공학박사 학위논문

Development of Performance Measurement System using Internet of Things

사물인터넷 기반 성과 측정 시스템에 관한 연구

2017 년 7 월

서울대학교 대학원

산업공학과

황 규 선

Development of Performance Measurement System using Internet of Things

사물인터넷 기반 성과 측정 시스템에 관한 연구

지도 교수 박 진 우

이 논문을 공학박사 학위논문으로 제출함

2017 년 7 월

서울대학교 대학원

산업공학과

황 규 선

황규선의 공학박사 학위논문을 인준함

2017 년 7 월

위 원 장	문 일 경	(인)
부위원장	박 진 우	(인)
위 원	서 은 석	(인)
위 원	장 태 우	(인)
위 원	정 재 윤	(인)

Abstract

Gyusun Hwang
Department of Industrial Engineering
The Graduate School
Seoul National University

The ability to measure operational performance is an important factor for competing enterprises in the global market. Performance measurement helps in the evaluation of the long term effects of outputs for improving competitiveness and decision-making power. A company's competitiveness and profits are reduced by a consistent continuation of subpar performance, as this eventually leads to a failure to meet customer need. In this overall perspective, using performance measurement to understand the company's circumstances is necessary for the manufacturing system to have rapid reactive ability. Although manufacturing companies have used information systems to manage performance, there has been the difficulty of capturing real-time data to depict real situations. The recent rapid proliferation of Internet of Things (IoT) has enabled the resolution of this problem. With the maturity of IoT devices and databases technology, manufacturers are able to assess productivities and obtain real-time feedback from all production lines through IoT data. As IoT-based environment is well established, Industry 4.0 has evolved. It is the fourth stage of industrialization, and is also referred to as smart factory.

Indubitably, in a smart factory environment, the complexity of information system network has increased, because manufacturing systems consist of multiple servers and client applications. Interoperability among manufacturing information systems is a rising issue for a manufacturer who developed the inter-connected systems and systematic obedience. OPC-UA (Open Platform

Communication Unified Architecture) is a set of industrial standards providing a common interface for communications and represents a method to transmit any kinds of data. This thesis follows OPC-UA standard and explains how IoT data are exchanged among heterogeneous systems. Moreover, complexity of network causes IoT fault. If an IoT fault occurs, the performance measurement results cannot describe the production situation appropriately, because data-driven measurement is strongly connected with acquired IoT data. In other words, a reasonable value for Key Performance Indicators cannot be derived, if the IoT data have an error value. An IoT data anomaly detection and mitigation process is therefore required in response to the problem.

To resolve enumerate backgrounds and problems, the dissertation comprised five steps: (1) Development of an smart factory performance measurement model consistent with the ISA-95 and ISO-22400 standards, which define manufacturing processes and performance indicator formulas; (2) Identification of IoT applicable parts in ISO-22400 standard and selection of the Key Performance Indicators of the Net-Overall Equipment Effectiveness (OEE); (3) Configuration of the smart factory architecture and performance measurement process using Business Process Modelling, and adaptation of data exchange protocol by referencing OPC-UA; (4) Implementation of an IoT fault case classification and data anomaly detection and mitigation algorithm, using k-means and statistical inference methods; and (5) Validation of the proposed system through experimental simulation. The experimental simulation results showed that the proposed system represented the timestamp data acquired by IoT and captured the entire production process. In addition, these results indicated that the proposed data anomaly detection and mitigation algorithm have a positive impact on IoT data anomaly identification, thus enabling the determination of real-time performance indicators.

Keyword : Performance measurement; ISA-95; Internet of Things; OPC-UA; Fault management; Data anomaly analysis

Student Number : 2013-30318

Notice : This dissertation is composed of the published journal articles that are Gyusun Hwang is included as the first author.

[1] G. Hwang, S. Han, S. Jun, and J. Park, Operational Performance Metrics in Manufacturing Process: Based on SCOR Model and RFID Technology, *International Journal of Innovation, Management and Technology*, vol. 5, 2014.

[2] G. Hwang, J. Lee, J. Park, and T.-W. Chang, Developing performance measurement system for Internet of Things and smart factory environment, *International Journal of Production Research*, vol. 55, pp. 2590-2602, 2017.

Excluding those articles, I presented referenced journals following the thesis guideline.

Contents

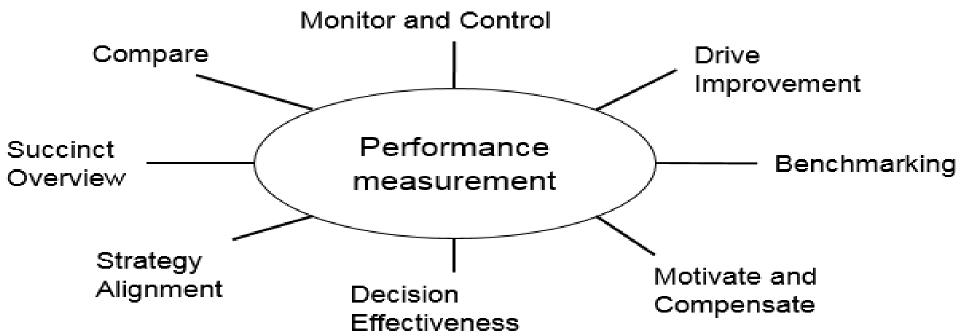
List of Tables

List of Figures

Chapter 1. Introduction

1.1. Performance measurement

Over the last decades, manufacturing processes have evolved from comprising single-stage processes to consisting of multiple-stage processes that utilize multi-level Bill of Materials (BOM). To cope with the consequent complexity, the manufacturing industry demands more rigorous requirements than ever before, such as compliance with low inventories, handling of demand uncertainties, standardization of the manufacturing process, and the development of more complex products [3, 4]. The monitoring and controlling of associated factory facilities and operational performance have thus become essential to manufacturing company, for effective management and enhanced productivity. Performance measurement is indispensable for managing the state of the system and taking the appropriate actions for maintaining company's competitiveness and rapid responsiveness. Many companies conduct performance measurement for measuring, evaluating, and monitoring their operations of the entire activities [5].

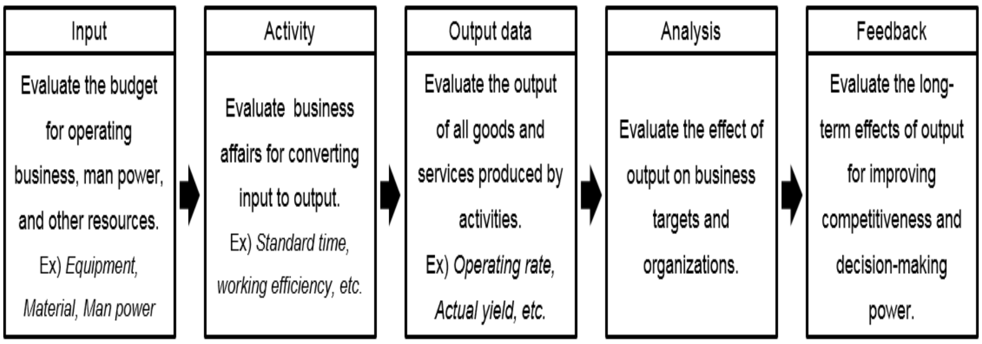


<Figure 1.1> Functions of performance measurement

There are some important criteria for effective performance measurement.

- The measures should link between strategy, execution, and value creation. In addition, the measures should indicate the comprehensive performance of activities. Performance measurement takes into account the overall company's activities. In order to achieve this, measures should align the activities from business to operations.
- The measures should involve intangible dimensions of performances. In general, on the point of shop floor level workers and human resource managers, there are many aspects concerned with intangible performance, such as the effectiveness of the scheduling, human activities, etc.
- The measures should capture the reality adequately. As the circumstance of the company's environment is dynamic status, measures could be obsolete. This obsolete prevents the measuring the performance effectively.
- The measures should be observable and measurable which have quantitative terms. This criterion ensures that measures can be applied to the analytic method.

To measure the performance, well-extracted performance indicator are important sources for the effective performance measurement, because most of the performance measure is evaluated by metrics. A production line manager may evaluate performance by analyzing the Key Performance Indicators (KPIs), which are used to quantify the efficiency and effectiveness of actions in a part of, or the entire production process, or of a system relative to a target pattern [6]. It is generally believed that inspecting all the processes of the company environment yields well-extracted that can increase the chances for success [7]. Before the 1990s, financial measurements such as firm revenue, market share, and return on investment, were the main methods for evaluating performance [8]. However, some shortcomings of this method have been discovered, i.e., the fact that it is easy to concoct and falsify financial measures. Moreover, especially in the manufacturing domain, it is easier for managers to focus on reducing cost. This causes a deterioration of quality and disturbs long term improvement. In this reason, this thesis deals with non-financial key performance indicators in accordance with international standard. Extracting the effectiveness metrics follows this procedure: First, identify the defining elements and different metrics. Second, position the metrics within the operations management research environment. Finally, identify the special research challenges associated with metrics. Finally, introduce the issues that comprise the special findings [9]. Considering extracted metrics, performance measurement flows denoted figure steps.



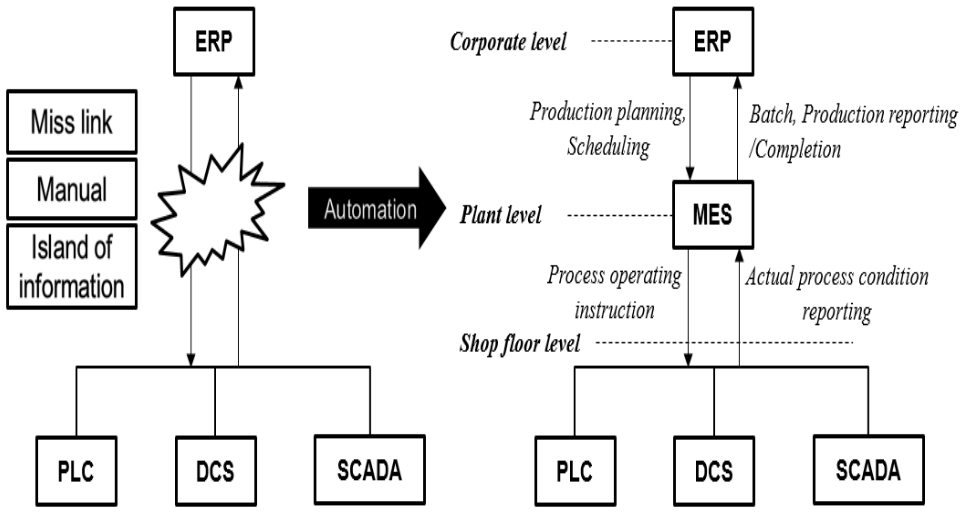
<Figure 1.2> Flow of the performance measurement

Until now, there have been various performance measurement methods for the evaluation of manufacturing performance in theoretical and empirical studies. Kaplan and Norton [10] developed a Balanced Score Card (BSC), which answered the call for a multi-source performance measurement system that used both non-financial and financial strategic indicators. From the BSC, many methods, such as Supply Chain Operations Reference (SCOR) and business excellence models, have been derived. The Baldridge, Deming and European Foundation for Quality Management (EFQM) model are the most well-known and commonly used models throughout the world [11]. Many organizations have adopted business excellence models because of the realization that these models promote the adoption of the best practices and tools that best fit a successful quality strategy [12]. Additionally, these models have also been established as the holistic approach to organizational performance measurement. This dissertation will identify the need of the development of a systematic IoT-based production performance measurement system that utilizes acquired IoT data and planning information. The implementation of the proposed performance measurement model will employ two international standards, ISA-95 and ISO-22400, and it will be demonstrated that IoT devices could be used to measure the production performance.

1.2. Manufacturing information system

Performance measurement has been used in more recent times to improve manufacturing systems, and practitioners and researchers have employed many different methods [13, 14]. As corporations increasingly use information systems to enhance productivity and measure performance, the employed software architecture and data consolidation methods have become more complex, with the structure of the small applications varying with time. The difficulty of integrating multiple point systems has caused software providers to package multiple execution management components into single and integrated solutions [15]. This solution, referred to as Manufacturing Execution System (MES), mainly focuses on the management of shop-floor operations such as material delivery and consumption, as well as production progress [16]. Owing to their tangible and intangible benefits, MES has been adapted and adopted in several manufacturing fields.

As <Figure 1.3> presents, MES is a control system for managing and tracking work-in process on a production line, and providing updated information to Enterprise Resource Planning (ERP). Before the development of the MES, there is a miss link between corporate level information system called ERP and Shop floor level information systems such as Programmable Logic Controller (PLC), Distributed Control System (DCS), and SCADA (Supervisory Control And Data Acquisition). For this reason, workers should put the acquired shop floor level data into the ERP systems manually and also some data was not shared with other systems due to lack of network connections.



<Figure 1.3> Architecture of manufacturing information system

To resolve those problems, MES was developed and plays a vital role as a bridge between ERP and the lower level manufacturing information system. As MES plays a key role in a manufacturing system, international standards called ISA-95 and ISO-22400 were developed. Those standards present a standard of MES enterprise-control integration, define manufacturing process and communication mechanism, and integrate business logics to the manufacturing system.

1.3. Internet of Things and smart factory

As advanced concept, namely Internet of Things (IoT), was recently developed and internet infrastructure is well established in the production line, the broad-adoption of the IoT provides information about the physical world and allows interaction with real situations. IoT is the networking paradigm with embedded smart sensors and continuously generate data for situational awareness. These concepts are not entirely new and emerged in a context of ICT (Internet Communication Technology) several years ago [17]. To improve the rapid responsiveness and high performance of firms, real-time manufacturing tracking is needed and IoT, a promising networking paradigm, plays a significant roles in solving these problem [18]. The basic idea of the concept is the wide application of industrial IoT systems such as Radio Frequency Identification (RFID) tags, sensors, actuators, and mobile phones [19]. Different types of IoT devices have been extensively used in the manufacturing domain, playing a very important role in monitoring and controlling process and operations [20]. The use of IoT to achieve smarter manufacturing and performance measurement has constituted a critical step in the industry. The application of advanced networking technologies and the adoption of devices with enhanced capabilities in the shop floor environment have afforded communication and computation capabilities in a large number of materials and enabled their interaction with MES. The use of IoT for shop floor management is facilitated by the fact that the technology can be installed in a limited area such as a production line, stocking line, or packing line. An important driving forces for adoption of IoT devices and related information systems is the possibility to integrate a large variety of devices and new services into existing legacy systems [21].

As the adoption of the IoT and integration with manufacturing information system increases, the concept of smart factory is evolved. In a conventional manufacturing system, data requests are driven by events or are initiated periodically, always in response to requests from the client level. Within smart factory, IoT devices are spreads around production lines, and this situation allows the manufacturing company to have a better understanding of the activities on the production line. The core idea of smart factory is to use emerging technologies to implement an IoT-based manufacturing system and a manufacturing information system, so that business processes and production operations are deeply integrated, thus making production process a flexible and efficient with constantly high quality and low cost [22]. In a smart factory environment, IoT devices will be built around machines, storage and production lines, and data streaming would be done. The main function of a smart factory is to improve the overall manufacturing performance by generating valuable information from the acquired IoT data. The acquisition, analysis, and application of data in manufacturing systems are vital for the success of the smart factory. One of the most significant components in the development of smart factory is represented by Cyber Physical System (CPS). CPS is defined as transformative technologies for managing interconnected systems between its physical assets and computational capabilities [23]. Unlike traditional embedded systems, which are designed as stand-alone devices, the focus in CPS is on networking several devices so that it is inevitably core components of the smart factory [24]. CPS combines the cyber aspects of computing and communicates with the dynamics and physics of physical systems operating in the real world. As a consequence, CPS in a communication networks crosses a frontier of the interaction between the physical and cyber world [25].

MES that gathers, processes, and transmits acquired CPS data to upper level information system has the radically influence because of the broad range of CPS data. Even though existing MES has a responsiblility for data management issues, it has a weak point in that it does not meet the needs of comprehensive monitoring and management of processes and of tracking the manufacturing status, due to the absence of consideration of IoT data management [26]. In addition, MES faces all the classic big data challenges in more extreme forms; these include increasing volume, broader variety, increasing complexity, rapid changes, and more challenges of veracity to ensure the development of trust [27]. This problem results from the situation that there may be large volumes of data stored in the database, because IoT devices constantly transmit identified objects. In this reason, this dissertation proposed the performance measurement system which has a data exchange protocol and management module and business logic for deriving a valuable KPIs from acquired IoT data.

Chapter 2. Overview of this dissertation

2.1. Problem definition

The four problems of this dissertation are defined as follows:

Problem 1: There are no related researches which explain how IoT is a source of the performance measurement and state which KPIs need IoT data as Key Performance Indicator.

As the adoption of the IoT and manufacturing information system increases, the complexity and interactivity of communications within the information systems is expected to expand. In this reason, most industries and manufacturers are reluctant to implement IoT-based manufacturing information systems due to unclear benefits, a lack of clear implementation details, and a seemingly large investment requirements. Note that, in the development of the smart factory environment, establishing process standards is more time and cost consuming than installing the manufacturing information system. Moreover, incomplete business processes may cause low system performance and deteriorate the reliability of the production result analysis. This highlights the need of this dissertation, which shows the fundamental performance measurement process considering the international standards.

Problem 2: Huge number of incompatible data exchange protocols are used for data acquisition.

The IoT devices and related independent system such as CPS communicate with the manufacturing information system by connecting machines, data-warehouse, and network components. To facilitate the robust data communication, data exchange protocol which is to exchange data among different devices via some transmission protocol is needed. There are a huge number of data exchange protocols such as Ethernet, ProfiBus, Bluetooth, Firewire, Modbus, etc. Using different data exchange protocols among IoT-related information system prevents the universal data usage due to incompatibility. Simplifying the data exchange protocol and presenting scalable way to consolidate the abundant data are required for developing smart factory network architecture. Adapting OPC-UA helps constructing plug and produce environment and interlinks devices, machines, and objects without any human invention to collect the IoT-based data ware house.

Problem 3: The IoT environment is unpredictable and failure prone.

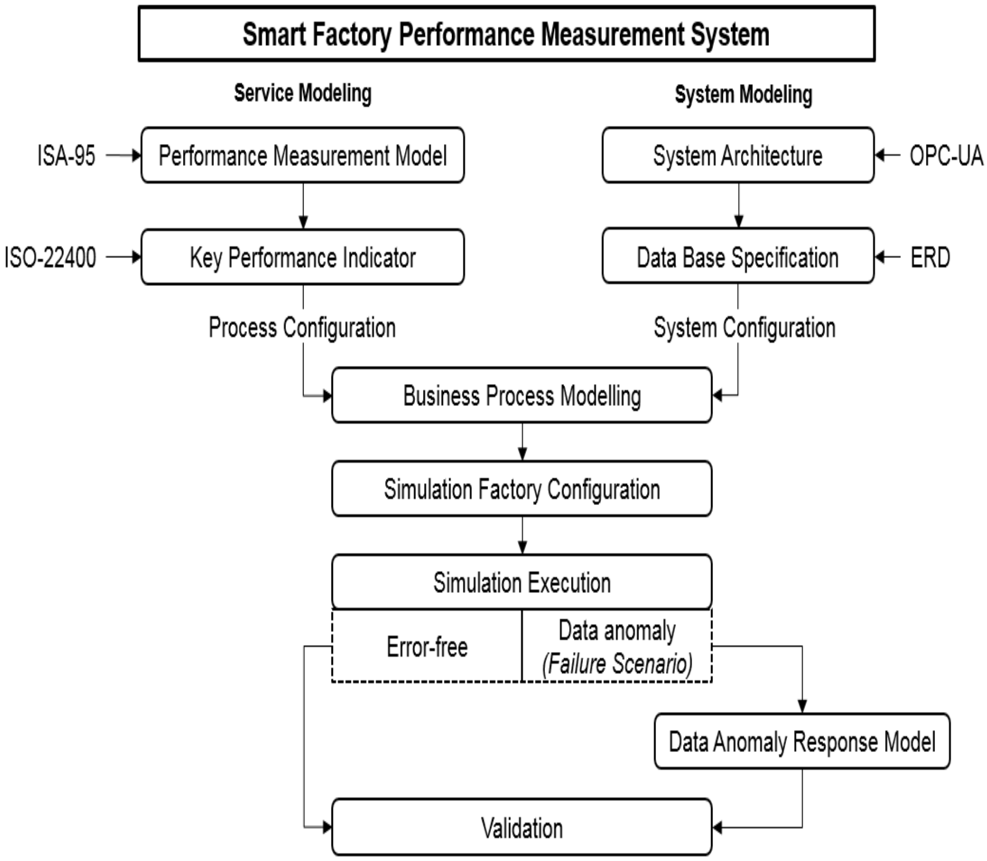
Indubitably, integration of IoT into the manufacturing execution system allows shop floor events to be observed. Historically, the most common means of obtaining situational awareness would be to identify the information that is required for the mission and then to proceed with search routines through printed material, data repositories, user's feedback, activities, etc [28]. This manual process is very time consuming and may not produce timely results. In addition, occasional data anomalies would inevitably occur when IoT devices are broken or during data communication failure occurred. The rate of IoT failure remains high in comparison with information system due to low device reliability and lack of robust data exchange protocol. In these reasons, precise data anomaly detection which refers to the identification of any behavior or pattern that is abnormal or unexpected is essential for the network breakdown to be resolved with minimum down time and high accuracy of performance measurement.

Problem 4: Data-driven performance measurement system is strongly affected by IoT fault.

The KPIs are used to gauge and measure how a manufacturing business is performing, which is a measure of how effectively the operations and business are achieving their defined goals. And KPIs are important source of the performance measurement, because production performance values come from many production related KPIs. In this reason, KPIs based on IoT data could not present a real situation, if IoT data have some problems. Regarding that IoT environment is failure prone as this study denoted at Problem 3, I have to develop IoT failure response model to derive an effective performance measurement.

2.2. Research statement

The general objective of this study was the development of a method for the systematic analysis of an IoT-based production performance model in accordance with ISA-95 and ISO-22400 standard. These two standards outline how a manufacturing process can be formalized and the process for formulating the performance indicators. This dissertation develops a smart factory performance measurement system that integrates the business process and software architecture using the Business Process Management Notification method. Furthermore, this research work is aims to develop a unified process to generate a smart factory performance measurement system by applying the IoT data anomaly response model. The IoT data anomaly response model is executed in the case of IoT data failure. The objective of the response model is to detect an IoT data anomaly and mitigate the impact of the IoT data anomaly using a *k*-means clustering method and a statistical method. This study particularly focus on investigating the relation among the planned and actual and abnormal production data based on the 'Overall Equipment Effectiveness'. Developed simulation factory shows how suggested performance measurement model and data anomaly response model are executed and result analysis validate the effectiveness and application of suggested models. To resolve stated four enumerated problems, the procedure of this study and four contributions of the study are as follows:



<Figure 2.1> Research model

- Analysis of the ISA-95 and ISO-22400, which are related to MES. I also defined applicable IoT parts based ISO-22400. This dissertation particularly focused on one KPI, namely, Net-OEE, the calculation of which is based on a parameter that is detected by IoT devices. One of the core contributions of this work is the presentation of a means of converting highly granular data obtained by IoT devices into meaningful KPI.
- Specification of the architecture of an IoT-based smart factory performance measurement system. This study applied Business Process Management Notification for developing a

system that integrates the production performance model from the ISA-95 and architecture of the smart factory performance measurement system. In addition, this dissertation described OPC-UA standard for following the data exchanging protocol.

- Implementation of an IoT fault case classification and data anomaly detection and mitigation algorithm. By providing a list of the most common IoT data anomaly cases, the IoT data anomaly detection algorithm is more robust in identification of the data anomaly situation. This study also developed the mitigation algorithm in accordance with anomaly data types.
- Presentation of the results of an experimental simulation and acquired performance data and log files, as well as the use of the simulation outputs to verify the proposed performance measurement process. This study simulated the normal error-free situation as a basis of comparison among the normal, abnormal, and mitigation applied situation.

2.3. Literature reviews and outlook of the dissertation

In recent year, numerous studies have attempted to develop a production-focused performance measurement system [29-34]. Performance measurement has been a traditional research topic, since the necessity of measuring performance was recognized for evaluating the production efficiency in production management. Previous studies proposed a measuring process and framework, and analyzed whether traditional key performance indicators and measuring process presented manufacturing situation well. However, relatively few studies have applied the IoT technology concepts to the performance measurement and did not identify whether IoT is applicable for the source of the performance measurement or not.

