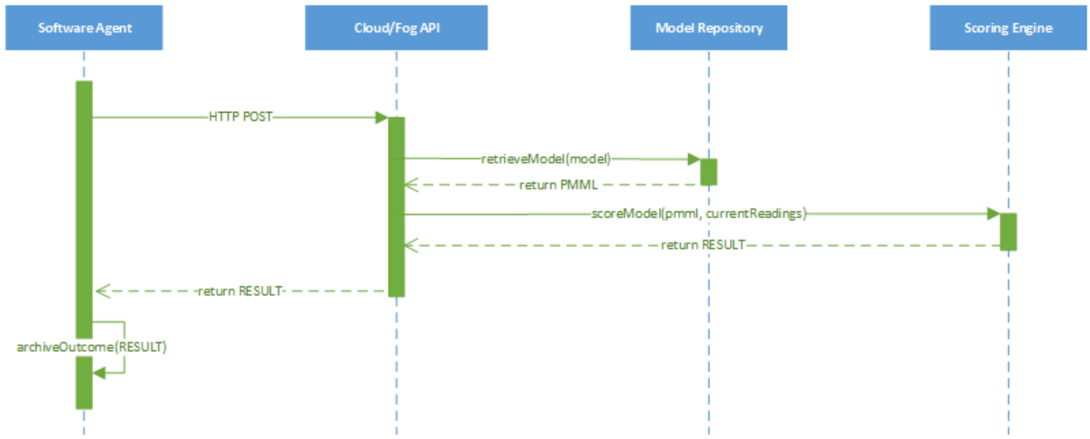



Figure 98 Sequence diagram for score model use case

# Relay score sequence diagram

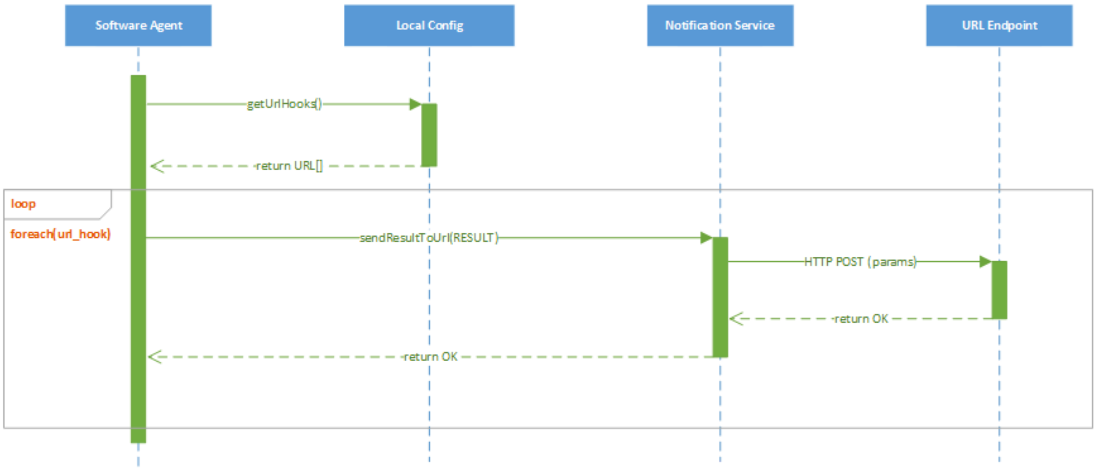



Figure 99 Sequence diagram for relay score use case