Business Process Management method is used to specify the business process in the present study, because it is widely accepted and enables pre-designing and evaluation of the business flow before the actual implementation [35, 36]. BPM presents a flexible approach for aligning business process models with workflow specification [37]. There have been some proposals for the functional modelling of an MES using BPM and integrating the ISA-95 standard [38, 39]. However, relatively few studies have applied BPM to an IoT information system. This dissertation identified that only one paper dealt with the IoT concept using BPM [40].

<Table 2.1> Related studies of function modelling using BPM

Author	Year	Key components	Standard	Outputs
Desbiens *et al.*	2014	MES, ERP	BPM, ISA-95	Integration model
Prades *et al.*	2013	MES	IDEF0, BPM, ISA-95	Conceptual framework
Meyer *et al.*	2013	IoT devices, S/W	BPM	Process meta model

Desbiens and Chaabane [39] analyzed the business process of a manufacturing company in accordance with ISA-95 and BPM. They focused on the description of the methodology for designing and implementing a BPM based on ISA-95. Prades, et al. [38] presented a conceptual framework for modelling a manufacturing company that uses BPM and ISA-95. The main contribution of the paper was the description of the correlation between ISA-95 and BPM notation. Meyer, et al. [40] investigated how IoT devices and the associated software can be expressed as a resource using BPM. They proposed the use of a process meta-model and presented a general semantic model for capturing resource allocation. However, there has been no work on the application of BPM to performance measurement using IoT and MES standard. We therefore had no foreknowledge of how IoT data could be processed for performance measurement, or of which KPIs could use IoT data for production performance evaluation. Nevertheless, I attempted to develop a method for directly integrating multidisciplinary information with IoT devices in a business process and present systematic architecture considering business process and hierarchical structured manufacturing system.

ISO-22400 standard has been used for the source of the performance measurement in production related studies [21, 41-44]. Theorin, et al. [21] developed Line Information System Architecture (LISA) which is an integration of devices and services on all levels, simplifying hardware changes and integration of new smart services in accordance with ISO-22400. Helu, et al. [41] designed data-driven decision making model for smart manufacturing and ISO-22400 is used for manufacturing operation management. Bauer, et al. [42] described the practical integration between scheduling and control by referencing ISO-22400 and especially took example of two KPIs: efficiency and energy efficiency. Yoon, et al. [43] proposed a reference architecture for the information service bus or middleware for the smart factory. They used Total Performance Index as a generic smart factory KPI. Productivity based on ISO-22400, environment, and social impact are components of the Total Performance Index. Most studies only referenced ISO-22400 standard as a source of the KPI and only Yoon, et al. [43] and Theorin, et al. [21] studies considered IoT technology and smart factory concept for application of ISO-22400.

Many previous studies have been published in relation to the definition of OEE and its various applications [45-47]. Tsarouhas [45] derived OEE of the beverage production line over a period of 8 months and evaluated the hand-written record data failure of OEE. Marcello, et al. [46] described the OEE drawback in that OEE can only measure the efficiency of individual equipment installed in a productive facility and proposed a new efficient metric called OEEML (OEE Manufacturing Line) which successfully highlights the progressive degradation of the ideal cycle time. Kang, et al. [47] investigated 34

KPIs which are introduced in ISO-22400 and introduced a multi-level structure for identification and analysis of KPIs and their intrinsic relationships in production systems. The main distinction of the present work is the use of IoT devices that can detect the degree to which a process accomplishes its purpose on the shop floor, and the investigation of how the Net-OEE and other KPIs, vary IoT data that define the relationship between a planned schedule and the actual production operation.

A promising system modeling standard called OPC-UA is a noticeable research topic in recent days. Many studies used OPC-UA for presenting the architecture of the smart factory [48-51]. Henßen and Schleipen [48] simplified the creation of OPC-UA information models based on existing Automation Markup Language data by examining the analogies between Automation Markup Language and the OPC-UA information model. In this research, Automation Markup Language which presents the data exchange format combined with OPC-UA and analogies and differences between two standards are discussed. Seilonen, et al. [49] presented a design of an aggregating server based on OPC-UA and tests it with two different experimental applications. They proved that suggested design enables transparent access to the data. Rentschler, et al. [50] developed OPC-UA extension of IP auto-configuration in Cyber Physical Systems and compared OPC-UA benefits with DHCP protocol. They described that both mechanisms are similar communication but OPC-UA has more benefits, because it is easy to integrate and migrate of legacy systems. Schleipen, et al. [51] developed experimental OPC-UA based information system using various development resources such as Windows, Linux, and Raspberry Pi, etc. They concluded that OPC-UA can be used as data and information hub for dedicated purposes and the system based on OPC-UA is much more flexible compared to a simple applications. In summary, previous studies developed information model and system architecture using OPC-UA. They also concluded that OPC-UA has a strength in designing complex architecture and presents an extensible way to consolidate into legacy system. Our dissertation, therefore, applied OPC-UA to take data interoperability situating in the middle of MES and ERP.

There have been some proposals for methods of IoT fault identification and management [52-57]. Oh [52] simply presented possible causes of sensor malfunctions as follows: sensor failure or aging, node damage due to impacts, node battery exhaustion, and other errors on the board. Wang, et al. [53] analyzed the healthy index of devices in real time that faults of certain devices can be detected through the conditional probability analysis, which makes it possible to predict faults of other highly related devices based on the causual relationship between them. Liu, et al. [54] presented a self-learning sensor fault detection framework using group based fault detection algorithm. Misra, et al. [55] developed mixed cross layered and learning automata algorithm to assure successful anomaly detection, even in the presence of faults between a pair of source and destination nodes.

In summary, a developing performance measurement system has been a conventional topic since 1990s and especially most studies mapped performance measurement system into MES using international standard such as ISA-95, ISO-22400, etc. However, there are few studies focusing on IoT as a source of the performance measurement. For this reason, simultaneous consideration of performance measurement and IoT technology is imperative to having an accurate performance measurement system in industry 4.0 era. Many studies considered OEE as a fundamental KPI presenting the overall production performance. In addition, ISO-22400 standard is considered for making coherent OEE calculation. In general, as manufacturing information systems have become more complex than ever before, OPC-UA is widely regarded as a tool for data interoperability. However, I acknowledged that most studies only presented data information modelling and, therefore, did not describe overall processes that are from acquiring IoT log files to storing the database in detail. Finally, in the IoT fault topics, most studies are limited to enumerating the IoT fault cases and develop concept model of IoT fault response.

The remainder of this dissertation is organized as follows. Chapter 3 describes the smart factory production performance model and proposes a measuring process in accordance with international standards. Chapter 4 presents the architecture of our proposed system and a BPMN for integrating the architecture and the performance measurement process and, moreover, describes the OPC-UA standard. Chapter 5 enumerates the IoT data anomaly cases and presents data anomaly detection and the mitigation algorithm. In Chapter 6, a proposed experimental simulation factory is presented and its simulation and analysis are described. Chapter 7 discusses findings, further research and final conclusions. Following <Table 2.2> denotes the outlook of this dissertation.

<Table 2.2> Organization of the dissertation

Objective	Chapter	Standards/ Methods	Outputs
Building performance measurement system	Chapter 3	● ISA-95 ● ISO-22400	➢ Performance measurement model ➢ Target KPIs
	Chapter 4	● BPMN ● OPC-UA	➢ Business logic ➢ Architecture
Developing data anomaly response model	Chapter 5	● *k*-means ● Statistical	➢ IoT fault cases ➢ Data anomaly detection and mitigation algorithm
Validating proposed system	Chapter 6	● Simulation and result analysis	➢ Comparison analysis using Net-OEE values

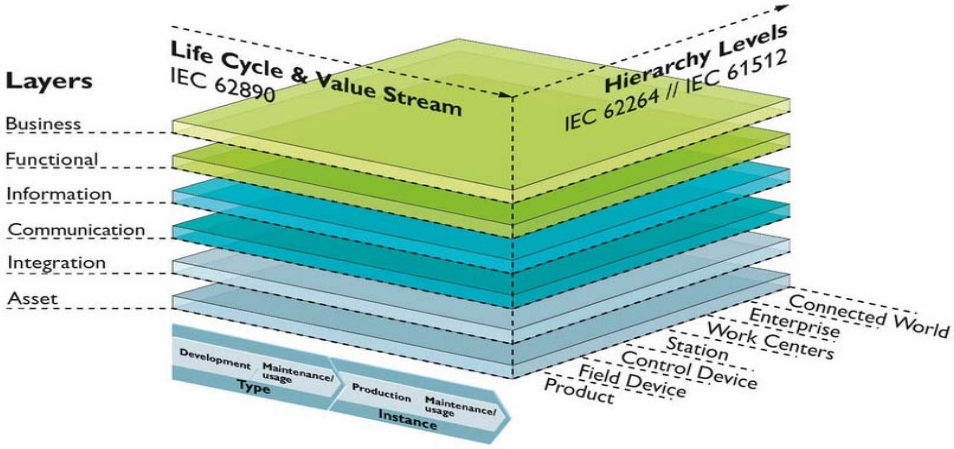
Chapter 3. Development of smart factory production performance model

3.1. Introduction of international standards

3.1.1. Introduction of ISA-95 (IEC-62264)

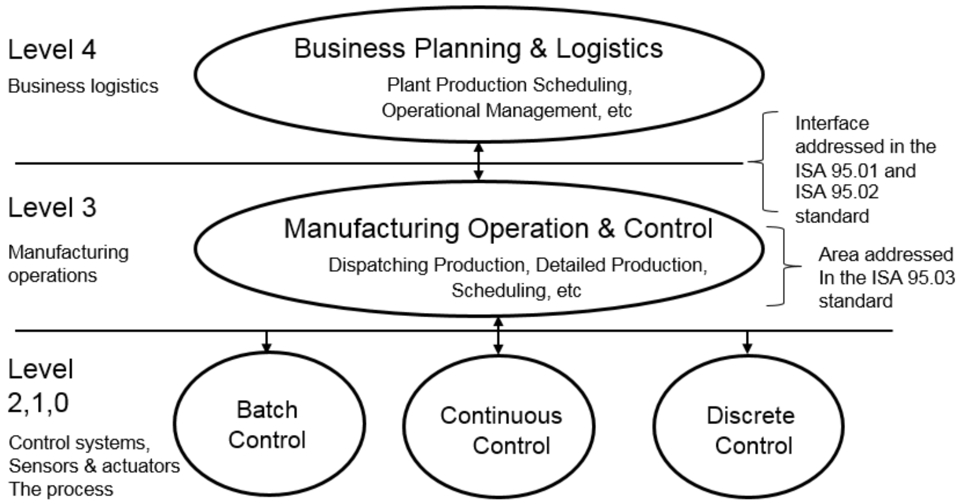
Integration of Enterprise Resource Planning (ERP) with the shop floor level information system such as PLC, HMI, etc., is the one of the difficult problem for Information Communication Technology (ICT) providers. Lack of bridging between business level and shop floor level is resolved by the Manufacturing Execution System. The Manufacturing Enterprise Solutions Association (MESA), International Society of Automation (ISA) and International Electro-technical Commission (IEC) collaborated to develop the MES standard called ISA-95/IEC-62264 [58].

This standard has lead more attention, because Reference Architectural Model Industrie 4.0 (RAMI 4.0), which precisely describes the vital elements of a smart factory using dimensional layers, currently uses ISA-95 to consider different MES hierarchy levels. As <Figure 3.1> presents, the RAMI 4.0 model broadens the hierarchical levels of ISA-95 by adding the product or workpiece level at the bottom, and the connected world goes beyond the boundaries of the individual factory at the top [59]. In other words, while coverage of ISA-95 is limited in the single independent factory, RAMI 4.0 considers not only managing small IoT devices, but also migrating comprehensive manufacturing eco-system.



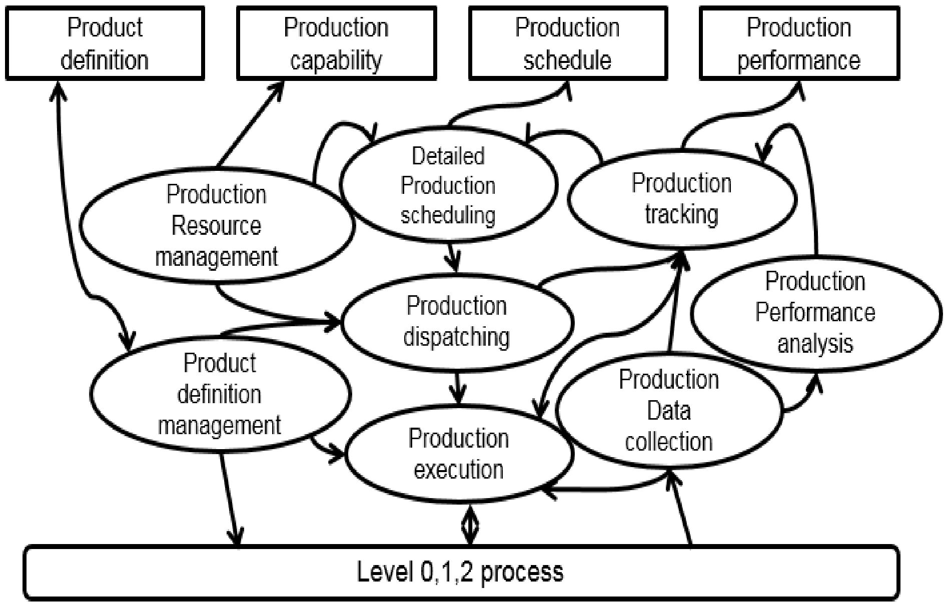
<Figure 3.1> Reference Architecture Model for Industrie 4.0

The major contribution of ISA-95 is that it specified the data flows and interfaces between upper and lower-level manufacturing systems [60]. ISA-95 is to formalize the interactions of the manufacturing system to other business process of the company and define the manufacturing control operations to enterprise level activities. It provides solutions for simplifying the integration difficulties between the enterprise and process control levels by not only dividing the activities of the manufacturing control to groups with clear boundaries and responsibilities but also by defining the terminology and the contents of the exchanged information, thus allowing the different systems to communicate using a standard language [61]. <Figure 3.2> depicts the different levels of a functional hierarchical model: business planning and logistics, manufacturing operation and control, and batch, continuous, or discrete control [58]. Level 2, 1 and 0 is composed of physical processes, sensing and manipulating the physical process, and monitoring and controlling the physical process. This level includes shop floor system such as PLC, DCS, etc. Level 3 is in charge of managing the activities of the workflow to produce the desired final products so that MES is situated in this level. Finally, level 4 defines the business related activities and corresponds physical and logically to the ERP. Our proposed smart factory performance measurement system can be embedded in ERP or MES or separate it them, based on the type of installation required. In other words, our system could be defined as an engine or add-on, because proposed system database is derived from the ERP and MES. Section 4 describes this topic in detail.



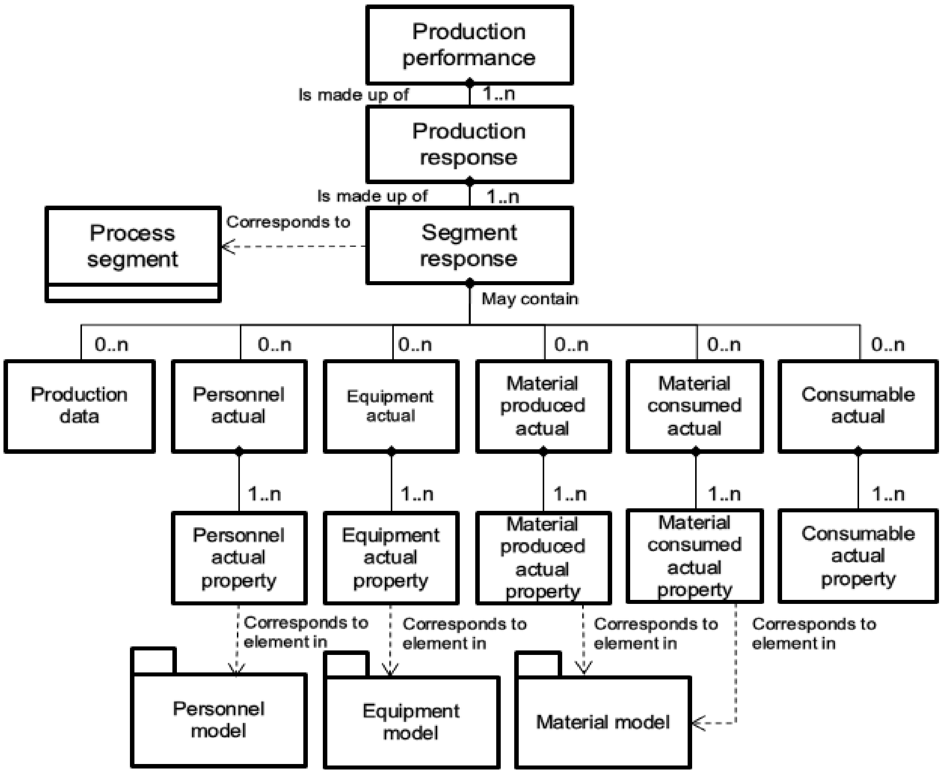
<Figure 3.2> Functional hierarchy defined in ISA-95 standard

ISA-95 also shows the activity models and data flow for presenting the integration of related manufacturing information systems. It includes the activities of managing information about the schedule, usage, capability, definition, history, and status of all of the resources (personnel, equipment, and material) within the manufacturing facility [62]. The most important activity model defined in ISA-95.03 at MES level is presented in <Figure 3.3>.



<Figure 3.3> Activity model of production operations management

There are four generic information exchange between MES and ERP: Product definition, Production capability, Production Schedule, and Production performance. Product definition means what must be defined to make a product. This function includes Bill of Material (BOM) from the ERP production planning module so that sub-products and required materials are decided. Production capability includes what resources are available to produce product. This part manages all resources such as machines, human resources, and information systems to check whether manufacturing operation is available. Production schedule indicates what to make and use from the ERP production planning module. This function is a collection of production work sequence and assigned production time for each resources. Finally, production performance, which is our target activity, means what was made and used for production. Based on collected data from level 0, 1, 2 components, this part analyzes the production performance in accordance with pre-defined measurement formulas. This study follows the procedure of production performance function considering each of the sub-functions: Production data collection for IoT data, Product definition management for Bill of Material (ERP), Production resource management for IoT network architecture and machines, and Production tracking for IoT data flow. ISA-95.02 provides a generic performance analysis process and production performance model to measure performance using the defined Unified Modelling Language (UML).



<Figure 3.4> Production performance model in ISA-95

<Table 3.1> Attributes of production performance

Attribute name	Description	Example
ID	Unique ID	2016-11-03-A15
Description	Additional information	Report on Nov.03, production schedule
Production Schedule	Specific expected time	2016-11-03-A15
Start time	Actual starting time	11-04-2016
End time	Actual ending time	11-07-2016
Current status	Status of performance	In work
Location	Associated equipment	#8 production line
Element type	Type of equipment	Production line

Production performance is a report on requested manufacturing and is a collection of production responses which are responses from several production requests [58]. A segment response shall be made up of zero or more sets of information of lower part. The production performance model presented in the ISA-95 is limited by the fact that it only includes a list of domains and their attributes, without specifying any process. Hence, Section 3.2 attempts to integrate ISA-95 and ISO-22400, which introduce some KPIs and data, as well as production performance. It enables the easy understanding of KPI flow and the potential application of IoT.

3.1.2. Introduction of ISO-22400

Performance measures can be obtained through a combination of various operational measurements, i.e., KPIs. KPIs play vital role of enabling the use of an enterprise value to identify a real production situation relative to certain operational objects, because well-defined KPIs enable identification of performance gaps between the current and desired operations, and can be used to track progress towards closing the gaps [63]. The productivity resources in production lines could be calculated through the KPIs which present a proxy understanding of a situation, and provide decision-relevant information for manufacturers. Many manufacturing related KPIs are based on data from production lines. Hence, low-level events should be transformed and updated to a more standardized structure with attribute names and semantics based on the international standard called ISO-22400 [21]. This standard defines the application of KPIs, which are presented with their formulas and corresponding elements. It specifies 34 standard KPIs used in manufacturing operations management (MOM) as defined in IEC-62264-1 and suggests other sources and production KPIs and sub-KPIs, including their definitions, formulas, and benefits [64]. Excluding duplicated KPIs, this thesis provides 28 KPIs to investigate the IoT applicability. As <Table 3.2> denoted, ISO-22400 standards is comprised of four parts which specify an industry-neutral framework for defining, comprising, exchanging, using, and designing KPI network for manufacturing operations management. <Table 3.3> and <Appendix> provide the KPI description and formula.

<Table 3.2> Description of ISO-22400 parts

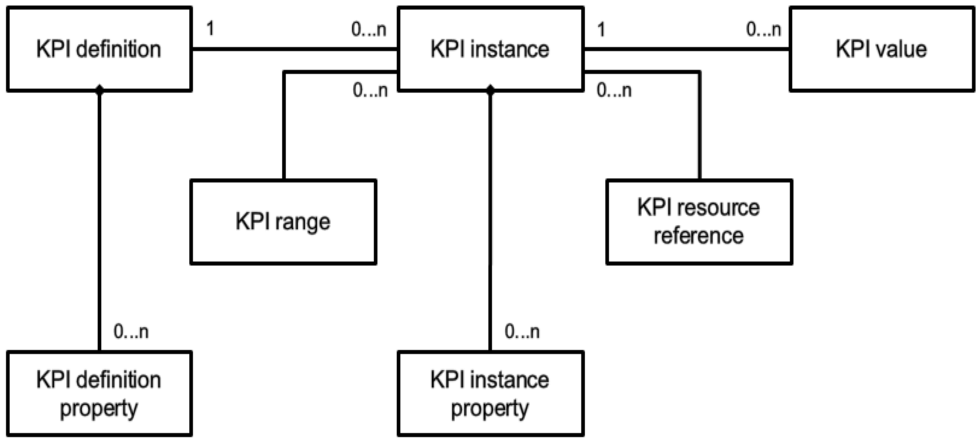
Part	Description
ISO 22400-1	KPI for MOM part 1: Overview, concepts and terminology
ISO 22400-2	KPI for MOM part 2: Definitions and descriptions
ISO 22400-3	KPI for MOM part 3: Exchange and use
ISO 22400-4	KPI for MOM part 4: Relationships and dependencies

<Table 3.3> List of defined KPIs in ISO-22400

Worker efficiency	Allocation ratio	Throughput rate
Allocation efficiency	Utilization efficiency	Overall equipment effectiveness
Net equipment effectiveness	Availability	Effectiveness
Quality ratio	Setup rate	Technical efficiency
Production process ratio	Actual to planned scrap ratio	First pass yield
Scrap ratio	Rework ratio	Fall of ratio
Machine capability	Critical machine capability	Process capability
Critical process capability	Finished goods ratio	Production process ratio
Equipment Load rate	Mean time to failure	Mean time to repair
Corrective maintenance ratio		

<Figure 3.5> is KPI modeling using UML notations described in ISA-95. Each object in the KPI model has as set of associated attributes. The KPI definition property and the KPI instance property correspond to user-defined attributes. Each property has a set of associated attributes. KPI definition is the definition of a KPI which

should be used if the formula and correlated information should be exchanged between two applications. KPI instance is an object such as work-units, persons, and several orders. KPI value is connected with KPI instance which may vary over time [64].



<Figure 3.5> KPI model (UML notation)

Although ISO-22400-1 and ISO-22400-2 describe the KPI definition and measuring processes well, the data which are the source of the KPI should be investigated more rigorously due to ambiguous relations among the data and be identified whether IoT is able to be applied to them. This work outlines how IoT data can be consolidated into ISO-22400 and formalizes the smart factory production performance model as well as smart factory network architecture.

3.2. Identification of key performance indicators

3.2.1. IoT applicable parts in ISO-22400

KPIs are composed of some sub-KPIs, because KPI cannot be derived using only one sub-KPI. Specifically, sub-KPIs can be decomposed into some perspectives: Time elements, logistical elements, and quality elements. Note that, some elements also can be divided into categories based on their characteristics.

Firstly, time elements are related to time duration that is for production plan from ERP, actual operations and maintenance activities from production lines. 34 sub-KPIs are presented at the <Table 3.4> and those sub-KPIs are categorized into three types. Production plan sub-KPIs are the fixed value from the ERP scheduling engine. When the quantity of final products is decided, part explosion would be executed following the Bill of Material. The time table for each production line with buffer time will be allotted and planned operation time will be generated regardless of any possibility of down time. Considering that IoT detects the real situation, IoT cannot be applied to production plan type, because this type is just planning phase. However, actual operation and maintenance activity types can be applied IoT, because these phases are the result of something happens. In addition, all these sub-KPIs are the time-dependent values so that IoT can calculate the time detecting the start and finish time. The following tables provides the sub-KPI description.

<Table 3.4> Description of production plan related sub-KPI [47]

Sub-KPI	Description
Planned Operation Time (POT)	The scheduled time during which a machine can be utilized.
Planned Busy Time (PBT)	The planned time during which a machine is busy.
Planned Order Execution Time (POET)	The scheduled time for executing an order.
Planned Unit Setup Time (PUST)	The planned time for a machine to setup for an order.
Planned Runtime per Item (PRI)	The planned time to produce one piece or part.

<Table 3.5> Description of maintenance related sub-KPIs [47]

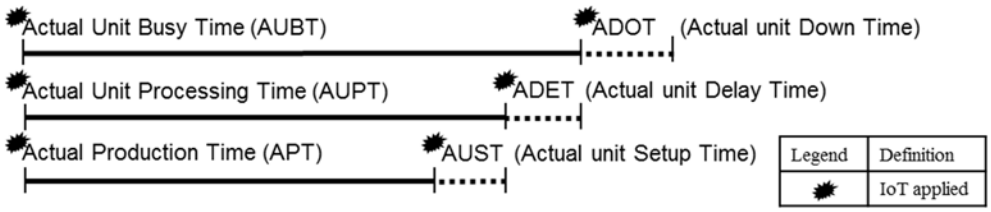
Sub-KPI	Description
Time Between Failures (TBF) = Time To Failure (TTF)	The actual time during which a machine is able to produce, starting from the completion of the repair and ending.
Time To Repair (TTR)	The actual time during which a machine is unavailable due to a failure.
Failure Event (FE)	The count over a specified time interval of the termination of the ability for a machine to perform an operation.
Corrective Maintenance Time (CMT)	The part of maintenance time during which corrective maintenance is performed on a machine.
Preventive Maintenance Time (PMT)	The part of maintenance time during which preventive maintenance is performed on a machine.

<Table 3.6> Description of Actual operation related sub-KPI [47]

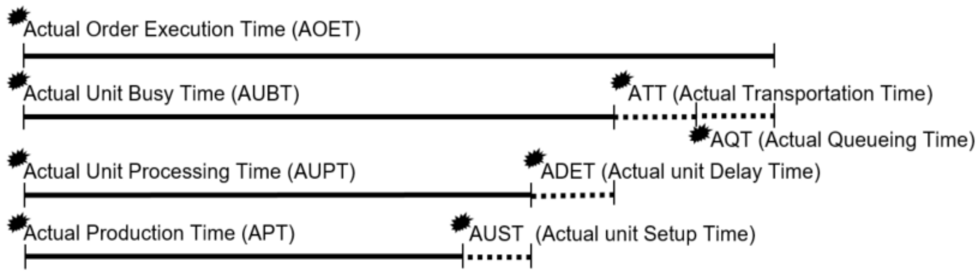
Sub-KPI	Description
Actual Unit Busy Time (AUBT)	The actual time that a machine is used for the execution of a production order.
Actual Unit Processing Time (AUPT)	The time necessary for production and setup on a machine for an order.
Actual Unit Delay Time (ADET)	The actual time associated with malfunction and interruptions.
Actual Unit Down Time (ADOT)	The actual time in which the production process is delayed although it is available.
Actual Production Time (APT)	The actual time in which the machine is producing for an order, which only includes the value-adding functions.
Actual Unit Setup Time (AUST)	The time used for the preparation.
Actual Order Execution Time (AOET)	The time from the start of an order to its completion on a machine.
Actual Transport Time (ATT)	The actual time for transporting parts on or between machines, such as loading or unloading.
Actual Queuing Time (AQT)	The actual time during which the material is waiting to go through a manufacturing process.
Actual Personnel Attendance Time (APAT)	The actual time that a worker is available to work on production orders.
Actual Personnel Work Time (APWT)	The time that a worker needs to execute a production order.

Within the time elements, ISO-22400 presents timeline model from the point of view of work units, production order sequence, and personnel. <Figure 3.6> denotes the loss of operation time such as down time, delay time, and setup time in the basis of reference time.

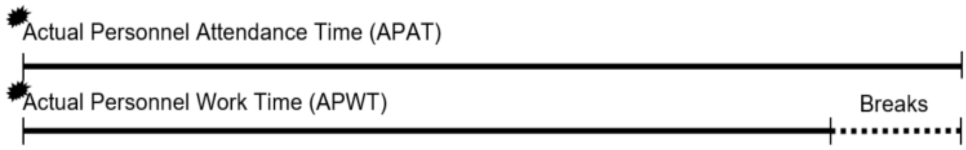
<Figure 3.7> shows the production order sequences timelines consisting of multiple operations. Due to continuous jobs, transportation time and queueing time are considered. The last figure indicates the personnel work that is presenting break time.



<Figure 3.6> Timelines for work units and IoT applicable parts



<Figure 3.7> Timelines for production order sequence and IoT applicable parts

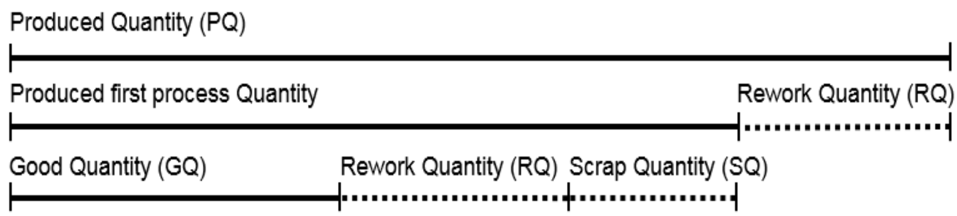


<Figure 3.8> Timelines for personnel and IoT applicable parts

Secondly, logistical elements provide produced quantity information in the range of raw materials to final outputs. In addition, it checks overall losses during storage and transportation activities. All sub-KPIs are able to be included in IoT applicable part.

<Table 3.7> Description of logistical related sub-KPI [64]

Sub-KPI	Description
Planned order quantity (POQ)	Planned quantity of products for a production order.
Scrap quantity (SQ)	Amount of failure of quality requirement.
Planned scrap quantity (PSQ)	Amount of process-related crap expected when manufacturing execution.
Good quantity (GQ)	Amount of passed inspection products.
Rework quantity(RQ)	Amount of subsequent work.
Produced quantity (PQ)	Amount of produced products.
Raw materials (RM)	Materials prior to produce.
Raw materials inventory (RMI)	The inventory of materials that are changed into intermediates.
Finished goods inventory (FGI)	Amount of acceptable quantity which can be delivered.
Consumable inventory (CI)	Materials transformed in quantity or quality during the production.
Consumed material (CM)	The sum of quantity of consumed.
Production loss (PL)	Quantity lost during production.
Equipment production capacity (EPC)	The maximum production quantity of production equipment.



<Figure 3.9> Description of logistical related sub-KPIs

Finally, quality elements are the measure of satisfaction that meets the quality requirement. Good part (GP) is the count of identifiable part that is a single produced item as well as a specified material lot satisfying a quality requirements. Inspected part (IP) is almost same with GP but it is a previous part that waits the inspection process. Those two concept have a count types so that IoT is able to be applicable. Upper Specification Limit (USL) is a value below which performance of a product or process is acceptable. In other words, it is an acceptable maximum value and Lower Specification Limit (LSL) is the opposite concept. Below <Table 3.8> describes the parameter-indicator matrix which also provides the applicability of IoT. 7 sub-KPIs (Italic and underlined in the table) out of 36 sub-KPIs cannot be detected through the IoT so that 8 KPIs (Italic and underlined in the table) which are calculated from each of 8 sub-KPIs cannot be derived through the IoT.

<Table 3.8> Parameter-indicator matrix applying IoT applicability

		Worker Efficiency	Allocation Ratio	Throughput Rate	Allocation Efficiency	Utilization Efficiency	OEE-index	NEE-index	Availability	Effectiveness	Setup Rate	Technical Efficiency	Production Process Ratio	First Pass Yield	*Machine Capability*	*Critical Machine*	*Process Capability*	*Critical Process*
Planned Times (Not available)	PBT																	
	PRI																	
Actual Times (IoT available)	AUBT																	
	AUPT																	
	ADET																	
	APT																	
	AUST																	
	AOET																	
	ATT																	
	AQT																	
	APAT																	
	APWT																	
Quality (Partly, IoT available)	GP																	
	IP																	
	AVG																	
	USL																	
	SD																	
	LSL																	
	ED																	

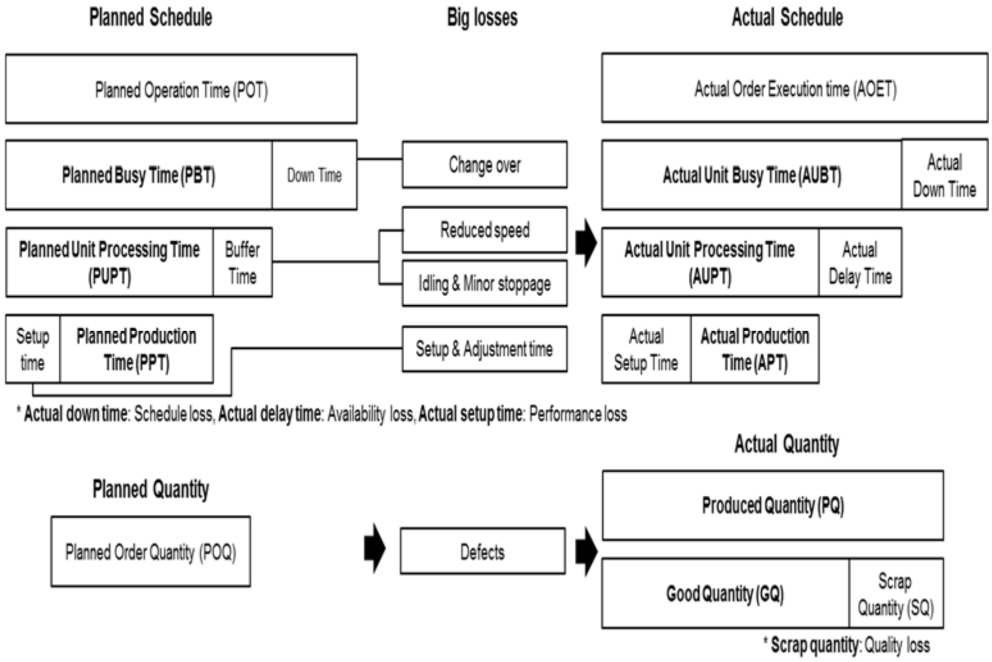
		Throughput Rate	OEE-index	Effectiveness	Quality Ratio	Actual to Planned Scrap Ratio	Scrap Ratio	Rework Ratio	Fall Off Ratio	Finished Goods Ratio	Production Loss Ratio	Equipment Load Rate	*Mean Time to Failure*	*Mean Time to Repair*	*Corrective Maintenance Ratio*
Logistical (IoT available)	SQ														
	PSQ														
	GQ														
	RQ														
	PQ														
	CM														
	PL														
	EPC														
Maintenance times (Not available)	*TBF*														
	TTR														
	FE														
	CMT														
	PMT														

(Italic and underlined style contents indicate that IoT cannot be applicable)

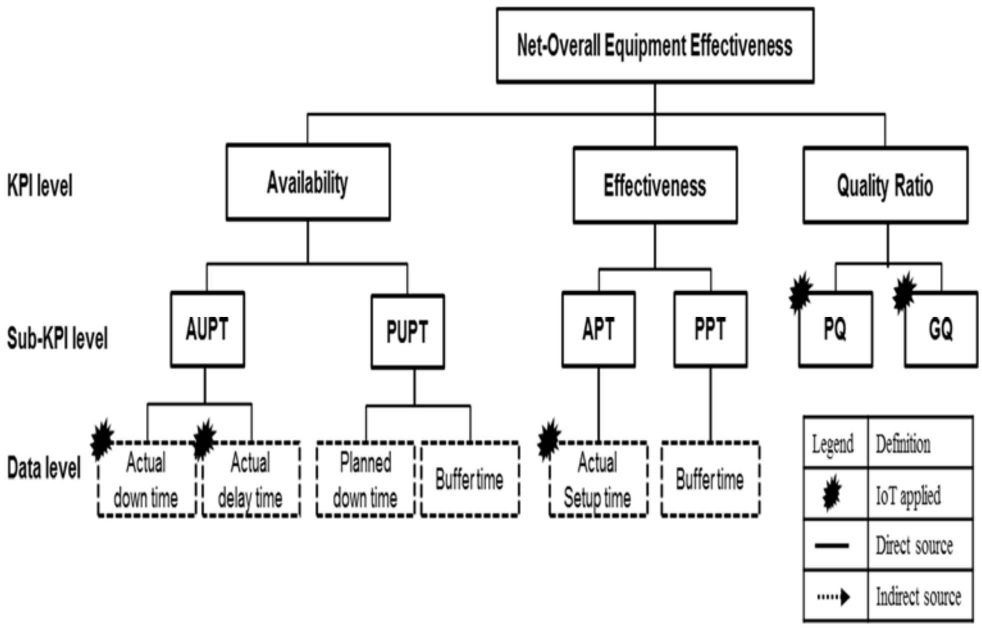
3.2.2. Selection of key performance indicator; Net Overall Equipment Effectiveness

As one of the 26 KPIs, four of these KPIs were selected for the present experimental simulation study, namely, 'Quality ratio', 'Availability', 'Effectiveness', and 'Net-Overall Equipment Effectiveness (Net-OEE)'. This was done due to the increasing popularity of the Net-OEE, and its wide application as a quantitative tool essential for measurement of productivity in manufacturing operations [65]. Analytically, Net-OEE is defined as a measure of total equipment performance, that is, ratio between what was actually manufactured and what could ideally be manufactured. One of the objectives of this study was to investigate how the Net-OEE derived from IoT data, and to present the response model if anomaly data are detected. Net-OEE has three KPIs for equipment performance based on the 'Availability', 'Effectiveness', and 'Quality ratio' of the output. To calculate these KPIs, it was necessary to first calculate the sub-KPIs, namely, Planned Busy Time (PBT), Actual Production Time (APT), Actual Unit Processing Time (AUPT), Produced Quantity (PQ), Good Quantity (GQ), and additional sub-KPIs which are not presented in ISO-22400: Planned Unit Processing Time (PUPT) and Planned Production Time (PPT).

As <Figure 3.9> presents, the Net-OEE calculation process can be applied within sub-KPIs and other measurement sources. Sub-KPIs have two perspectives on time – one is the standard time based on the operation calendar; and the other one is the real-time perspective, which is acquired by the IoT devices.



<Figure 3.10> Description of overall process for deriving Net-OEE



<Figure 3.11> Network of Net-OEE and IoT applicable parts

First, ‘Availability’ is a measure of how well the capacity of a production unit is used relative to the scheduled capacity. It presents the relationship between the planned target cycle and the actual cycle. A score of 100% means that the operation is running perfectly without any operation delay.

Availability = AUPT / PUPT (1)

AUPT = Actual Order Execution Time (AOET) – (Actual down time + Actual delay) (1.1)

PUPT = Planned Order Time (POT) – (Planned down time + Buffer Time) (1.2)

PUPT is the expected planned operation time without any down time or setup time losses. Planned Order Time is the time during which the machine is scheduled for production. It is necessary to allow for losses in the POT, because machine breakdown may occur during production and repair time would be required. AUPT is the actual time during which a work unit is executed, and it considers the fact that the actual production time may differ from that assumed during planning. Availability is calculated based on the time properties, which can be detected through an IoT device attached to the machine. In the event of the occurrence of actual down time, delay time, the IoT device measures the corresponding start and finish times.

Second, ‘Effectiveness’ is a measure of how effective a work unit is during the production time. It captures the deleterious effects due to breakdown, setup, and adjustment. A score of 100% implies that the actual process is running on schedule.

Effectiveness = PPT / APT (2)

APT = AUPT – Actual setup time (2.1)

PPT = PUPT – Setup time (2.2)

PQ = # of final products before the inspection process (2.3)

PPT is the planned run time for producing one work unit (WIP), also referred to as the actual cycle time. This sub-KPI represents the value-added processing time minus the minor or idling losses. PQ is the quantification of how well a final product complies with the production order.

Third, 'Quality ratio' is the ratio of the Good Quantity (GQ) to the Produced Quantity (PQ). It captures loss due to defects, rework, and yield. This sub-KPI is designed to exclude the effects of Availability and Effectiveness in presenting not-qualified process.

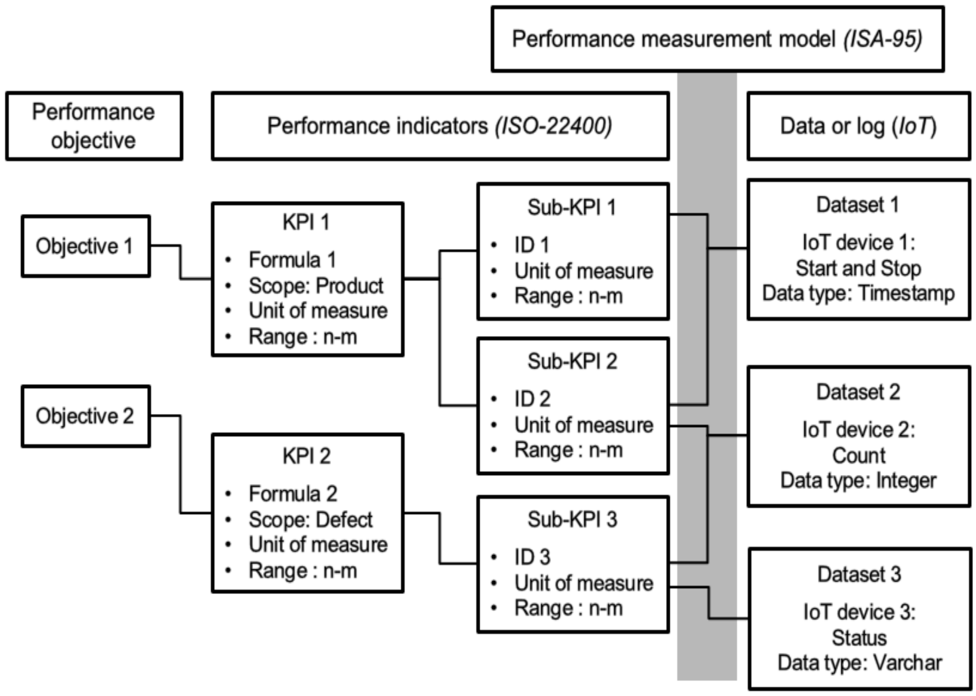
Quality ratio = GQ / PQ (3)

Finally, 'Net-Overall Equipment Effectiveness', which is the most important KPI, is a single indicator that integrates the availability of a work unit (Availability), the effectiveness of the work unit (Effectiveness), and the completeness of the produced goods (Quality ratio).

Net-OEE = Availability * Effectiveness * Quality ratio (4)

3.3. Suggestion of smart factory production performance model

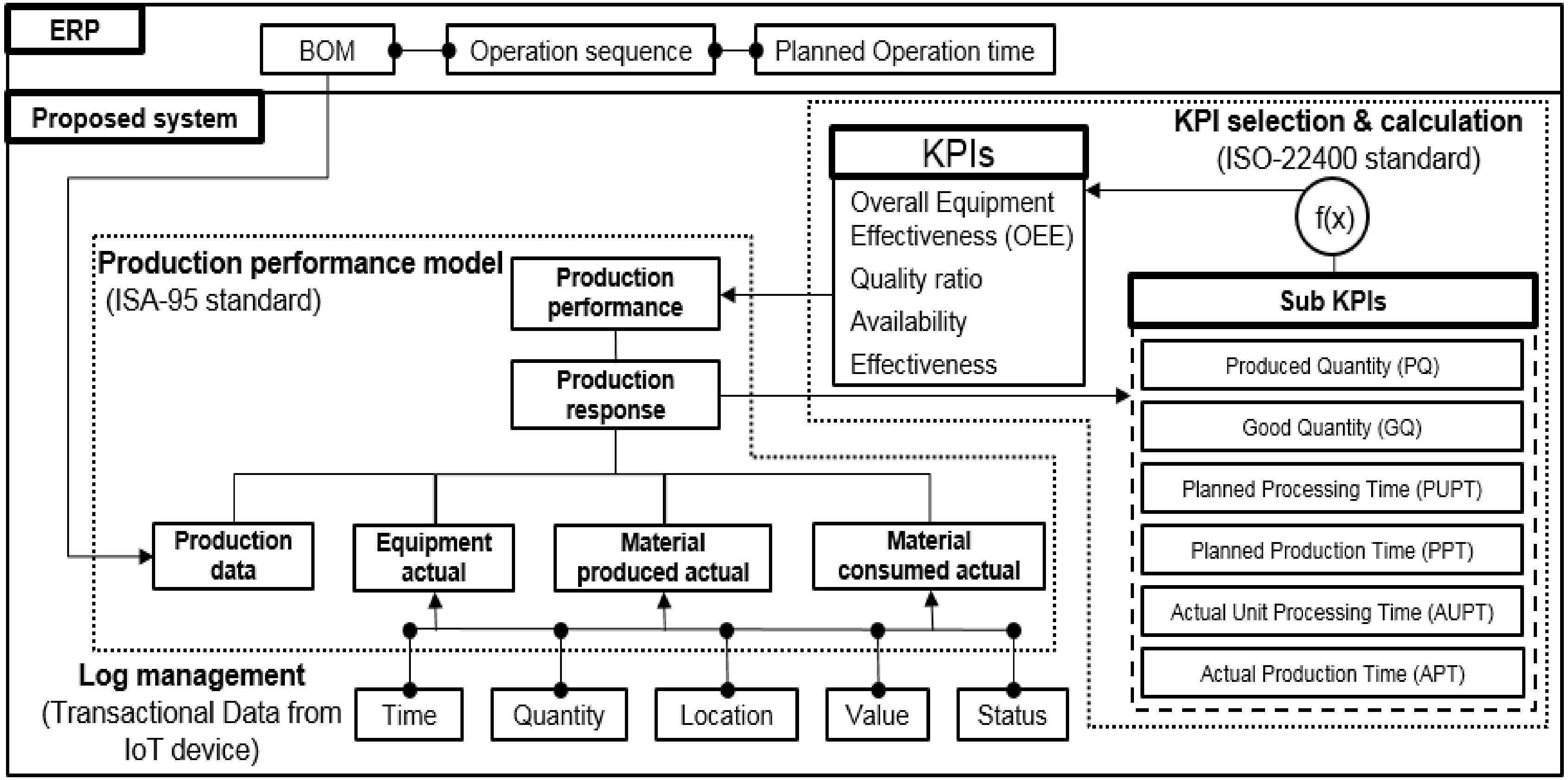
Data, information, knowledge, and wisdom hierarchy linking was specified by Ackoff [66] in 1988 and this linking has been regarded as a standard procedure of data analysis for deriving a valuable outputs. The performance measurement process can be also applied to this concept in that data which represents the properties of objects and values can be deducted as IoT data or log files and information what is the presentation of description of data can be same with performance indicator and finally knowledge, which is the collection of information, is a performance objective that is used to aid decision making.



<Figure 3.12> Combination of international standard and performance measurement

As a previous phrase denoted, performance indicator from ISO-22400 implies the status of production lines so that performance measurement model in ISA-95 is designed to aggregate the real-time IoT data. One of the core contributions of this work is the presentation of multidisciplinary information model for deriving Net-OEE. This study conjoined ISO-22400 for KPIs calculation with a production performance model in ISA-95 considering IoT data characteristics. In addition, to extract production plan for comparing plans with actual, BOM, operation sequence, planned operation time from ERP are considered. <Figure 3.13> suggests the smart factory production performance model in accordance with ISO-22400 and ISA-95.

When IoT device detects a product on the production line, its data is sent to the MES. Considering the IoT functionality, transactional data that include information about the 'time', 'quantity', 'location', 'value', and 'status' can be acquired from the product. The acquired data is then aligned with the production performance model, which consists of three sub-parts, namely, the equipment actual, material produced actual, and material consumed actual. Not all KPIs can be evaluated directly using the IoT data; other sources are often required for additional data such as production data that is aligned with the Bill of Material (BOM), operation sequence, and planned operation time. Production data are obtained from the upper system and includes schedule information. Using planned and actual data, the above mentioned sub-parts classify the IoT data based on their characteristics to provide upper level resource for calculating the sub-KPIs. The production response then uses the sub-KPIs to calculate the final KPIs.



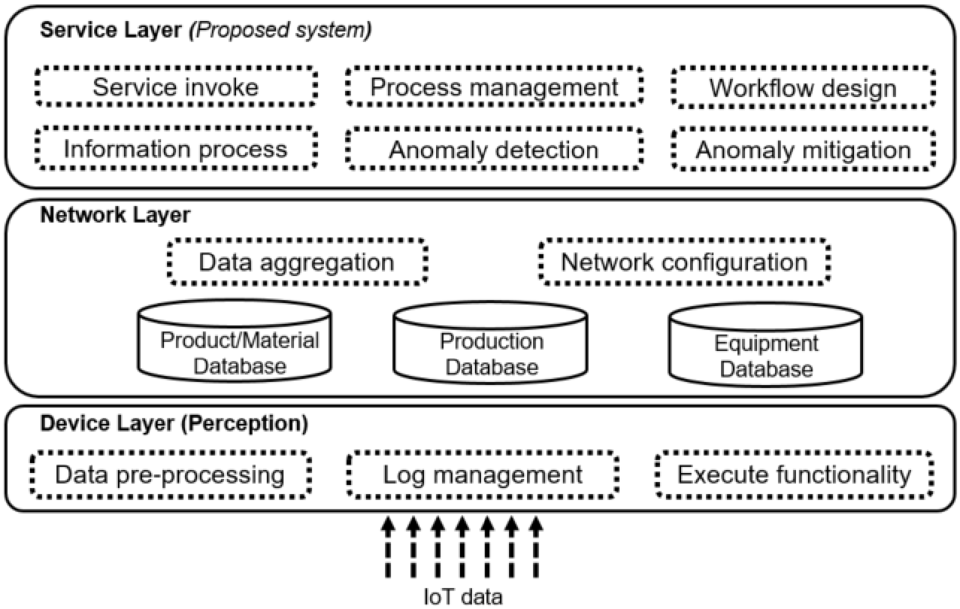
<Figure 3.13> Multidisciplinary information conjoined with smart factory production performance model

Chapter 4. Implementation of smart factory performance measurement system

4.1. Configuration of smart factory architecture for performance measurement

4.1.1. Development of network architecture

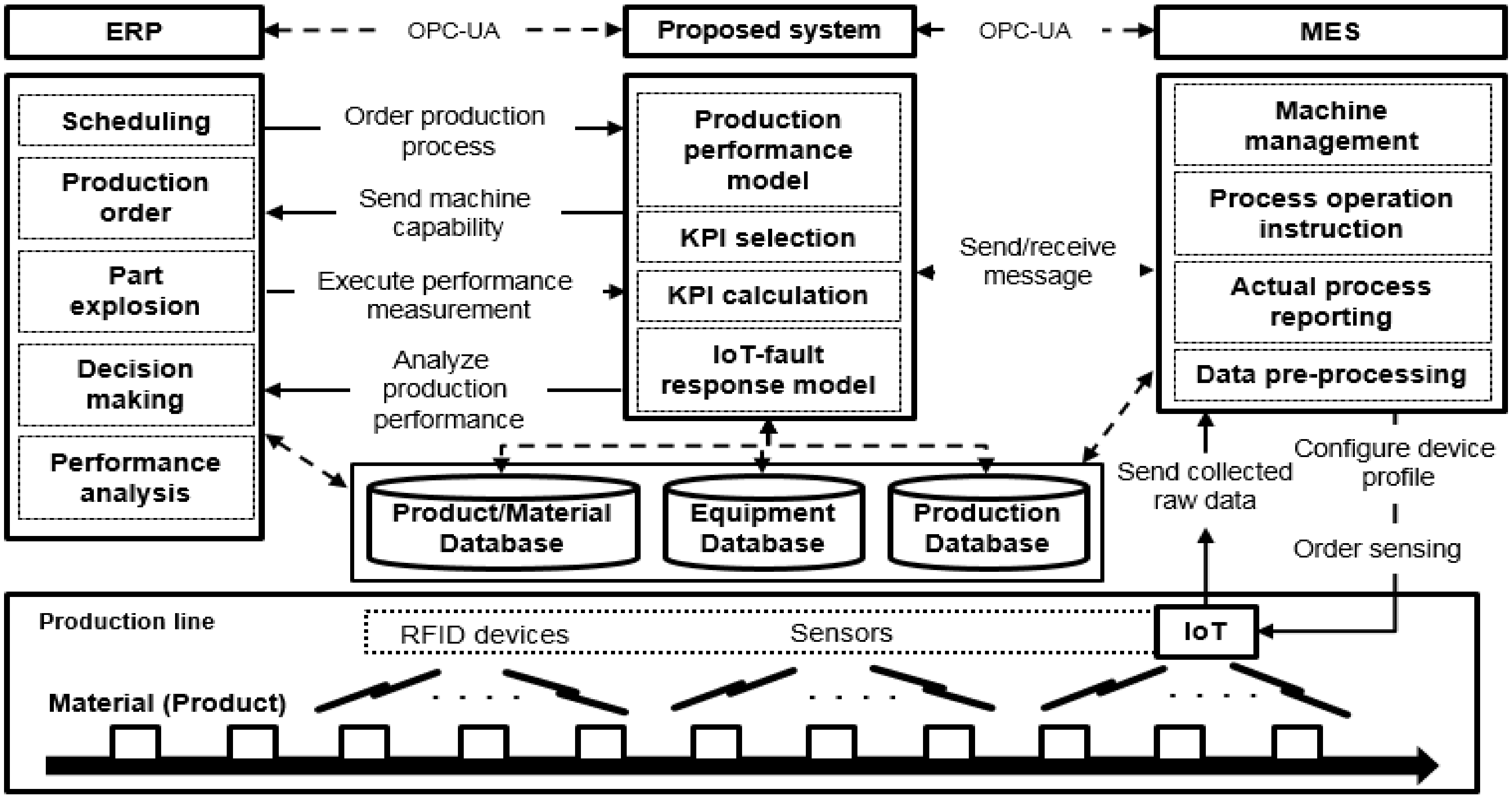
Most studies that propose the architecture of the Internet of Things typically divided layers into three levels [67-70]. This dissertation also configures smart factory architecture as three layers and following <Figure 4.1> shows the suggested architecture. Though other studies extends layers levels more detail for describing their developed models [71-73], dividing three layers are most widely accepted to specify the IoT network architecture.



<Figure 4.1> Smart factory performance measurement system architecture

Device layer (also called perception) is made up of the IoT devices and a data aggregation set. This layer is responsible for acquiring, collecting, and processing the data from a physical object, which is situated on production lines. IoT devices are distributed over a large area and detect physical objects to transfer real things into digital data the upper level network layer supports massive volumes of IoT data. This network layer includes server and gateway devices and is in charge of data aggregation, configuration management, data processing, and data transfer protocols. In addition, the network layer is divided into three types, according to data usage and data originality, as denoted in <Table 4.1>. The product and material database presents product information such as Bill of Material and Order information, etc. As production plan decisions will be based on this database, standard KPIs are used. The production database results from actual production, such as the amount produced; it primarily stores the sub-KPIs of the Net-OEE. Furthermore, in the case of data mitigation, this dissertation added an additional dataset for distinguishing the anomaly data and mitigated. The Equipment Database describes the IoT device property and machine status. Finally, top level service layer includes a manufacturing information system that has algorithms for information analytics, security control, process modelling, and device management.

Generally, as materials flow along the shop floor, IoT detects their status and sends information to the database to afford a better understanding of what is actually happening on the shop floor.



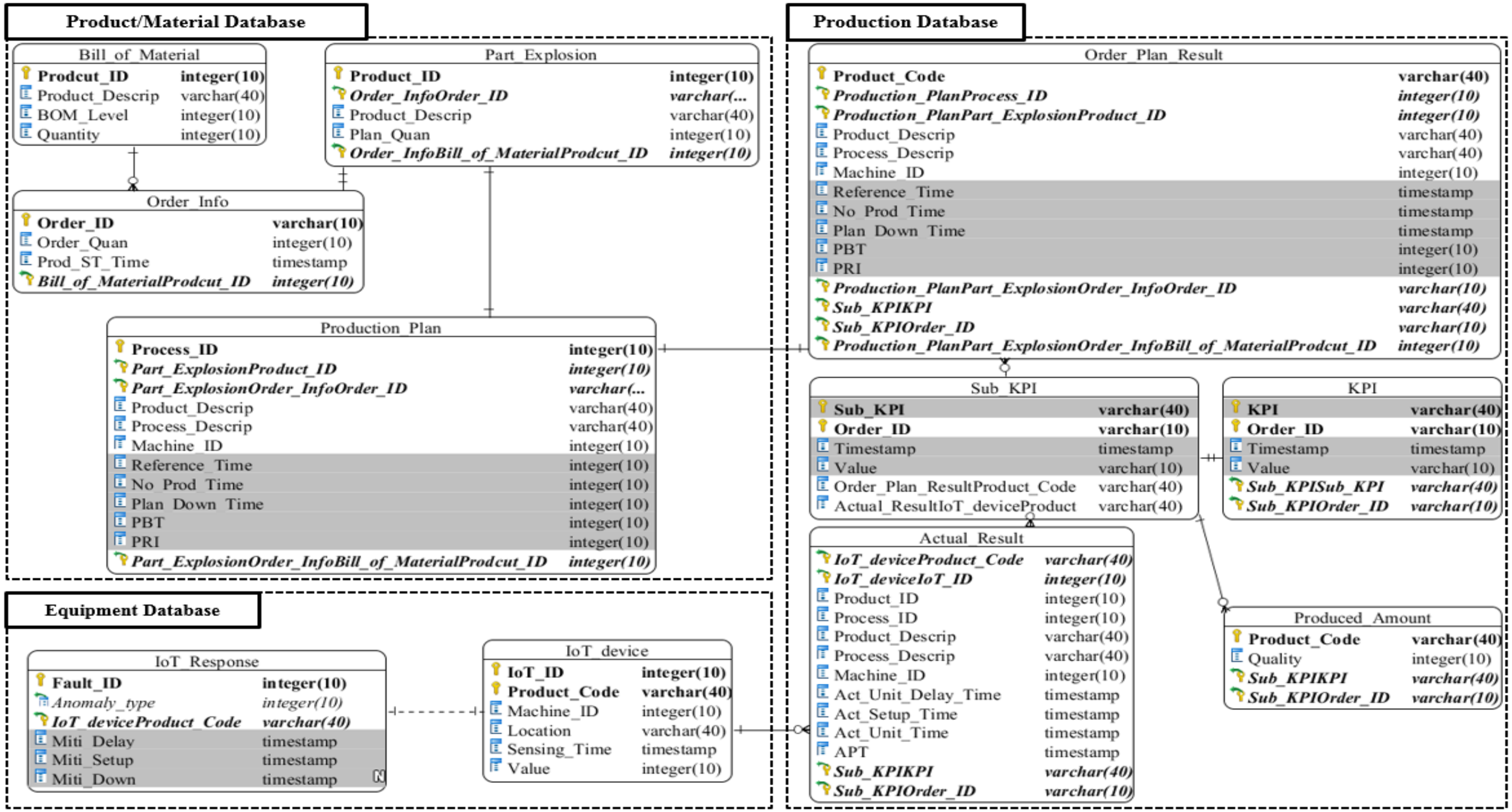
<Figure 4.2> Data flow of proposed architecture

- The data, acquired by the IoT devices, are gathered into the production database.
- Based on the collected data, the KPI calculation is executed in a proposed engine where production performance model is used to conduct performance evaluation
- The outputs of the business logic (i.e., the KPIs) are used for performance analysis.
- In this dissertation, to solve communication issues among existing information systems, uses OPC-UA (Open Platform Communication Architecture) standard.
- If some IoT data faults happen, IoT data anomaly response model will be executed for mitigating the effects of faults.

In this section, a physical Entity Relation Diagram (ERD) is used to present the minimum architectural components required for performance measurement. The diagram enables the linking of the data flow and database schema level with the process configuration to form a complete performance measurement system. To derive our target KPI (Net-OEE), it was necessary for the IoT device to detect the location, sensing time, and ID-value. The location data identifies the place of the material processing. The sensing time identifies the actual down time, actual delay time, and actual setup time (see, equation 1.2), as well as sub-KPI (APT). Finally, the value is used to check the number of produced materials and to determine two the sub-KPIs (PQ and GQ). Additional information of dataset composition is presented at the <Table 4.1>. The noticeable dataset of composition is the Equipment DB’s relational table, namely, IoT_Response. Other relational tables are used for configuration of the processes of Net-

OEE derivation using planned and actual production data. However, IoT_Response relational table includes the data, if some IoT devices may exhibit some IoT data anomalies, in which case the performance measurement would be incorrect or unusable for calculating KPIs. The details of data description and its related model is described in Chapter 5. Based on the composition and functions of the IoT devices and IoT data anomaly response model, the ERD was developed as shown in <Figure 4.3>. Following figure highlights the certain attributes which are remarkable points of this dissertation.

The cardinality ratios are determined from the relationships between the entities; this paper considers 1:1, 1:N, N:1, and M:N cardinality. For example, one IoT device is required to be attached to one machine to detect the machine status. Therefore, these two tables have a 1:1 relationship. In addition, as all relationships must share their data, all relationships are shown to be connected by a solid line, which represents a mandatory relationship. <Table 4.1> describes the functions of each relationship and <Table 4.2> shows the description of datasets.



<Figure 4.3> ERD for proposed performance measurement system

<Table 4.1> ERD description based on information level

Architecture	Relation	Description
Product/ Material DB (ERP)	Bill_of_ Material	Contains a list of the products, sub-assemblies and the quantity of each material required to produce the final products.
	Production_ Plan	Includes the standard time required to produce a product based on ISO-22400
	Order_Info	Presents the basic order information, including the quantity of the final product and the production start time.
	Part_ Explosion	Calculates required material quantities based on the BOM and order quantity.
Production DB (MES)	Order_Plan_ Result	Generates the time-phased production plan based on the standard time required to produce each product.
	KPI, Sub_KPI	Contains the performance measurements' for the semi and final outputs.
	Produced_ Amount	Stores the total produced amount of each final product and the amount that satisfies the quality requirements.
	Actual_Result	Is derived using the timestamp data contained in the IoT data.
Equipment DB (MES/IoT)	IoT_ Response	Provides the information required to store additional fault response, if the IoT experiences failure during its operation.
	IoT_Device	Describes the IoT properties and the product target code to enable the storing of each actual result.

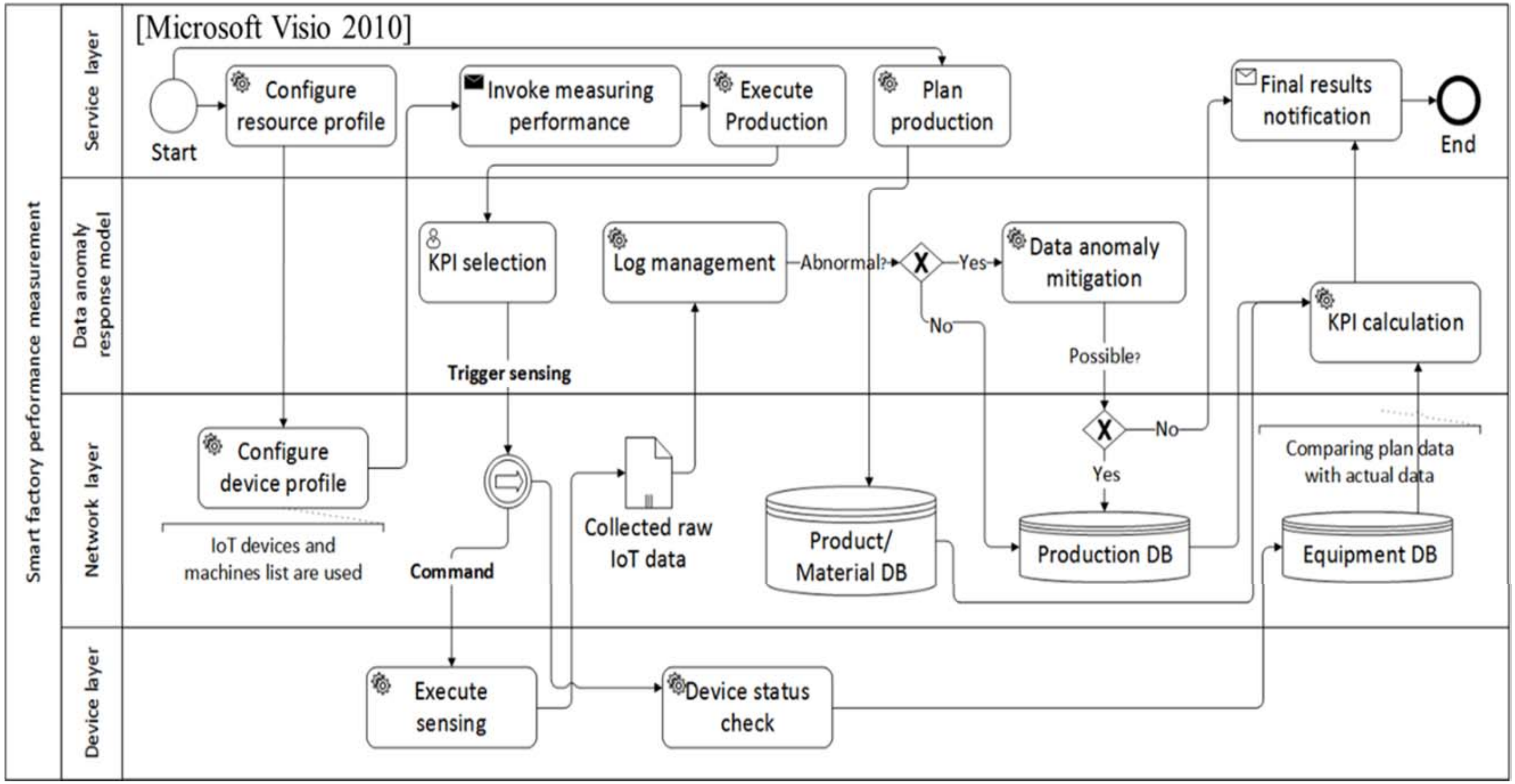
<Table 4.2> Dataset description

Relation	Attribute	Data type	Related KPI	IoT
Common	Product_ID	Varchar	-	-
Bill_of_ Material	BOM_Level	Number	-	-
Order_Info	Order_Quantity	Number	Production Quantity	○
Production_ Plan	Reference_time	Number	Reference time	-
	No_Prod_time	Number	No production time	-
	Plan_down_time	Number	Planned down time	-
Common	Production_Code	Number	-	-
	Machine_ID	Varchar	-	-
	Process_ID	Number	-	-
	Product_ID	Number	-	-
Actual_ Result	Act_Delay	Timestamp	Actual delay time	○
	Act_Setup	Timestamp	Actual setup time	○
	Act_Down	Timestamp	Actual down time	○
IoT_ Response	Anomaly_Type	String	-	-
	Miti_Delay	Timestamp	Actual delay time	○
	Miti_Setup	Timestamp	Actual setup time	○
	Miti_Down	Timestamp	Actual down time	○
IoT_Device	IoT_ID	Varchar	-	-
	Sensing_Time	Timestamp	Actual KPI	○
	Value	Varchar	Good Quantity	○
	Status	Binary	Actual down time	○

4.1.2. Designation of business logic with BPMN

A business process is an activity or set of interrelated activities that are performed by employees or departments working together to accomplish an organizational goal. There are several standards and tools for the development of a business process model, such as Petri nets, Integration DEFinition Methods (IDEF), and Business Process Modelling Notations (BPMN). This study used BPMN which is a graphical notation to facilitate comprehension of the work flow and database view. Several studies also work on BPMN include a process instance notation which was developed, and based on BPMN, to support the visualization of process execution rather than process definition [35, 74].

With the integration of the Net-OEE calculation process with the suggested system architecture, our system business logic used BPMN for addressing how a smart factory architecture and MES standard could be applied to a performance measurement process and defined all the possible paths and actions that need to be taken. Using the multidisciplinary overview information in <Figure 3.12> as a functional template and the smart factory system architecture (ERD) in <Figure 4.3> as a specified data template, a BPMN with a smart factory performance measurement system was developed. The following figure illustrates the measurement workflow and system hierarchy.



<Figure 4.4> Abstract BPMN with smart factory performance measuring process

This model shows how the new IoT concept could be applied into the performance measurement system (to-be model) and focuses on intra-organizational perspective and internal communication activities within one pool (orchestration). With orchestration, this model is driven by the sequence or flow of activities as part of work to be carried out performance measuring. All the lanes are in accordance with the architecture as indicated in <Table 4.1>, except the data anomaly response model, which represents the production performance model and response model. The bottom lane, which is the device layer, represents the IoT architecture for executing sensing, and also checks the device status determine whether IoT devices working well or not. The network layer represents the MES architecture for configuring the machines and attached IoT devices. In addition, this lane's activities are used to collect raw data and derive the sub-KPIs for considering the actual production results. The service layer lane is the ERP architecture and manages the production plan and actual production results. By comparing the planned and actual processes, the performance can be analyzed to determine the KPI. While event-based performance model is executed when the KPIs are required to be checked or managers trigger the KPI analysis, our proposed IoT-based real-time performance model derives KPIs close to the real-time updates. The procedure of the present BPM is as follows:

(1) When a stakeholder performs a measurement, the KPIs and sub-KPIs are selected and their formulas are obtained from ISO-22400. The production BOM is also identified to compare the planned performance with the actual performance (Product/Material DB).

(2) After the configuration of the machine profile and IoT device profile, the sensing process is executed to decide the amount of production performance.

(3) The IoT device status is checked to determine whether the performance measuring is operable or not. If there is a fault in the operation of the device, the IoT fault data would be transmitted to the upper level and its related data are stored into the equipment DB.

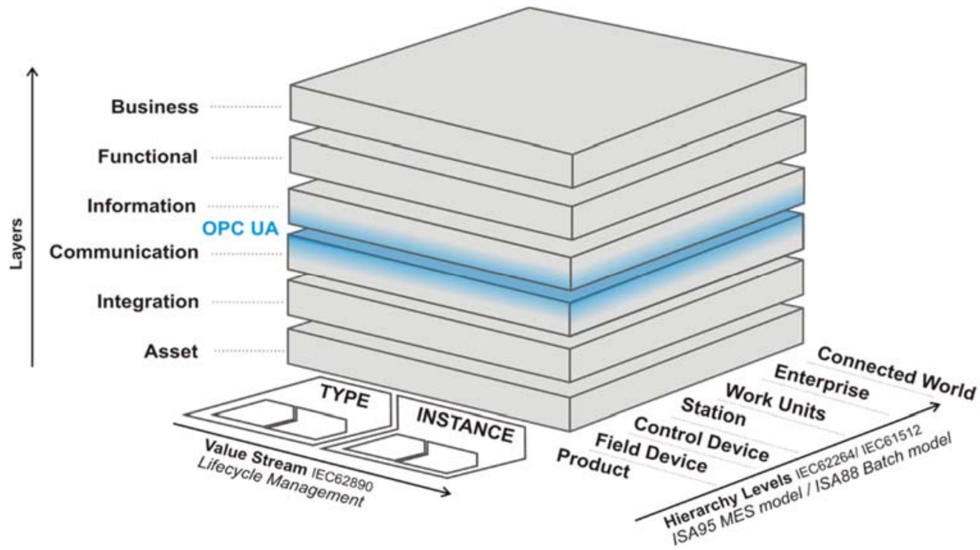
(4-1) If the operation has normal condition, the production would be executed and the IoT device would sense the start/finish times of each standard time and send the raw IoT data to the network level (Production DB).

(4-2) From the sensing devices, log files are collected and the IoT response model makes a decision whether data anomaly is identified or not. In the case of error-free data, log files are stored at the Production DB. However, if data anomaly is detected, mitigation is executed to replace anomaly data with inferenced data.

(5) Based on the sub-KPIs, which are calculated using data obtained from the previous process, the performance analysis is used to determine the KPIs to complete the performance measurement process.

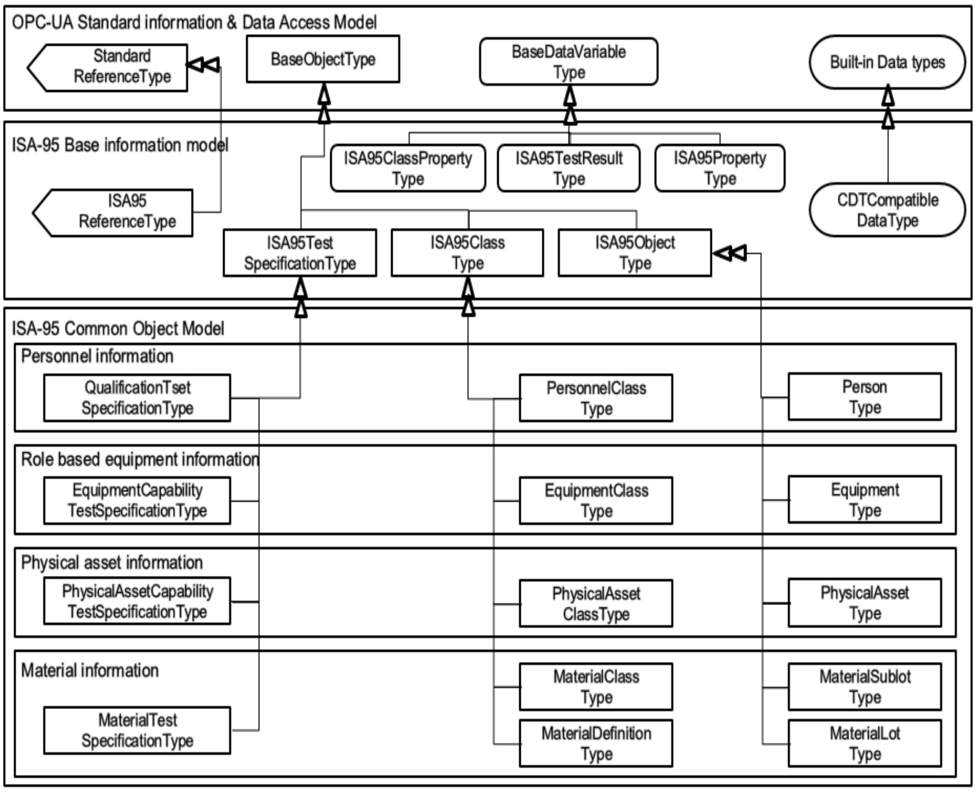
4.2. Adaptation of OPC-UA

Open Platform Communications-Unified Architecture (OPC-UA) is a standardized communication protocol and provides a modelling interface for integrating the whole manufacturing information system without any interoperability problem [75]. It is a communication framework for any types of information in the industrial environment, supporting custom complex data structures. OPC-UA is fundamentally about data modelling and uses object-oriented techniques, including type hierarchies and inheritance, to model information [76]. OPC-UA is envisioned as an enabler for a seamless vertical integration in automation systems and does allow any combination of client/server component at various levels of the automation hierarchy for a specific application [77]. As following figure denotes, RAMI 4.0 accepted OPC-UA for presenting interoperability.



<Figure 4.5> Relationship between OPC-UA and RAMI 4.0 (Source: https://www.bitctrl.de/de/industrieautomation/opc-ua.html)

ISA-95 standard specifies the standard interface among the manufacturing information system using B2MML (Business To Manufacturing Markup Language). However, as the OPC-UA has been widely used in smart factory environment, the OPC-UA implementation will complement the existing B2MML implementations by providing a secure and reliable environment that is widely accepted in the manufacturing industries [78]. OPCfoundation [78] presents an extension of the overall OPC UA standards and defines an information model that conforms to the ISA-95. This extension version describes the object model for Personnel information, Role base equipment information, Physical asset information, and Material information. As <Figure 4.6> denoted, this version tries to integrate ISA-95 with OPC-UA.

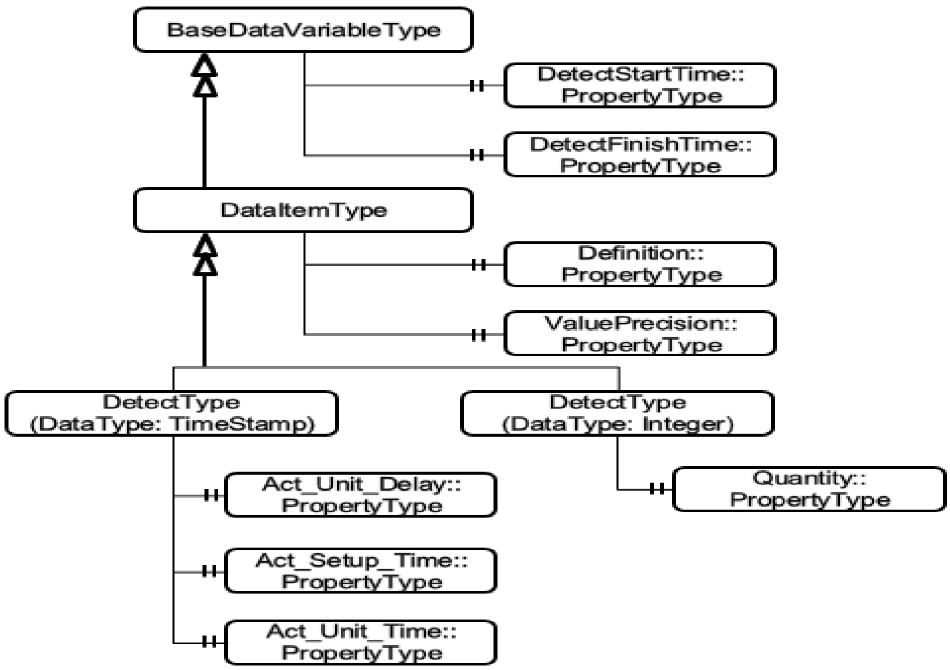


<Figure 4.6> ISA-95 OPC-UA information model [78]

The OPC data access, which is the first successful classic OPC standard, was initially designed as an interface to communication drivers [79]. The new OPC-UA is introduced as a real replacement for the existing Component Object Model (COM)-based specification, but maintains a classic OPC features. Applying OPC-UA allows customizing how data is organized by showing the systemized architecture. All the components of OPC-UA are designed to be a scalable including information model, communication model, and security method. One of the fundamental components of the OPC-UA is the information model, because it serves understanding of semantic of the provided data which are rudimentary information such as unit of measure, formulation, etc. An information model provides a set of standardized node, reference types and instances, which define the way how the server address space has to be constructed in a structured manner [80].

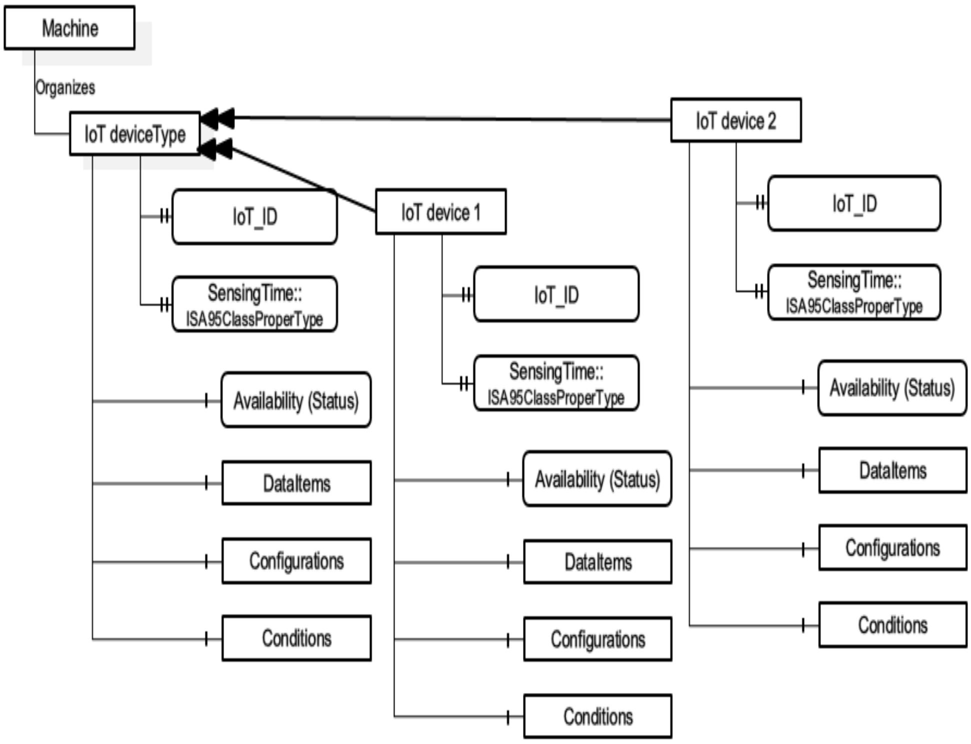
To derive general concepts and recommendations on how to implement information models in automation, Mahnke, et al. [81] discussed several relevant international standards that provide information models and investigated which standards are compatible for OPC-UA. They denoted that ISA-95 is enough to be exchangeable with OPC-UA in that performance information model in ISA-95 includes central means of modeling such as aggregation, association, and inheritance. However, Dennis Brandl, et al. [82] pointed out ISA-95 limitations in that ISA-95 information exchange model is mainly for periodic MES and ERP level, not for sub-low level such as PLC and HMI based data exchange. Through the implementation of information model conjoining ISA-95 and OPC-UA, this study is able to specify the use-case scenario, system hierarchy, and data type description.

Though OPC-UA foundation and ISA-95 association co-worked for developing integrated standard based information model, a number of companion specification models have been proposed such as data access information model, etc. For aggregating several system components, this study proposes three different models: data access information model, device information model, and integrated model. The outputs of OPC-UA application can exhibit several outputs within process views and systematic views.



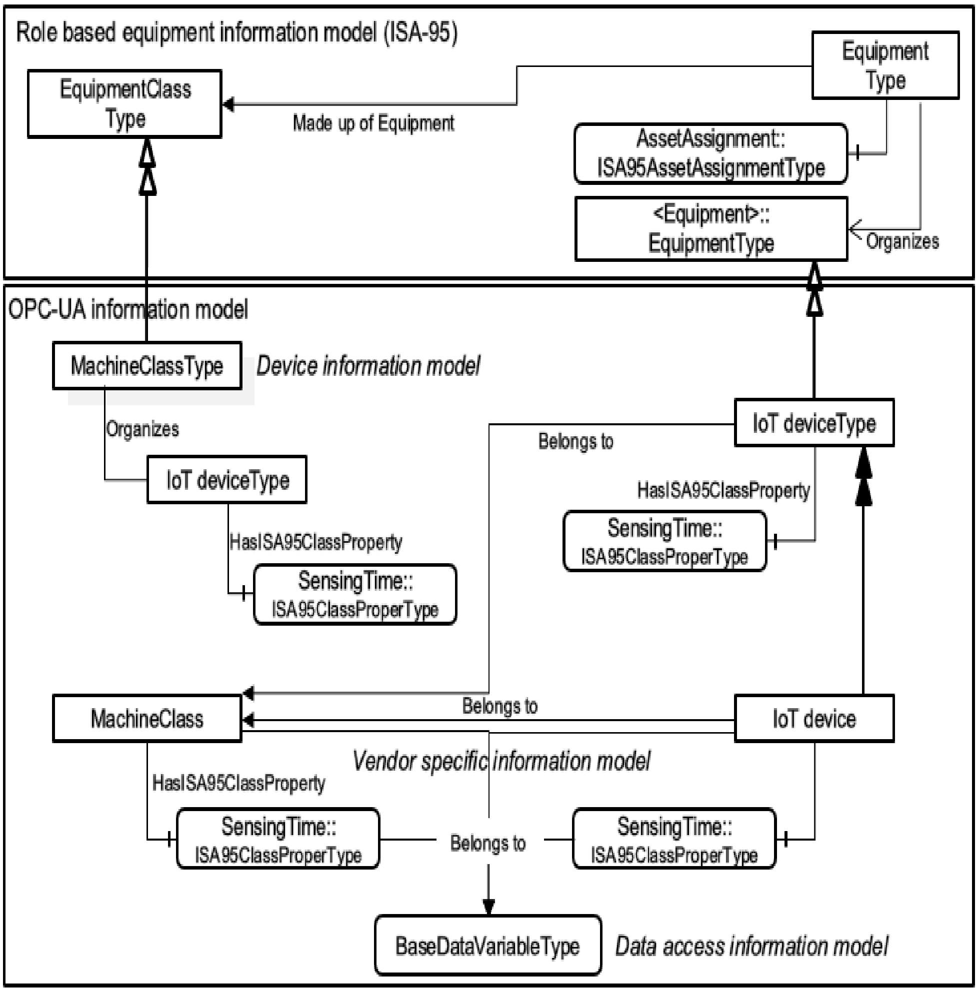
<Figure 4.7> Data access information model

First, data access information model presents an additional definition of variable types and a complementary description of data objects. BaseDataVariableType variables have two property types which are fundamental functions of IoT device: DetectStartTime and DetectFinishTime. In the production lines, IoT devices are situated in the front and backside of machines for detecting production start and finish time. To describe more detail detect types, DataItemType has two data types: Time stamp and Integer. As section 3.2.2 describes Net-OEE calculation components, Actual unit delay time, Actual setup time, and actual unit time should be considered for deriving the final Net-OEE values. Those sub-KPI should have a time stamp value and for deriving quality ration sub-KPI, production quantity should be considered so that integer data types is also be defined.



<Figure 4.8> Device information model

Second, device information model describes the relationship among the network components. In the scope of the information model, a device is defined as “an entity that provides sensing, actuating, communication, and/or control functionality [83]. IoT deviceType defined any instance of IoT devices so that each IoT devices can be modelled as a subtype of IoT device Type and six of these subtypes are introduced in the specification. IoT_ID is the unique string data type and SensingTime is a variable that comprised of three detect types (Actual delay time, Actual setup time, and Actual unit time). Availability is the binary data types that show whether IoT device is working well or not. DataItems has a string value which includes additional information such as device history. Configuration shows the relationship among the network components.



<Figure 4.9> Integrated information model

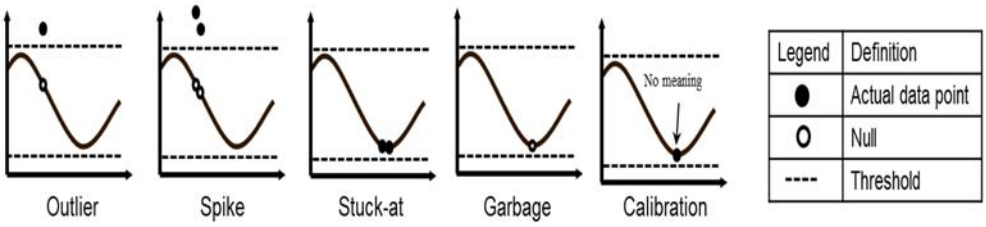
Finally, the role based equipment model contains information about specific equipment, equipment classes, and maintenance associated with equipment in accordance with ISA-95 performance measurement model. This model is extended with a data access model, a generic information model, and vendor specific model addressing specific abilities. The suggested performance measurement system is strongly associated with IoT devices so that among the ISA-95 class properties (Personal, Equipment, Material, Material consumed), this study chooses equipment classes for identifying the IoT device data operability. <Figure 4.9> shows summarized three proposed information model (Device information model, Data access model, and Role based information model). Suggested three kinds of information model have a relationship with each other and fundamental classes are made up of ISA-95 properties. An integrated model includes typical extensions that a company may generate to adapt the generic IoT-based ISA-95 information model for a deployment specific information model.

Chapter 5. Development of the IoT data anomaly detection and mitigation algorithm

5.1. Classification of the data anomaly types

In a normal situation, connected IoT devices transmits the correct real data, and updates a centralized data repository with the status of the device for performance analysis. However, there is a possibility of IoT fault occurrence due to hardware or software malfunction. A major malfunction is from the lack of robustness in IoT devices, which do not provide stable measurement accuracy. This malfunction interferes with performance measurement so that manual intervention and adjustment are therefore required to describe the real situation. To reduce the possibility of performance measurement malfunction, fault detection should be executed. Complete data anomaly classification is an essential prerequisite for identifying the IoT faults and improving the data anomaly detection ability.

Classifying data anomaly cases is a difficult and complex task, due to many factors that influence IoT-data and could cause data anomaly. An IoT fault problem results from breakdown and deterioration of IoT devices and network components beyond the decided threshold and brings about loss of reliability of manufacturing performance. This classification of IoT faults serves as a basis of their possible causes, and could be utilized in several ways to design the data anomaly detection algorithm. Ni, et al. [84] presents the five possible sensor network data fault types. Five fault types are referred to in this study (outlier, spike, stuck-at, garbage and calibration) as a root causes of IoT failure. This research uses a statistical method, which includes the mean, standard deviation, and correlation-coefficient parameter, for detecting the outlier and spike while stuck-at, garbage, and calibration using data type identification are able to be detected. Following <Figure 5.1> describes the detail information of data anomaly types.



<Figure 5.1> Data anomaly fault types description.

1) An outlier represents a data point that is very different from others and is out of range. The outlier is situated in an unexpected data position, which is also far from other data clusters. This error type can be identified using the k-means clustering algorithm. Outlier is main target of detecting abnormal, because it is strongly related to the spike fault type. Zhang, et al. [85] defined outlier as “those measurements that significantly deviate from the normal pattern of sensed data” on the point of sensor network view.
2) A spike is a multi-data point with a much greater than expected rate of change which may or may not return to the normal value. A spike is different from an outlier, which is in the distribution of levels of time series. If the standard deviation is large enough, data is regarded as spike. In this thesis, if previous IoT data defined as outlier and following IoT data is also be labeled as outlier, those dataset anomaly type is modified as a spike.

3) Stuck-at means that the variance of the acquired IoT data is close to zero for a long period, also called constant. IoT data may or may not return to normal operating behavior after the fault. The acquired IoT value is either very high or very low compared to the normal situation. If the standard deviation is equal to zero and the correlation-coefficient value is bigger or smaller than zero, data are considered as Stuck-at. When the signal from a certain communication medium is constant, Stuck-at fault occurs [86].
4) Garbage means the data are contaminated by garbage data such as the null data type. Garbage data do not follow the overall trends, which has no information.
5) Calibration data represents un-preprocessed data that still provide meaningful information. This fault usually occurs at resource failure which has a data preprocessing process. Garbage and calibration can be identified through data type detection.

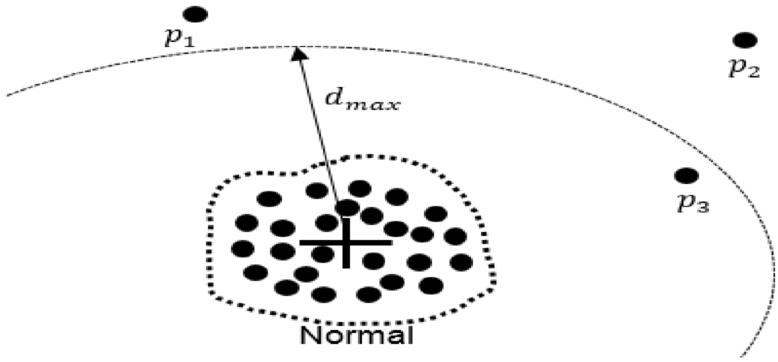
5.2. Designation of the data anomaly response model

5.2.1. Data anomaly detection algorithm

Due to the dynamic variation of the condition of the IoT network, data monitoring is important for checking reliability. Lekhi and Mahajan [87] present the categorization of the anomaly detection approaches as follows: Distribution based, Distance based, Depth based, Cluster based, Control chart technique, and Outlier detection integrating semantic knowledge. In this thesis, cluster based approach is mainly applied to detect the outlier and spike anomaly types based on the type of data anomaly classification already presented at <Figure 5.1>. k-means algorithm is a popular method of the unsupervised learning algorithms that are used to solve the well-known clustering problem. It is an iterative clustering algorithm widely used in data mining for finding statistical structures in data [88]. The k of the k-means clustering method is a positive integer number indicating the number of groups which is established by a priory by experts or learning dataset. To calculate the degree of homogeneity and heterogeneity, the k-means clustering method employs the Euclidean distance as a measure of the similarity between observations and groups [89]. The following equation describes the distance measurement formula.

$$\mathrm{D} = \sum_{i=1}^{k} \sum_{x_j \in s_i} | x_j - \mu_i |^2 \qquad (5)$$

where k clusters s_i, i = 1, 2, ..., k ; μ_i means centroid of all the points $x_j \in s_i$



<Figure 5.2> Outlier detection using k-means algorithm

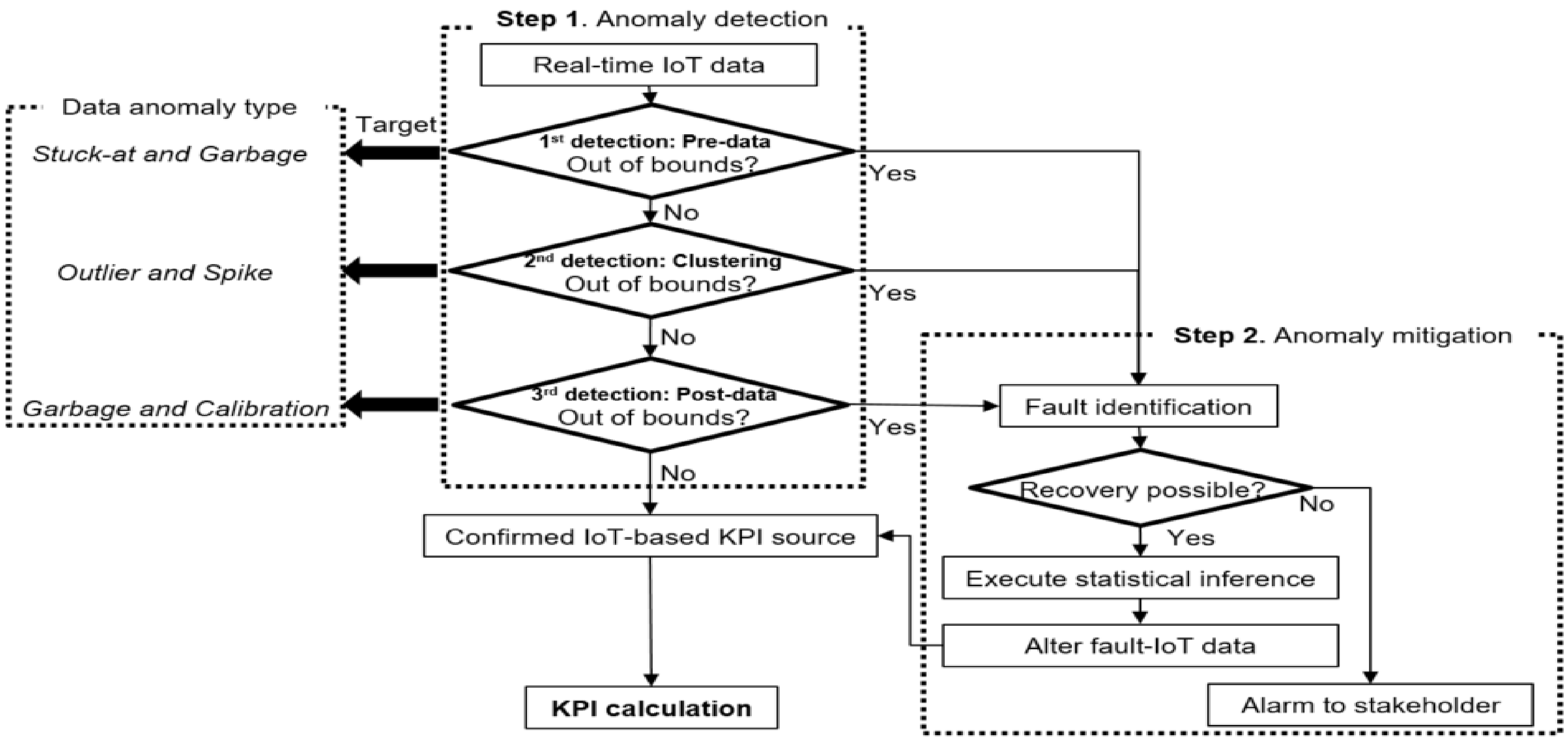
As <Figure 5.2> indicated, predefined threshold called d_{max} is important to define the data points labeling. The target data p_3 is classified as normal if it is situated in the d_{max} radius. However, p_1 and p_2 are labeled as abnormal data by the same token. In contrast to the classification method, outlier detection does not make use of the anomaly cluster, because it is same with binary discrimination [90]. In this reason, the k-means algorithm itself is extremely sensitive to outliers, and such outliers may have a disproportionate impact on the final cluster configuration [91]. The processing steps of k-means algorithm can be summarized as follows: first, Training data set containing normal and abnormal IoT data are configured into specified datasets. Second, the datasets are categorized into different clusters for normal and abnormal cluster using Euclidean distance method. Finally, the final cluster centroids are deployed for fast detection of outlier and individual target data is labeled normal or abnormal data considering whether the data is situated in normal cluster or abnormal cluster. Following table describes the summarized pseudo code of the k-means algorithm.

<Table 5.1> Pseudo code of the *k*-means algorithm

k-means Algorithm

```
Input: Data set with x variables and n observation denotes
        X={x1,…,xn}
Choose randomly selected x data points as the initial center
Repeat
    For each data point x ∈ D do
    Compute the distance between x and each center
    Assign x to the cluster with the nearest center
  end for
Re-compute the distance between x and all the center points
Until the convergence condition is satisfied
Compute distance of each centroids in terms of average of
  planned data
Set normal cluster which is the closest to the average of plan
Output: A set of outlier observation
```

Besides outlier and spike, other anomaly types (Stuck-at, garbage, and calibration) should be identified, because data loss or the data anomaly problem often occur during data transmission between the network and device layers. In this reason, this thesis proposes two step data response model considering the data storage types. Stuck-at and garbage is categorized as pre-data type, because these data anomaly types can be examined when data is defined as log file. After data preprocessing executed, outlier, spike, calibration, and garbage which also has the possibility of getting abnormal data in the middle of data preprocessing should be investigated. When the data anomaly is identified, our model decides whether the fault is fixable or not. If recovery cannot be executed, a notification is issued to prompt the stakeholder to take an appropriate recovery decision. However, if recovery is operable, statistical inference is automatically applied to mitigate the anomaly data. Following figure shows the overall process of the data anomaly response model.



<Figure 5.3> Overall process of the data anomaly response model

The core idea of anomaly detection is an identification of the gap between planned production time and actual production time acquired by IoT data. In addition, detection also compares acquired IoT data with previous acquired IoT data. First, following acquisition IoT data, pre-data analysis is undertaken to find stuck-at and garbage anomaly types. These anomaly types should be examined, since the related faults are log based and easily detected by examination of the aggregation data set pattern (Stuck-at) and data value type (Garbage). To detect the stuck-at anomaly type, our comparison time window is around 10 prior IoT data. If standard deviation of 10 prior data and the acquired IoT data is equal to zero and correlation-coefficient is not zero, then detection considers the acquired IoT data as the stuck-at anomaly type. The next anomaly type, Garbage, can be detected easily when the IoT data has a null data type.

$$Correlation - Coefficient\ (CC) = \frac{E(x_t - y_t) - E(x_t) \cdot E(y_t)}{S(x_t) \cdot S(y_t)} \qquad (6)$$

where E(*x*); the mean of *x*: standard deviation of *x*; CC=0:

x and y are not correlated

Second, the data points that are distinctively separate from the rest of the data are defined as outlier and spike. If the anomaly data deviate the mean of a normal data cluster and exceed the threshold of the deviation, then it can be regarded as a data anomaly. The main idea of the outlier and spike detection algorithm is to partition the dataset into clusters using *k*-means algorithm. It is based on the Euclidean distance between two data points and follows <Table 5.1> pseudo code.

Finally, after data preprocessing, IoT data are stored in the production database. The remaining calculation, which is already proposed in <Figure 4.4>, is used to measure the final Net-OEE. While the possibility is low, the calculation of numerous equations during this process may lead to make errors. For this reason, KPI or sub-KPIs could have null type (garbage) or unprocessed data type (calibration). These anomaly types can be easily detected through data value inspection using database query language. Following table describes the overall process of the data anomaly detection algorithm.

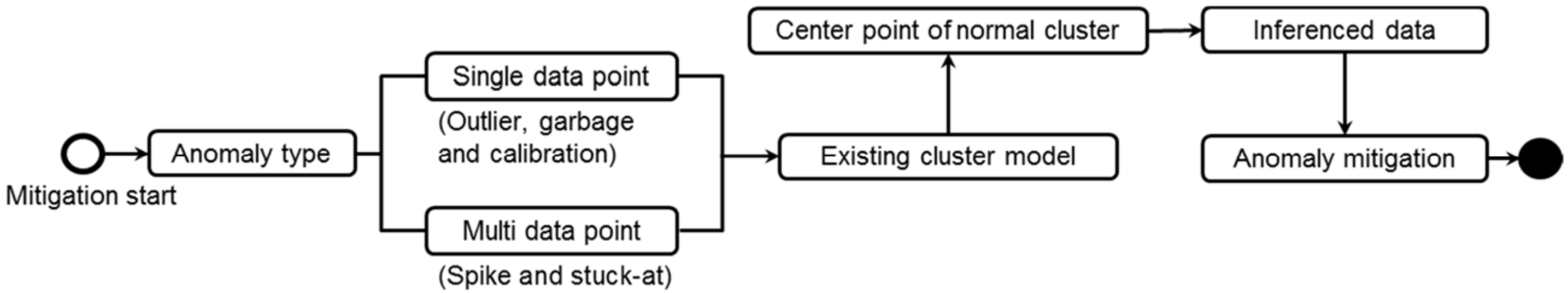
<Table 5.2> Pseudo code of the anomaly detection algorithm

Anomaly detection algorithm
Input: n: Data set, S_n: Acquired IoT data, f_n: assigned fault type
Output: E_{st}: Stuck-at, E_G: Garbage, E_O: Outlier, E_{sp}: Spike, E_c: Calibration
For i = 10: n do
if (σ_{i-10}^2 == 0 && $Correlation - coefficent_n$!= 0) **then** f_i = E_{st},
else if (S_i is null) **then** f_i = E_G,
else if (k-means identify S_i is outlier) **then** f_i = E_O,
else if (f_{i-1} == E_O and f_i == E_O) **then** f_i = E_{sp},
else if calculating actual time occurs error **then** f_i= E_c,
End if
End

5.2.2. Data anomaly mitigation algorithm

Having discussed techniques for data anomaly detection, data anomaly mitigation is also important to reduce the effectiveness of data anomaly. Data anomaly mitigation is the property that enables the system to recover in the event of failure and to continue the operation properly [92]. Data recovering in packet communication network researches have been studied actively, relatively few studies deal with sensor data anomaly mitigation. Most of data anomaly mitigation studies mainly aim at the single sensor situation and rarely discuss multifunction or multiple sensors faults such as spike and stuck-at in actual application [93-97]. Most studies applied neural network for data mitigation and only Shen and Wang [94] proposed a sparse relevance vector machine (RVM) predictor coupled with PCA to accomplish the multifunctional self-validating sensor.

In order to mitigate data anomaly rapidly and to minimize the mitigation cost, this study takes a partial strategy according to the characteristics of anomaly types. The anomaly mitigation model has two dataset types: one is the training data, which is set of historical normal data to train the adaptive mitigation model; the other type comprises inferenced data, which are used to evaluate the recovering capability and final output of mitigation. In the situation where IoT data are constantly acquired from the production lines, the fixed mitigation model cannot reliably determine the inferenced data. An adaptive mitigation model could provide an appropriate method to cope with a dynamic production environment. In other words, instead of inferencing mitigation value based on the fixed dataset, the adaptive model dynamically inference and modify the abnormal data depending on the current historical normal data set.



<Figure 5.4> Data anomaly mitigation process by adaptive mitigation model

Adaptive mitigation model is executed in accordance with data anomaly type; one is the single data point error and the other one is a multiple data point error. As I described in section 5.1, IoT data anomaly data type is classified into five categories. While these categories result from different causes, though it has its own detection method, these categories can be mitigated in terms of two aspects.

First, the single data point aspect (outlier, garbage, and calibration) is mitigated by the pattern of the historical data set, because those errors are the only single point that is out of the normal cluster. For this reason, inferenced data could be deducted using the center points of the normal cluster, which is already derived from the anomaly detection process. While outlier and garbage mitigation can be finished just using normal cluster center point, garbage requires one more process prior to applying the *k*-means outputs. As a calibration error results from the error of data preprocessing due to temporarily breakdown of network layers, there is need to resend anomaly data for identifying whether it is temporarily breakdown or not. If anomaly still occurs, mitigation should be applied.

Second, spike and stuck-at fault types are inferenced by the comparison between planned standard time and previous normal labeled actual production time. While an outlier is single point error apart from the normal cluster, spike and stuck-at fault types are sets of data which deviate from the normal data cluster. Furthermore, deviation of a dataset means that it has the possibility for changing an actual production pattern and it means that applying the changed situation is required for mitigation. For this reason, this anomaly types are also applied cluster centroid points in same with single point anomaly time mitigation method.

Considering that first spike anomaly data are almost same with outlier characteristics, outlier mitigation method is first applied. If the consecutive acquired IoT data are also regarded as outlier, the previous and target data are processed by means of existing detection model. This entwined process results from that IoT environment has a streaming data based that IoT data arrives continuously in the ordered sequence so that it is impossible for predicting consecutive following dataset. Following table describes pseudo code for overall anomaly response model.

<Table 5.3> Pseudo code of the anomaly response model

Anomaly response model
Input: n: Data set, S_n: Acquired IoT data, f_n: assigned fault type
$value$: Error data value
Output: (1) Fault type : E_{st}: Stuck-at, E_G: Garbage, E_O: Outlier,
E_{sp}: Spike, E_c: Calibration
(2) Fault input value and mitigation value
procedure Anomaly_Response();
execute K-means_traning();
do
get a identified normal IoT value set V_n for generating normal cluster;
execute K-means algorithm;
get a normal cluster centroid points (P_1, P_2)
while n dataset are executed;
init target data i, loading from Production_DB;
execute Procedure Anomaly_Det();
For i = 11: n do
if (σ^2_{i-10} == 0 && $Correlation-coefficent_n$!= 0)
then mark f_i = E_{st} as Stuck-at,
execute Anomaly_Miti_Multi($value$, i);
else if (S_i is null)
then mark f_i = E_G as Garbage,

```
                execute Anomaly_Miti_Single(value);
        else if (k-means identify S_i is outlier)
            then mark f_i = E_O as Outlier,
                execute Anomaly_Miti_Single(value);
        else if (f_{i-1} == E_O and f_i == E_O)
            then mark f_i = E_sp as Spike,
                execute Anomaly_Miti_Multi(value, i);
        else if calculating actual time occurs error
            then mark f_i= E_c as Calibration,
                execute Anomaly_Miti_Single(value);
    End;
procedure Anomaly_Miti_Single(value);
    get normal cluster center point;
    set value = P_1 or P_2;
    return value;
procedure Anomaly_Miti_Multi(value, i);
    get normal cluster center point;
    set value = P_1 or P_2;
    return value;
```

Chapter 6. Execution of experimental simulation study

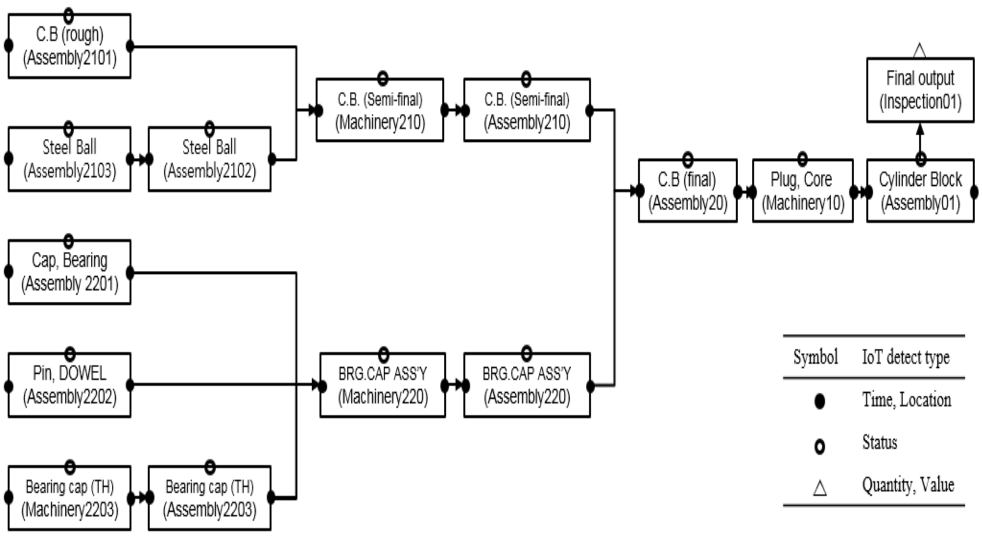
6.1. Creation of IoT-based smart factory

To demonstrate the suggested BPM and anomaly response model, this dissertation conducted a pilot test of an experimental factory simulation using C# and Oracle XE 11g. Many authors have attempted to develop an experimental simulation factory with a representations of all the major of a factory for use in validating their models [98-102]. The experimental simulation factory developed in this thesis included a job shop environment that contained two production lines with 12 machines each. The products in a line were processed by all the machines in the line consecutively, with each production line equipped with three IoT devices, which are positioned on the front and back sides of the machines to detect the start time and finish time. Manufacturing information systems and business logic follow the proposed IoT architecture and smart factory performance measurement logic, which was already discussed in the previous section. Furthermore, to simplify the shop floor environment and facilitate understanding of the scheme, the factory was assumed to produce only one type of engine part, namely cylinder block (BOM: 0 level), with Work-In-Process (BOM: 1 and 2 levels) and raw materials (BOM: 3 level). The production of the parts required specific types of machines and the estimated production standard time was in accordance with pre-defined production sequence as denoted at <Table 6.1>, <Figure 6.1>, and <Table 6.2>.

<Table 6.1> Bill of Materials and part explosion

Bill of Material								
BOM level				Description	Quantity required	Process	Machine	# of IoT
0	1	2	3					
1				Cylinder Block	1	Assembly01	ASS_M13	2
	10			PLUG,CORE	1	Machinery10	GRI_M12	2
	20			Cylinder Block (Final)	1	Assembly20	ASS_M11	2
		210		Cylinder Block (Semi-Final)	1	Machinery210	GRI_M10	2
						Assembly210	ASS_M09	2
			2101	Cylinder Block (Rough)	4	Assembly2101	ASS_M08	2
			2102	Steel Ball	4	Assembly2102	ASS_M07	2
						Assembly2103	ASS_M06	2
		220		BRG.CAP ASS'Y	1	Assembly220	ASS_M01	2
						Machinery220	GRI_M05	2
			2201	Cap, Bearing	4	Assembly2201	ASS_M04	2
			2202	PIN,DOWEL	4	Assembly2202	ASS_M03	2
			2203	Bearing Cap (TH)	4	Assembly2203	ASS_M02	2
						Machinery2203	GRI_M01	2

As Bill of Material indicated, this factory produces a part of a vehicle engine called cylinder block, one of which requires a large number of sub-parts for its production. For example, to produce one cylinder block, one plug, core and one cylinder block (final) are required. Furthermore, one steel ball and one cylinder block (rough) are needed to make one cylinder block (final). Following <Figure 6.1> shows the production process sequence according to Bill of Material and part explosion information.



<Figure 6.1> Production process sequence

6.2. Execution of factory simulation

6.2.1. Simulation of normal (Error-free) IoT data case

The numerical simulation of the normal production situation was based on the BOM and the part explosion information (see <Table 6.2>. This study assumed an order ID is '1000', an order quantity of 1,000 so that considering BOM, a total of 35,000 sub-parts should be produced. To facilitate the numerical simulation, a 24-hour production working calendar was assumed. Because the order information was specified, the exploded required quantity of sub-products was calculated based on the BOM. This research generated product codes as milestones of each process, with the generated codes totaling 35,000. Based on the planned production standard time (see <Table 6.2>), this study was able to calculate the planned production standard time of each sub-part and stored all the data as a basis for comparing the planned and actual times for a given product. To store the planned standard time stamp data in the database, the following equation was used.

$$\text{Planned standard time stamp} = \text{Previous standard time} + \text{Estimated time} \quad (5)$$

When BOM-3(or 2) level sub-products are assembled at a BOM-2 (or 1) level, the latest finish time at the BOM-3 level is the start time of the BOM-2 level production. For example, the grinding 210 start time is the latest finish time compared to the casting 2101 finish time and forging 2102 finish time.

<Table 6.2> Estimated standard time

Estimated standard time (Unit of measure: second)								
Process	Machine	Planned schedule				Sub-KPI		
		POT	Down time	Buffer time	Setup time	PBT (POT- Down time)	PUPT (PBT - Buffer time)	PPT (PUPT - Setup time)
Assembly01	ASS_M13	160	40	25	18	120	95	77
Assembly10	ASS_M12	160	40	25	18	120	95	77
Machinery20	GRI_M11	370	82	46	28	288	242	214
Assembly210	ASS_M09	160	40	25	18	120	95	77
Machinery210	GRI_M10	790	166	88	49	624	536	487
Assembly2101	ASS_M08	160	40	25	18	120	95	77
Assembly2102	ASS_M07	160	40	25	18	120	95	77
Assembly2102	ASS_M06	160	40	25	18	120	95	77
Assembly220	ASS_M01	160	40	25	18	120	95	77
Machinery220	GRI_M05	430	94	52	31	336	284	253
Assembly2201	ASS_M04	160	40	25	18	120	95	77
Assembly2202	ASS_M03	160	40	25	18	120	95	77
Assembly2203	ASS_M02	160	40	25	18	120	95	77
Machinery2203	GRI_M01	370	82	46	28	288	242	214

In the previous paragraph, the generation of the planned production time stamp was described. Based on the POT, it would be necessary to add more time if the actual time is longer than the planned time. Moreover, in a fully event-driven monitoring, it would be necessary to update developed KPIs accordingly and refresh the result. To satisfy the above requirements, this thesis assumed the followings:

- The actual delay time, actual down time, and actual setup time are added some seconds using triangular distribution to describe the machine status volatility.
- To derive the actual order execution time (AOET), first, actual setup time is added to triangular distribution and then, actual delay time is added to present actual unit busy time (AUBT). Finally, AOET can be derived when actual down time is provided according to triangular distribution.
- There was no complete breakdown of the machine.
- All the products were acceptable and passed the inspection process, implying a 'quality ratio' of 100%.

Using the estimated time as a deterministic actual time, this study attempted to determine the actual production timestamp. The cumulative changes in the actual production timestamp enable an analysis of the actual IoT-based data. Moreover, the planned time stamp can be compared with the actual time stamp to calculate the 'Net-OEE'. Based on equations (1), (2), (3), and (4), the Net-OEE of each process can finally be determined, as indicated in following table.

<Table 6.3> Simulation result for normal production case

	Normal production case (Error free)		
Process	Availability	Effectiveness	Net-OEE
Assembly01	0.947368421000	0.939024390000	0.889602053635
Assembly10	0.936842105000	0.939024390000	0.879717586174
Machinery20	0.979338843000	0.977168950000	0.956979508909
Assembly210	0.947368421000	0.927710843000	0.878883956477
Machinery210	0.988805970000	0.987829615000	0.976771820655
Assembly2101	0.947368421000	0.927710843000	0.878883956477
Assembly2102	0.982394366000	0.976833977000	0.959636195522
Assembly2102	0.947368421000	0.927710843000	0.878883956477
Assembly220	0.947368421000	0.927710843000	0.878883956477
Machinery220	0.947368421000	0.939024390000	0.889602053635
Assembly2201	0.936842105000	0.927710843000	0.869118578987
Assembly2202	0.947368421000	0.927710843000	0.878883956477
Assembly2203	0.947368421000	0.927710843000	0.878883956477
Machinery2203	0.975206612000	0.977168950000	0.952941621081

Although the foregoing was a hypothetical simulation and results do not have any practical values, it demonstrated the followings:

(1) The timestamp data detected by the IoT devices can be used to track and capture all the production process.

(2) A Smart factory performance measurement system can be used to automatically determine the 'Availability' and 'Effectiveness', and hence the real-time 'Net-OEE'.

6.2.2. Simulation of abnormal IoT data case

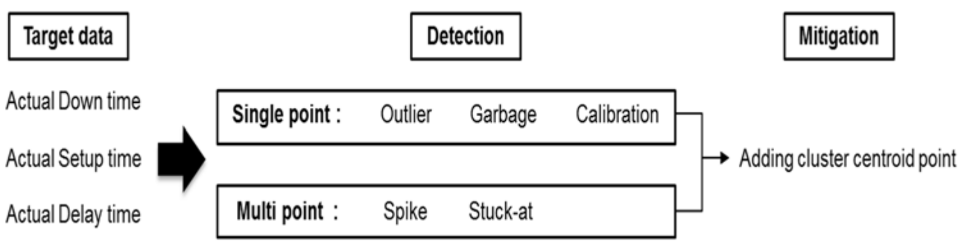
This section generates an IoT fault cases in accordance with the proposed data anomaly types within some probability distributions. When this study made an actual production IoT data, triangular distribution is applied into actual down time, actual delay time, and actual setup time in accordance with estimated actual time. After making the normal production case data, this study generated the data anomaly simulator for presenting the IoT failure. To present the frequency of the IoT failure, this dissertation applied error rate between 5 % ~ 100 %. However, to demonstrate the simulation process, following description presents the representative 20 % error rate which is identified that data response model generates best result. If failure labeled data is derived, one of actual delay time, actual setup time, and actual down time is selected randomly and one of five anomaly types is chosen randomly. Following table describes the simulation data result and to satisfy the data anomaly requirements, this study assumed followings:

<Table 6.4> Simulation data description

Feature	Value
Order quantity (Final output)	1,000
Total # of acquired IoT data	350,000
Production Start	17/03/01 09:00:00
Production Finish	17/03/18 09:48:46
# of outliers	1,849
# of spikes	2,263
# of stuck-ats	2,248
# of garbages	656
# of calibrations	265

- Outlier increases 5 ~ 35 seconds in accordance with normal actual production case.
- If certain IoT data is chosen to present the spike anomaly type, actual production time of selected IoT data and previous IoT data is also changed using outlier increasing method. The scope of previous IoT data is randomly chosen.
- Stuck-at anomaly type also changes the actual production time of previous IoT data. The previous actual production time should be same with chosen IoT data actual production time.
- Garbage anomaly type makes a certain actual production time within null data type.
- Calibration anomaly type makes a certain actual production time by putting string data type such as 'Sensing'.
- All the products satisfied inspection process, so that quality ratio is 100%.

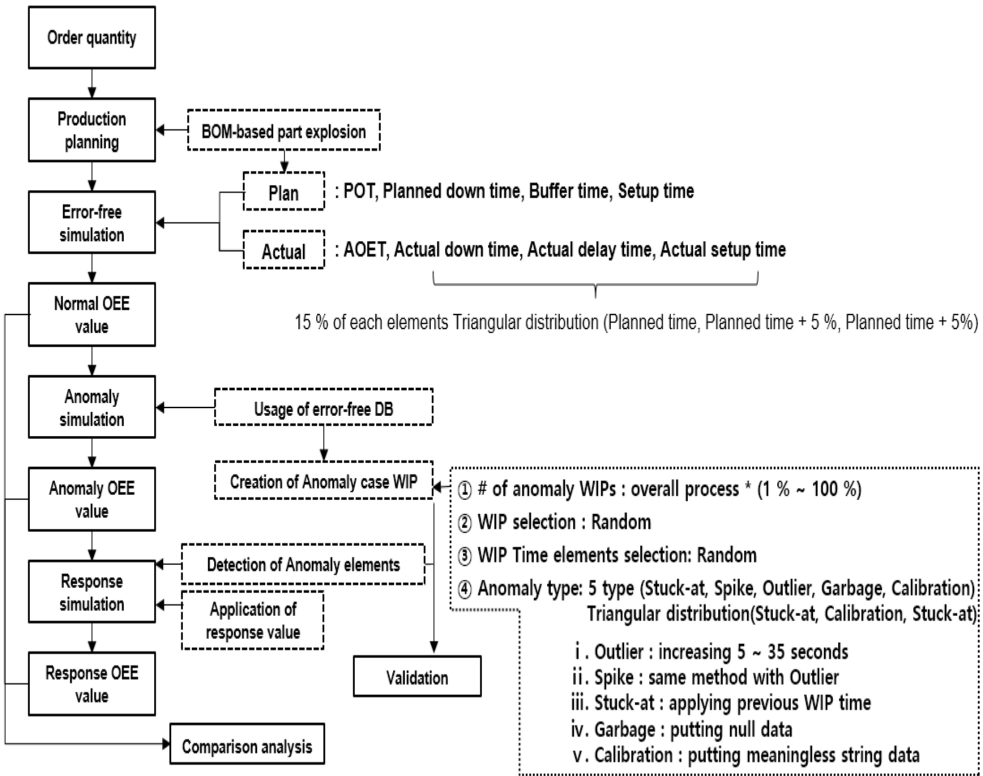
After simulating the abnormal IoT data case, data anomaly response model will be applied. The final output of data response model is also stored into database so that comparison among normal case, IoT data anomaly case, and response model applied case can be worked.



<Figure 6.2> Overall process of data anomaly response model

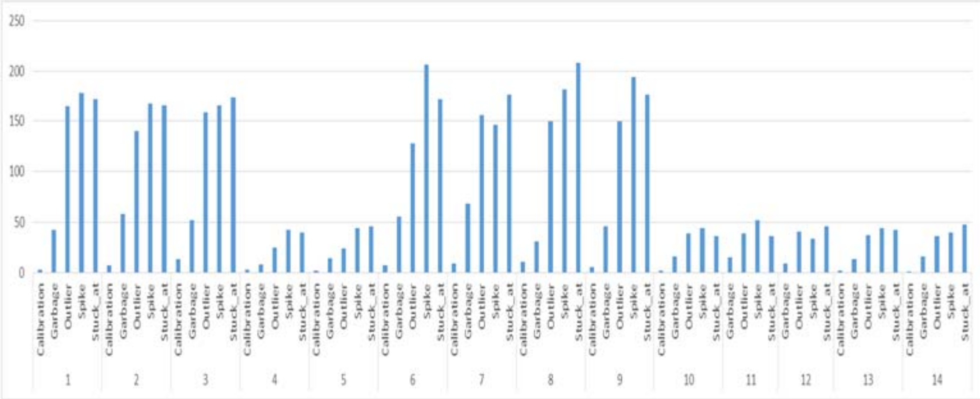
6.3. Results analysis and validation of the proposed algorithm

In order to obtain quantitative comparison analysis of the three different types of OEEs, production planning data (POT, Planned down time, Buffer time, and Setup time) and actual production data (AOET, Actual down time, Actual delay time, Actual setup time) should be collected. Once part explosion is implemented in terms of Bill of Materials, planning data can be collected regardless of the IoT anomaly cases because it is the deterministic defined values. <Figure 6.3> describes the overall process of simulation procedure and result analysis.



<Figure 6.3> Overall processes of simulation result analysis

In the case of normal production case, triangular distribution is applied for deriving actual production time in terms of planned values. Many related studies which simulated the OEE values for deriving actual production time applied triangular distribution [103-105]. Actual times are evenly derived from the planned values by using triangular distribution. Comparing planned and actual data, normal OEE values can be deducted. When IoT anomaly situation is considered, many scenarios are presented and stored into the data table. Comparing planned and deteriorated actual data, anomaly OEE values are presented. Finally, suggested data anomaly response model detects the expected anomaly data and mitigates the inappropriate values and derives the response OEE values. Analyzing the difference among the three different cases, deterioration effects of IoT anomaly data can be examined by comparing normal OEE and anomaly OEE and adequacy of anomaly response model can be established by comparing anomaly OEE and response OEE. Moreover, providing the confusion matrix and other derived values (precision, recall, and accuracy), performance of anomaly response model could be validated. Following paragraph describes the three different cases simulation and simulation validation results. In this thesis, simulation was executed 100 times which applied different IoT anomaly error rate from 1 % to 100 %. From 100 times simulation, reliability of proposed data anomaly response model could be validated by analyzing confusion matrix. Excluding confusion matrix, other validation results describe 15 % error rate simulation results. Following figure shows that data anomaly distribution follows the assumptions that Stuck-at, Spike, Outliers anomaly types occurred more than Calibration and Garbage anomaly types.



<Figure 6.4> Distribution of data anomaly types in terms of process

Following <Table 6.5> shows that total number of anomalies occurred according to the total number of data. <Table 6.6> describes the final output of OEE which simulated normal and fault cases.

<Table 6.5> Total number of anomalies in terms of process

Process_id	# of data	# of anomalies
1	4,000	834
2	4,000	802
3	4,000	815
4	1,000	182
5	1,000	225
6	4,000	881
7	4,000	888
8	4,000	845
9	4,000	791
10	1,000	208
11	1,000	172
12	1,000	221
13	1,000	205
14	1,000	212

<Table 6.6> Simulation result for normal and anomaly cases

	Normal case (Error free)			IoT anomaly case		
Process	Availability	Effectiveness	Net-OEE	Availability	Effectiveness	Net-OEE
Assembly01	0.947368421000	0.939024390000	0.889602053635	0.936842105000	0.950617284000	0.890578297392
Assembly10	0.936842105000	0.939024390000	0.879717586174	0.936842105000	0.950617284000	0.890578297392
Machinery20	0.979338843000	0.977168950000	0.956979508909	0.975206612000	0.981651376000	0.957312912554
Assembly210	0.947368421000	0.927710843000	0.878883956477	0.936842105000	0.950617284000	0.890578297392
Machinery210	0.988805970000	0.987829615000	0.976771820655	0.988805970000	0.995910020000	0.984761773359
Assembly2101	0.947368421000	0.927710843000	0.878883956477	0.936842105000	0.950617284000	0.890578297392
Assembly2102	0.982394366000	0.976833977000	0.959636195522	0.978873239000	0.984435798000	0.963637858176
Assembly2102	0.947368421000	0.927710843000	0.878883956477	0.936842105000	0.950617284000	0.890578297392
Assembly220	0.947368421000	0.927710843000	0.878883956477	0.936842105000	0.950617284000	0.890578297392
Machinery220	0.947368421000	0.939024390000	0.889602053635	0.936842105000	0.950617284000	0.890578297392
Assembly2201	0.936842105000	0.927710843000	0.869118578987	0.936842105000	0.962500000000	0.901710526063
Assembly2202	0.947368421000	0.927710843000	0.878883956477	0.947368421000	0.950617284000	0.900584795318
Assembly2203	0.947368421000	0.927710843000	0.878883956477	0.947368421000	0.950617284000	0.900584795318
Machinery2203	0.975206612000	0.977168950000	0.952941621081	0.979338843000	0.981651376000	0.961369322801

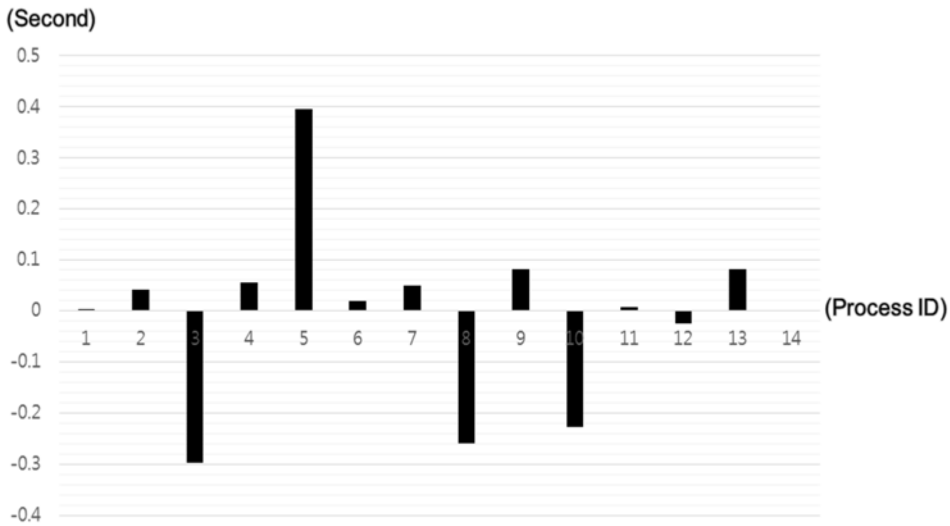
As <Table 6.6> shows the results of normal and anomaly cases, net-OEE are varied from -0.034 % ~ - 3.259 %. Considering that OEE is the final output of production performance, anomaly cases strongly deteriorates the actual production identification.

From the IoT anomaly cases, firstly, IoT data anomaly detection model is executed using *k*-means algorithm and data type analysis for alleviating the deterioration. To detect the Outlier and Spike anomaly types, *k*-means algorithm is applied and its centroid values are stored in accordance with process type from which are average of expected normal data set. Those centroid values are applied into the expected abnormal data to alleviate the deteriorate values and results are described in <Table 6.7>.

PROCESS_ID	ANOMALY_TIME_TYPE	CENTROID
1	cal_setup	27.7087378640777
1	cal_unit_down	23.4786795048143
1	cal_unit_delay	42.6914600550964
2	cal_unit_delay	42.7263157894737
2	cal_unit_down	23.4026490066225
2	cal_setup	27.699865410498
3	cal_setup	48.6640625
3	cal_unit_delay	84.6395806028834
3	cal_unit_down	33.5850066934404
4	cal_unit_delay	42.6700507614213
4	cal_unit_down	23.2319587628866
4	cal_setup	27.7386934673367
5	cal_setup	90.6354166666667
5	cal_unit_delay	168.735751295337
5	cal_unit_down	54.5257731958763
6	cal_setup	27.6495388669302
6	cal_unit_delay	42.7369826435247
6	cal_unit_down	23.5136897001304
7	cal_unit_down	36.4808673469388
7	cal_unit_delay	96.6847414880202
7	cal_setup	54.6705426356589
8	cal_setup	27.6930171277997
8	cal_unit_delay	42.6542056074766
8	cal_unit_down	23.3643617021277
9	cal_setup	27.6532033426184
9	cal_unit_delay	42.7063711911357
9	cal_unit_down	23.3486111111111
10	cal_setup	27.5608465608466
10	cal_unit_delay	42.781914893617
10	cal_unit_down	23.4
11	cal_setup	27.6756756756757
11	cal_unit_delay	42.6065573770492
11	cal_unit_down	23.4836956521739
12	cal_setup	27.6852791878173
12	cal_unit_down	23.45
12	cal_unit_delay	42.6102564102564
13	cal_unit_delay	42.586387434555
13	cal_setup	27.7
13	cal_unit_down	23.35
14	cal_unit_delay	84.6868131868132
14	cal_unit_down	33.4615384615385
14	cal_setup	48.7287234042553

<Figure 6.5> Snapshot of *k*-means centroid values DB table

To validate the *k*-means centroid points representing the mitigation input values, I compared the centroid points and average of sum of normal time elements in accordance with each processes. The min value of the difference between centroid and normal time elements is – 0.8348 second and max value of difference between them is 0.9949 second. The results show that differences between them are under 1 second. These values mean that centroid values are quite close to normal value trends and it is reasonable for applying them into the mitigation values. <Figure 6.6> shows the difference of normal value and centroid values according to each process.

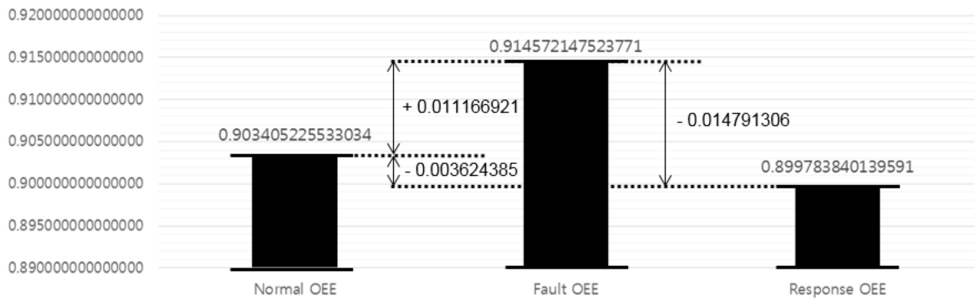


<Figure 6.6> Average difference between normal time and centroid values

<Table 6.7> Simulation result for anomaly and mitigation cases

	IoT anomaly case			Anomaly mitigation case		
Process	Availability	Effectiveness	Net-OEE	Availability	Effectiveness	Net-OEE
Assembly01	0.936842105000	0.950617284000	0.890578297392	0.940947368	0.930851064	0.875881859
Assembly10	0.936842105000	0.950617284000	0.890578297392	0.941263158	0.933559651	0.878725305
Machinery20	0.975206612000	0.981651376000	0.957312912554	0.97785124	0.974943052	0.953349272
Assembly210	0.936842105000	0.950617284000	0.890578297392	0.940315789	0.903861956	0.849915668
Machinery210	0.988805970000	0.995910020000	0.984761773359	0.989384328	0.986708809	0.976234232
Assembly2101	0.936842105000	0.950617284000	0.890578297392	0.938105263	0.939826681	0.881656356
Assembly2102	0.978873239000	0.984435798000	0.963637858176	0.980739437	0.978761267	0.959909774
Assembly2102	0.936842105000	0.950617284000	0.890578297392	0.941578947	0.932090546	0.877636835
Assembly220	0.936842105000	0.950617284000	0.890578297392	0.943473684	0.930851064	0.878233483
Machinery220	0.936842105000	0.950617284000	0.890578297392	0.941894737	0.934352627	0.880061822
Assembly2201	0.936842105000	0.962500000000	0.901710526063	0.941789474	0.935942628	0.881460915
Assembly2202	0.947368421000	0.950617284000	0.900584795318	0.944736842	0.922487121	0.871507569
Assembly2203	0.947368421000	0.950617284000	0.900584795318	0.942	0.933559651	0.879413191
Machinery2203	0.979338843000	0.981651376000	0.961369322801	0.976322314	0.976099252	0.95298748

Comparing the IoT anomaly and mitigation cases' OEE values, gaps of each of processes varied from - 4.066 % to 0.373 %. Those variations tend to close to the normal OEE values. For the convenience of gap analysis, I generated the average of each cases OEE summing up the all processes and final output could be provided, as <Figure 6.7> presented.



<Figure 6.7> Gap analysis of final OEE values

Based on the normal OEE value, the fault OEE values are increased by 1.116% and Response OEE values which is the results of data response model are decreased by 0.003%. This result shows that deterioration of IoT anomaly could be alleviated by applying the data anomaly response model. Normal OEE value and Response OEE values are quite same and this result means that applying proposed data anomaly response model could presents the real production situation regardless of the IoT data anomaly. Even though applying data response model generates similar normal OEE values, there is a need to investigate whether mitigated values are linearly related in normal time elements. To validate the similarity of normal, anomaly, mitigation values, this thesis applied correlation analysis methods and results are as follows.

<Table 6.8> Correlation analysis for Down time

		Normal	Anomaly	Mitigation
Normal	Pearson	1	***0.601*** **	***0.981*** **
	Sig.(2-tailed)		0.000	0.000
	N	35000	35000	35000
Anomaly	Pearson	***0.601*** **	1	***0.576*** **
	Sig.(2-tailed)	0.000		0.000
	N	35000	35000	35000
Mitigation	Pearson	***0.981*** **	***0.576*** **	1
	Sig.(2-tailed)	0.000	0.000	
	N	35000	35000	35000

**Correlation is significant at the 0.01 level (2-tailed)

<Table 6.9> Correlation analysis for Delay time

		Normal	Anomaly	Mitigation
Normal	Pearson	1	***0. 299*** **	***0. 932*** **
	Sig.(2-tailed)		0.000	0.000
	N	35000	35000	35000
Anomaly	Pearson	***0.299*** **	1	***0.398*** **
	Sig.(2-tailed)	0.000		0.000
	N	35000	35000	35000
Mitigation	Pearson	***0.932*** **	***0.398*** **	1
	Sig.(2-tailed)	0.000	0.000	
	N	35000	35000	35000

**Correlation is significant at the 0.01 level (2-tailed)

<Table 6.10> Correlation analysis for Setup time

		Normal	Anomaly	Mitigation
Normal	Pearson	1	***0.170*** **	***0.968*** **
	Sig.(2-tailed)		0.000	0.000
	N	35000	35000	35000
Anomaly	Pearson	***0.170*** **	1	***0.165*** **
	Sig.(2-tailed)	0.000		0.000
	N	35000	35000	35000
Mitigation	Pearson	***0.968*** **	***0.165*** **	1
	Sig.(2-tailed)	0.000	0.000	
	N	35000	35000	35000

**Correlation is significant at the 0.01 level (2-tailed)

The correlation results show that each of normal and anomaly have a similar linear association, because anomaly values are derived from normal values using triangular distribution. However, normal and mitigation correlation is stronger than normal and anomaly correlation. These results mean that even though normal values are deteriorated by the IoT anomaly (anomaly values), mitigation method alleviates the deterioration and helps closing to the normal values. By using correlation analysis, I could identify two findings: simulating anomaly cases follows normal cases without breaking dependent relation and mitigation cases follows normal cases so that strong relation could be deducted.

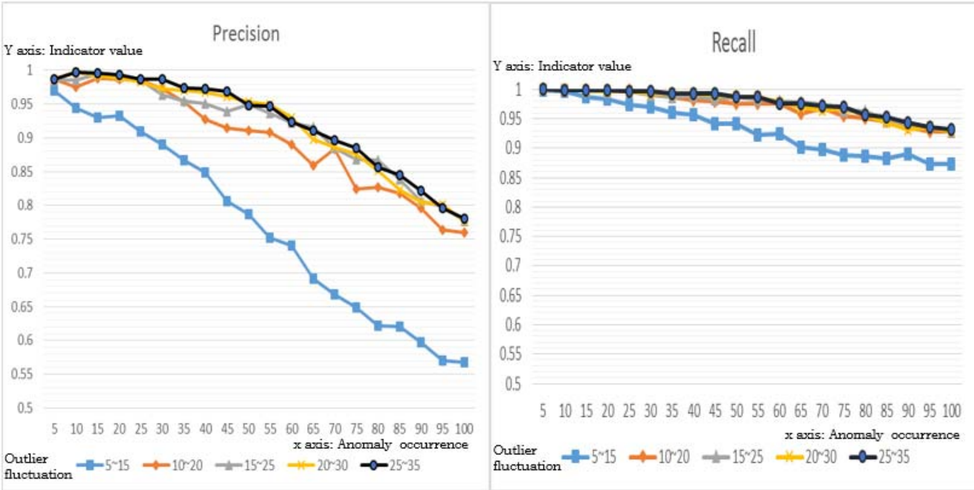
To assess the performance of proposed data anomaly response model, this thesis configured the confusion matrix. The performance of a proposed model is determined by how prediction reflects the actual observations. The number of correct and incorrect predictions is summarized in confusion matrix counting values and breaking down by each classes. True Positive (TP) is the number of correct predictions that an actual is positive. False Negative (FN) is the number of incorrect predictions that an actual is positive. False Positive (FP) is the number of incorrect of prediction that an actual is negative. True Negative is the number of correct prediction that an actual is negative.

<Table 6.11> Configuration of confusion matrix

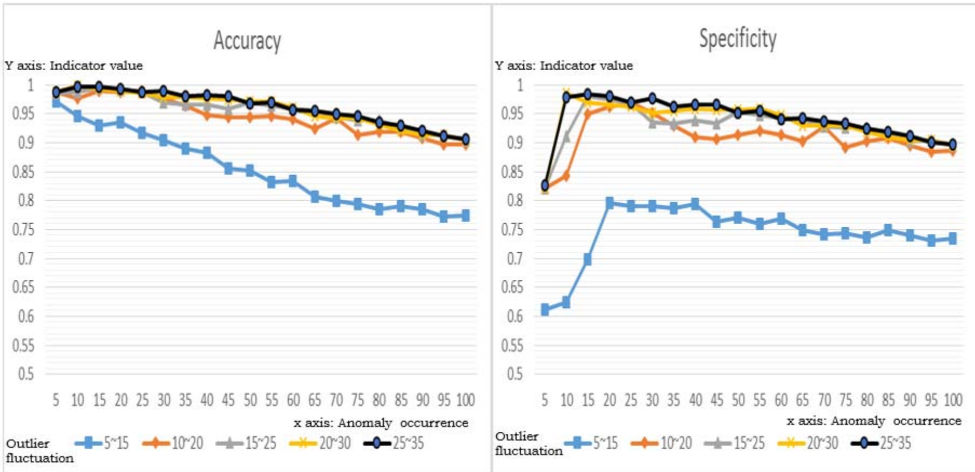
<table>
<tr><td colspan="2" rowspan="2"></td><td colspan="2">Prediction outcome</td></tr>
<tr><td>Positive prediction</td><td>Negative prediction</td></tr>
<tr><td rowspan="2">Actual value</td><td>Positive set</td><td>True Positive (TP)</td><td>False Negative (FN)</td></tr>
<tr><td>Negative set</td><td>False Positive (FP)</td><td>True Negative (TN)</td></tr>
</table>

Using confusion matrix components, five representative indicators could be derived and finally, I could identify the performance of proposed model. Precision is a measure of how many positive predictions were actual positive observations. The formula is TP/(TP + FP). Recall is a measure of how many actual positive observations were predicted correctly. The formula is TP/(TP + FN). Specificity is a measure of how many actual negative observations were predicted correctly. The formula is TN/(FP + TN). Accuracy is the ratio of correctly predicted observations. The formula is (TP + TN)/(TP + TN + FP + FN). F measure is the harmonic mean of accuracy and is defined as the weighted harmonic mean of the precision and recall. The formula is (2 * Prediction * Recall)/(Prediction + Recall).

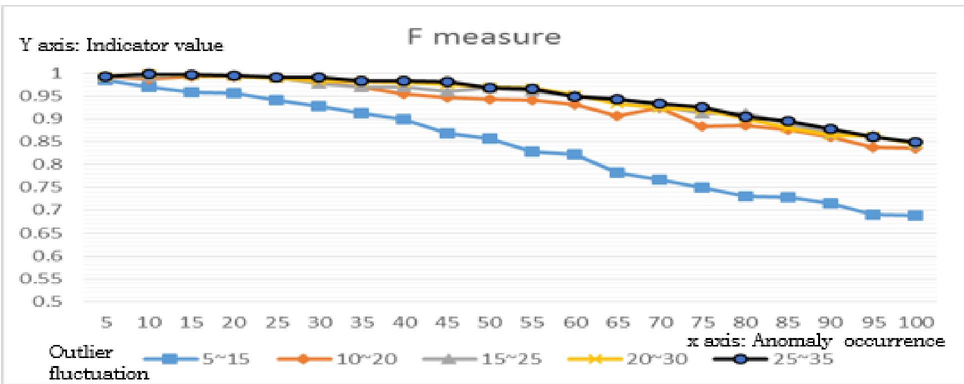
To derive confusion matrix with various simulation environments, two inputs are changed: one is outlier fluctuation value and the other is anomaly occurrence rate. If outlier is occurred, target time elements are added between fluctuation value and fluctuation value plus 10 seconds randomly. I simulated fluctuation values 5, 10, 15, 20 and 25 seconds. I could assess the *k*-means algorithm detection performance by differencing the fluctuation value. In addition, I simulated anomaly occurrence rates from 5 % to 100 % according to overall process quantity. If 8,750 data are produced and 5 % anomaly occurrence rate are decided, around 600 anomaly data are produced. I could assess the proposed model performance from excellent IoT environment (for example, anomaly occurrence rate is 5 %) to terrific IoT environment (for example, all produced data are anomaly data, namely, 100%). To derive the confusion matrix, I supposed 250 final products so that 8,750 sub-products are produced. Following figures are the results of confusion matrix indictors.



<Figure 6.8> Simulation results of Prediction and Recall

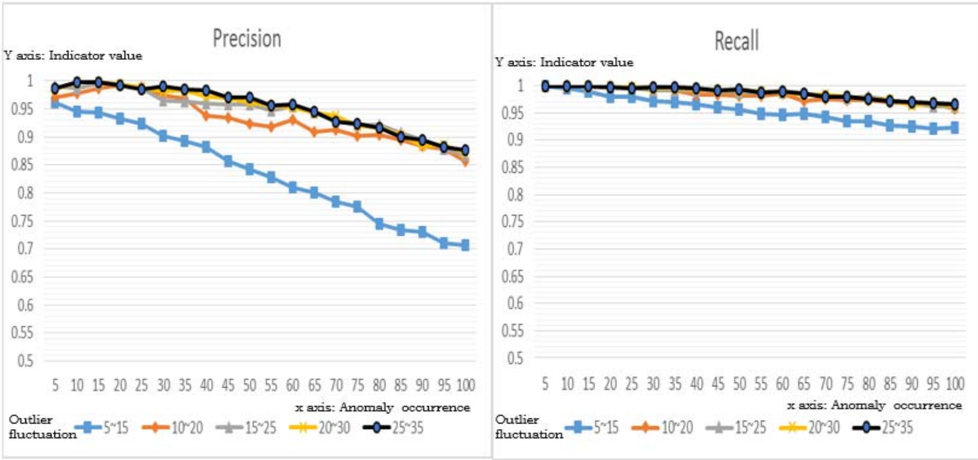


<Figure 6.9> Simulation results of Accuracy and Specificity

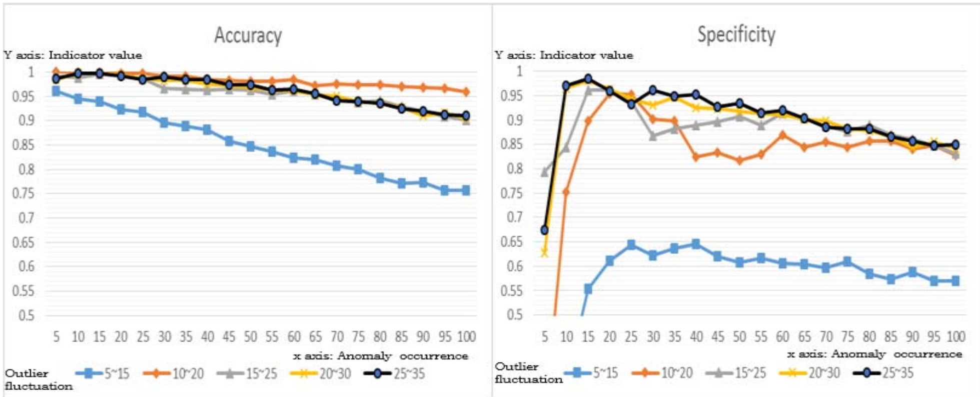


<Figure 6.10> Simulation results of F measure

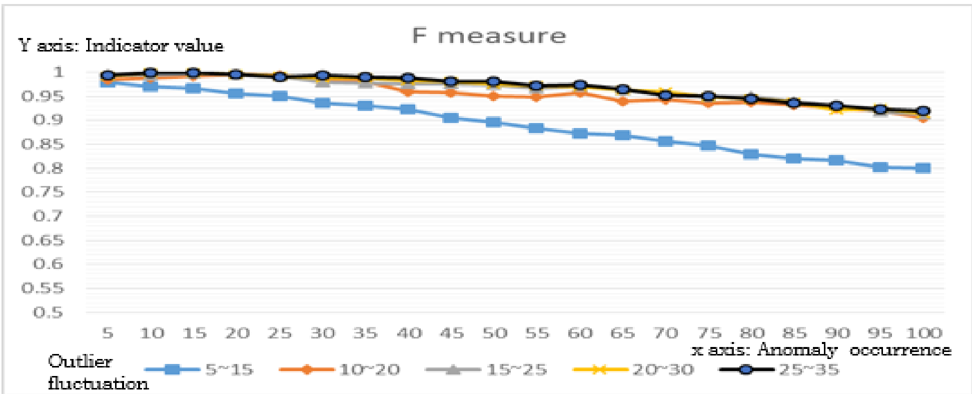
Drawing graphs applying the prediction, recall, accuracy, specificity, and F measure within various simulation environments, I could find out proposed response model performance. First, *K*-means algorithm which is used for detecting outlier and spike anomaly type is valued over the 10 seconds fluctuation times. All performance indicators are situated over the 0.8 which notifies that proposed model detects correct prediction. Second, considering that f-measure is derived from 0.6875 to 0.997872, proposed model has a valuable to find out data anomaly. Strength of detection algorithm provides better and precise mitigation values and finally, OEE values which alleviate the IoT fault could present the real situation values. Finally, however, provided that outlier happens around 5 ~ 15 seconds anomalies, *k*-means algorithm has a weakness to detect outlier and spike. From this reason, specificity indicator which explains how many negative observations could be identified as negative only presents 0.61111 when anomaly occurrence is 5 %. However, as anomaly occurrence increased, specificity is also increased. Over 15 % anomaly occurrence rate in 5 ~ 15 outlier fluctuation time shows over 0.7 specificity values so that proposed model could provide a valuable outputs. However, there is a need to analyze outlier and spike detection performance, because those indicators includes stuck-at, calibration, and garbage anomaly types of which detection rates are 100 %. Those anomaly types prevent identifying outlier and spike detection performance. In this reason, I simulated only considering outlier and spike anomaly types and results are as follows.



<Figure 6.11> Simulation results of Prediction and Recall (Outlier and Spike)

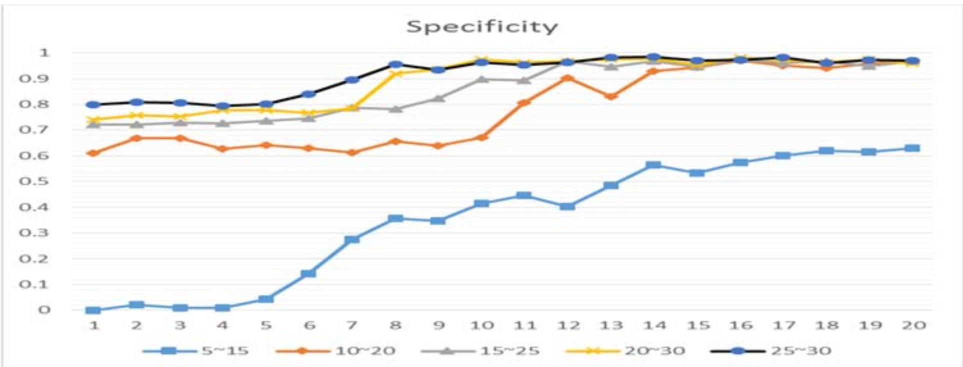


<Figure 6.12> Simulation results of Accuracy and Specificity (Outlier and Spike)



<Figure 6.13> Simulation results of F measure (Outlier and Spike)

Except for specificity indicator, other results trends are same with all type simulation results. However, specificity indicator is needed to be analyzed. As <Figure 6.12> indicates, when 5 % anomaly occurrence rate is happened, performance of *k*-means algorithm is low so that the possibility of predicting actual anomaly data as anomaly data is low. After analyzing the results data, I concluded that small sized anomaly data set could not be the criterion of anomaly cluster criterion, as a consequence, most inspected data are categorized into the normal cluster so that recall is almost 1. Also, I concluded that proposed model has a reasonable reliability regardless of error-rate within fluctuation time which overs 15 seconds. To find out more detail of model reliability, Simulation was performed by increasing 1 %. From the <Figure 6.14>, I could conclude that over 10 seconds outlier fluctuation has a strong values regardless of error rate. However, 5 ~ 15 seconds outlier fluctuation results shows that proposed model has a weak performance indicating specificity indicator from 0 ~ to 0.6. The performance of in this range is the limitation of this study.



<Figure 6.14> Simulation by increasing 1 % (Outlier and Spike)

Chapter 7. Conclusion

7.1. Discussion of findings and future works

The IoT concept is presently used in production lines to determine the production status through a variety of manufacturing systems. Interoperability is very important for the exchange of the acquired data, and our proposed architecture affords a stepping stone for improving the interoperability. In addition, it is important to note that our proposed BPM is not a universal solution that presents all the possible paths of a business process. However, our design of the fundamental business process in accordance with the MES standard and its hierarchical architecture. Our BPM can be used by small and medium size companies to predesign business processes and develop the performance measurement systems that utilize IoT. Furthermore, the Net-OEE determined from IoT data can be a more accurate index compared to what currently obtains, and also represents real-time performance.

With the proliferation of the IoT environment, a data response model is essential for deriving valuable information when the IoT fault occurs. While most recent studies present a data anomaly or outlier detection algorithm within the target data's point of views, this dissertation focuses on data anomaly types on the basis of data anomaly classification which has five fault types. This classification provides more robust mitigation effectiveness, because each data anomalies has to deal with a different mitigation approach. From this dissertation, the real-time Net-OEE can be derived from the IoT data and an acute anomaly response model enables an accurate KPI, even IoT failure occurs.

Present work could be improved through three focus areas for further study. First, though this dissertation only uses the Net-OEE KPIs, new and more useful KPIs and sub-KPIs should be developed to include the IoT data points. Traditional production performance-related KPIs only focus on production outputs and production plans. In the smart factory environment, more sensitive or broad perspective data can be acquired and used for measuring performance. Future research needs to investigate the relationships between current KPIs and their sources for introducing the new IoT related KPIs. Second, validating the data anomaly response model with other current methods and models is needed to show the effectiveness. As the literature review described, few related studies are now published so that comparison is impossible. However, I think that data anomaly detection is rising research topic. In the course of time, many methods and algorithm will be applied for detecting data anomaly and comparison will be able to execute. Finally, developing an IoT network topology is a requirement for presenting the smart factory information system. The main differentiator of the smart factory is the existence of Cyber Physical System (CPS). In brief, CPS is a small dependent manufacturing information system for managing the entire production environment. A small-sized CPS increases the complexity of the network components and the volume of data. To enhance the performance measurement effectiveness and reliability, designing the topology is a prerequisite for managing data flow and network architecture.

7.2. Conclusion

In general terms, this study demonstrates a way of designing the real-time Net-OEE calculation process, though there is a possibility of IoT data failure. A hierarchical KPI calculation structure was proposed to describe the Net-OEE calculation based on the International standard, ISO-22400. The Net-OEE KPI and sub-KPIs were defined and IoT applicable parts were identified for the design of the IoT-based architecture. Furthermore, for configuration of the architecture, this dissertation presented an ERD that enables the modelling of the complex relationships among the various data entities. This study combined the performance measurement model with the output of the architecture ERD to develop a BPM model, the accuracy of which was validated by a simulation factory. Particularly, this study generated the IoT fault classification types, which comprises five types as the basis of the proposed architecture level. In the end, this dissertation developed the IoT anomaly detection algorithm to evaluate the IoT-related data containing possible fault classifications and to automatically configure an appropriate detection set. Furthermore, based on an identified anomaly data sets, a data anomaly mitigation algorithm was applied to substitute anomaly data with newly inferenced data. This mitigation enhanced the reliability of the error-free calculation of the real-time Net-OEE. Finally, an experimental simulation shows the effectiveness of the proposed smart factory performance system. The employed factory, which consisted of sets of acquired machines and production processes, was developed for enhanced realism.

APPENDIX

Appendix contents referenced ISO-22400-2 standards [64]. Though ISO-22400 present 34 KPIs, this study excludes six KPIs considering duplication and vagueness: comprehensive energy consumption, inventory turns, storage and transportation loss, other loss ratio, mean operating time between failures, and integrated goods ratio.

KPI	Formula
Description	
Worker efficiency (작업자 효율)	= APWT / APAT
It considers the relationship between the working hours related to production orders and the total attendance time of the employees.	
Allocation ratio (분배 비율)	= AUBT / AOET
It is the relationship of the complete busy time over all involved work units and work centers to the throughput time.	
Throughput rate (처리율)	= PQ / AOET
It is an index for the performance of a process. This performance indicator is an important index for the efficiency in production.	
Allocation efficiency (할당 효율)	= AUBT / PBT
It is the ratio between the real allocation time of a work unit and planned time for allocating the machine.	
Utilization efficiency (이용 효율)	= APT / AUBT
It is the ratio between the actual production time (APT) and the actual unit busy time (AUBT).	
Overall Equipment Effectiveness (전체설비효율)	= Availability * Effectiveness * Quality ratio
It is an indicator for the efficiency of work units, work centers and areas with several work units or an entire work center.	
Net OEE (개별설비효율)	= AUPT / PBT * Effectiveness * Quality ratio

It is comparable with the OEE index but it includes the setup time within the availability. It indicates losses by work unit delays, cycle time and losses by rework.	
Availability (효용성)	= AUPT / PUPT
It indicates how strongly the capacity of a work unit for the production is used in relation to the available capacity.	
Effectiveness (유효성)	= PPT / APT
It is the index for the performance of a work unit. It represents the relationship between the planned target cycle and the actual cycle.	
Quality ratio (규질비)	= GQ / PQ
It is the relationship between the good quantity (GQ) and the produced quantity (PQ).	
Setup rate (준비율)	= AUST / AUPT
It indicates the relative loss of value adding opportunity for the work unit. The setup ratio has to be considered especially when the order lot size is decreasing which may happen in a response to the demand for a flexible supply chain.	
Technical efficiency (기술 효율)	= APT / (APT + ADET)
It is the relationship between the production time period and production time period including malfunction-caused interruption.	
Production process ratio (생산처리 비율)	= APT / AOET
It is an index for the efficiency of the production. It defines the relationship between the production time and the whole throughput time of a production order.	
Actual to planned scrap ratio (계획 대비 실제 스크랩)	= SQ / PSQ
The actual to planned scrap ratio calculated as the scrap quantity divided by the planned scrap quantity indicated how much scrap was actually produced compared with the expected value.	
First pass yield (초기 수율)	= GP / IP
It is a direct indicator for the process quality related to work place and product.	

Scrap ratio (스크랩 비율)	= SQ / PQ
It is the relationship between crap quantity and produced quantity.	
Rework ratio (재작업 비율)	= RQ / PQ
It is the relationship between rework quantity and produced quantity.	
Fall off ratio (감소비율)	= SQ / PQ of the first production order
It indicates the fall off quantity in each production operation step in relation to the produced quantity and produced quantity.	
Machine capability (기계능력지수)	= (USL - LSL) / (6 * s)
It is the relationship between the dispersion of a process and the specification limits (USL, LSL).	
Critical machine capability (중요기계 능력 지수)	= (USL - x) / (3 * S) or (x - LSL) / (3 * s)
It indicates the ability of a machine or a work mechanism to produce the specified quality for a specific characteristic.	
Process capability (공정 능력지수)	= (USL - LSL) / (6 * σ)
It is the relationship between the dispersion of a process and the specification limits. The method compares the range between the specification limits and the 6-sigma process dispersion	
Critical process capability (중요공정 능력지수)	= (USL – x) / (3 * σ) or (x – LSL) / (3 * σ)
It is the relationship between the dispersion of a process and the upper or lower specification limit and its average)	
Finished goods ratio (최종제품비율)	= GQ / CM
It is the ratio of the good quantity produced to the consumed material.	
Production loss ratio (생산 손실 비율)	= PL / CM
It is the relationship of quantity lost during production to the consumed material.	
Equipment load rate (설비 부하율)	= PQ / EPC
It provides information about the ratio of produced quantity in	

relation to the maximum equipment production capacity.	
Mean time to failure (평균 고장시간)	$= \sum TTF / (FE + 1)$
It is an indicator of expected system reliability calculated on a statistical basis from the known failure rates of various components of the work unit.	
Mean time to repair (평균 수리시간)	$= \sum TTR / (FE + 1)$
It is the average time that an item required to restore a failed component in a work unit.	
Corrective maintenance ratio (개량 보전 비율)	$= CMT / (CMT + PMT)$
It considers the corrective maintenance time in relation to the total maintenance expressed as the sum of corrective maintenance time and planned maintenance time. It reveals the magnitude of corrective tasks within all maintenance activities performed in a work unit.	

Bibliography

[1] G. Hwang, S. Han, S. Jun, and J. Park, Operational Performance Metrics in Manufacturing Process: Based on SCOR Model and RFID Technology, *International Journal of Innovation, Management and Technology,* vol. 5, 2014.

[2] G. Hwang, J. Lee, J. Park, and T.-W. Chang, Developing performance measurement system for Internet of Things and smart factory environment, *International Journal of Production Research,* vol. 55, pp. 2590-2602, 2017/05/03 2017.

[3] S. Mejjaouli and R. Babiceanu, Holonic Condition Monitoring and Fault-Recovery System for Sustainable Manufacturing Enterprises, in *Service Orientation in Holonic and Multi-Agent Manufacturing and Robotics*. vol. 544, T. Borangiu, D. Trentesaux, and A. Thomas, Eds., ed: Springer International Publishing, pp. 31-46, 2014.

[4] D. Kern, R. Moser, E. Hartmann, and M. Moder, Supply risk management: model development and empirical analysis, *International Journal of Physical Distribution & Logistics Management,* vol. 42, pp. 60-82, 2012.

[5] K. K. B. Hon, Performance and Evaluation of Manufacturing Systems, *CIRP Annals - Manufacturing Technology,* vol. 54, pp. 139-154, 2005/01/01 2005.

[6] R. G. F. Braz, L. F. Scavarda, and R. A. Martins, Reviewing and improving performance measurement systems: An action research, *International Journal of Production Economics,* vol. 133, pp. 751-760, 10// 2011.

[7] P. Taticchi, K. Balachandran, and F. Tonelli, Performance measurement and management systems: state of the art, guidelines for design and challenges, *Measuring Business Excellence,* vol. 16, pp. 41-54, 2012.

[8] V. F. Yu and K.-J. Hu, An integrated fuzzy multi-criteria approach for the performance evaluation of multiple manufacturing plants, *Computers & Industrial Engineering,* vol. 58, pp. 269-277, 3// 2010.

[9] S. A. Melnyk, D. M. Stewart, and M. Swink, Metrics and performance measurement in operations management: dealing with the metrics maze, *Journal of Operations Management,* vol. 22, pp. 209-218, 6// 2004.

[10] R. S. Kaplan and D. P. Norton, Using the balanced scorecard as a strategic management system, ed: Harvard business review Boston, MA 1996.

[11] H. S. Al-Dhaafri, A. K. Al-Swidi, and A. A. Al-Ansi, Organizational Excellence as the Driver for Organizational Performance: A Study on Dubai Police, *International Journal of Business and Management,* vol. 11, p. 47, 2016.

[12] P. Sampaio, P. Saraiva, and A. Monteiro, A comparison and usage

overview of business excellence models, *The TQM Journal,* vol. 24, pp. 181-200, 2012.

[13] Q. Dai, R. Zhong, G. Q. Huang, T. Qu, T. Zhang, and T. Y. Luo, Radio frequency identification-enabled real-time manufacturing execution system: a case study in an automotive part manufacturer, *International Journal of Computer Integrated Manufacturing,* vol. 25, pp. 51-65, 2012/01/01 2011.

[14] P. Muchiri and L. Pintelon, Performance measurement using overall equipment effectiveness (OEE): literature review and practical application discussion, *International Journal of Production Research,* vol. 46, pp. 3517-3535, 2008/07/01 2008.

[15] B. Saenz de Ugarte, A. Artiba, and R. Pellerin, Manufacturing execution system – a literature review, *Production Planning & Control,* vol. 20, pp. 525-539, 2009/09/01 2009.

[16] R. Y. Zhong, Q. Y. Dai, T. Qu, G. J. Hu, and G. Q. Huang, RFID-enabled real-time manufacturing execution system for mass-customization production, *Robotics and Computer-Integrated Manufacturing,* vol. 29, pp. 283-292, 4// 2013.

[17] M. Wollschlaeger, T. Sauter, and J. Jasperneite, The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0, *IEEE Industrial Electronics Magazine,* vol. 11, pp. 17-27, 2017.

[18] Y. Zhang, G. Zhang, J. Wang, S. Sun, S. Si, and T. Yang, Real-time information capturing and integration framework of the internet of manufacturing things, *International Journal of Computer Integrated Manufacturing,* vol. 28, pp. 811-822, 2015/08/03 2015.

[19] T. Qu, M. Thürer, J. Wang, Z. Wang, H. Fu, C. Li*, et al.*, System dynamics analysis for an Internet-of-Things-enabled production logistics system, *International Journal of Production Research,* pp. 1-28, 2016.

[20] N. Papakostas, J. O. Connor, and G. Byrne, Internet of things technologies in manufacturing: Application areas, challenges and outlook, in *2016 International Conference on Information Society (i-Society)*, pp. 126-131, 2016.

[21] A. Theorin, K. Bengtsson, J. Provost, M. Lieder, C. Johnsson, T. Lundholm*, et al.*, An event-driven manufacturing information system architecture for Industry 4.0, *International Journal of Production Research,* pp. 1-15, 2016.

[22] S. Wang, J. Wan, D. Li, and C. Zhang, Implementing Smart Factory of Industrie 4.0: An Outlook, *International Journal of Distributed Sensor Networks,* vol. 12, p. 3159805, 2016.

[23] J. Lee, B. Bagheri, and H.-A. Kao, A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems, *Manufacturing Letters,* vol. 3, pp. 18-23, 1// 2015.

[24] N. Jazdi, Cyber physical systems in the context of Industry 4.0, in *2014 IEEE International Conference on Automation, Quality and Testing, Robotics*, pp. 1-4, 2014.

[25] E. M. Frazzon, J. Hartmann, T. Makuschewitz, and B. Scholz-Reiter,

Towards Socio-Cyber-Physical Systems in Production Networks, *Procedia CIRP,* vol. 7, pp. 49-54, 2013/01/01 2013.

[26] Q. Gao, F. Li, and C. Chen, Research of Internet of Things applied to manufacturing execution system, in *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 661-665, 2015.

[27] A. Sheth, Internet of Things to Smart IoT Through Semantic, Cognitive, and Perceptual Computing, *IEEE Intelligent Systems,* vol. 31, pp. 108-112, 2016.

[28] S. Ghimire, F. Luis-Ferreira, T. Nodehi, and R. Jardim-Goncalves, IoT based situational awareness framework for real-time project management, *International Journal of Computer Integrated Manufacturing,* vol. 30, pp. 74-83, 2017/01/02 2017.

[29] C. Rongier, M. Lauras, F. Galasso, and D. Gourc, Towards a crisis performance-measurement system, *International Journal of Computer Integrated Manufacturing,* vol. 26, pp. 1087-1102, 2013/11/01 2013.

[30] P. Garengo and M. K. Sharma, Performance measurement system contingency factors: a cross analysis of Italian and Indian SMEs, *Production Planning & Control,* vol. 25, pp. 220-240, 2014/02/17 2014.

[31] C.-F. Chien, P.-C. Chu, and L. Zhao, Overall resource effectiveness indices for total resource management and case studies, *International Journal of Industrial Engineering,* vol. 22, 2015.

[32] R. Bhagwat and M. K. Sharma, Performance measurement of supply chain management using the analytical hierarchy process, *Production Planning & Control,* vol. 18, pp. 666-680, 2007/12/01 2007.

[33] M. R. Abdi and A. W. Labib, Performance evaluation of reconfigurable manufacturing systems via holonic architecture and the analytic network process, *International Journal of Production Research,* vol. 49, pp. 1319-1335, 2011/03/01 2011.

[34] K. K. Choong, Has this large number of performance measurement publications contributed to its better understanding? A systematic review for research and applications, *International Journal of Production Research,* vol. 52, pp. 4174-4197, 2014/07/18 2014.

[35] C. Ou-Yang and Y. D. Lin, BPMN-based business process model feasibility analysis: a petri net approach, *International Journal of Production Research,* vol. 46, pp. 3763-3781, 2008/07/15 2008.

[36] K. D. Barber, F. W. Dewhurst, R. L. D. H. Burns, and J. B. B. Rogers, Business-process modelling and simulation for manufacturing management: A practical way forward, *Business Process Management Journal,* vol. 9, pp. 527-542, 2003.

[37] S. Buchwald, T. Bauer, and M. Reichert, Bridging the gap between business process models and service composition specifications, *Service Life Cycle Tools and Technologies: Methods, Trends and Advances,* pp. 124-153, 2011.

[38] L. Prades, F. Romero, A. Estruch, A. García-Dominguez, and J. Serrano, Defining a Methodology to Design and Implement Business Process Models in BPMN According to the Standard ANSI/ISA-95 in a Manufacturing Enterprise, *Procedia Engineering,* vol. 63, pp. 115-

122, // 2013.
[39] E. Desbiens and A. Chaabane, Visual notation approach for MES integration, in *IIE Annual Conference. Proceedings*, p. 321, 2014.
[40] S. Meyer, A. Ruppen, and C. Magerkurth, Internet of Things-Aware Process Modeling: Integrating IoT Devices as Business Process Resources, in *Advanced Information Systems Engineering*. vol. 7908, C. Salinesi, M. Norrie, and Ó. Pastor, Eds., ed: Springer Berlin Heidelberg, pp. 84-98, 2013.
[41] M. Helu, D. Libes, J. Lubell, K. Lyons, and K. C. Morris, Enabling Smart Manufacturing Technologies for Decision-Making Support, p. V01BT02A035, 2016.
[42] M. Bauer, M. Lucke, C. Johnsson, I. Harjunkoski, and J. C. Schlake, KPIs as the interface between scheduling and control, *IFAC-PapersOnLine,* vol. 49, pp. 687-692, // 2016.
[43] S. Yoon, J. Um, S.-H. Suh, I. Stroud, and J.-S. Yoon, Smart Factory Information Service Bus (SIBUS) for manufacturing application: requirement, architecture and implementation, *Journal of Intelligent Manufacturing,* pp. 1-20, 2016.
[44] Y. Fukuda and R. Patzke, Standardization of Key Performance Indecator for manufacturing execution system, in *Proceedings of SICE Annual Conference 2010*, pp. 263-265, 2010.
[45] P. H. Tsarouhas, Evaluation of overall equipment effectiveness in the beverage industry: a case study, *International Journal of Production Research,* vol. 51, pp. 515-523, 2013/01/15 2013.
[46] B. Marcello, F. Marco, and Z. Francesco, Overall equipment effectiveness of a manufacturing line (OEEML): An integrated approach to assess systems performance, *Journal of Manufacturing Technology Management,* vol. 20, pp. 8-29, 2008/12/26 2008.
[47] N. Kang, C. Zhao, J. Li, and J. A. Horst, A Hierarchical structure of key performance indicators for operation management and continuous improvement in production systems, *International Journal of Production Research,* vol. 54, pp. 6333-6350, 2016/11/01 2016.
[48] R. Henßen and M. Schleipen, Interoperability between OPC UA and AutomationML, *Procedia CIRP,* vol. 25, pp. 297-304, 2014/01/01 2014.
[49] I. Seilonen, T. Tuovinen, J. Elovaara, I. Tuomi, and T. Oksanen, Aggregating OPC UA servers for monitoring manufacturing systems and mobile work machines, in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1-4, 2016.
[50] M. Rentschler, H. Trsek, and L. Dürkop, OPC UA extension for IP Auto-Configuration in Cyber-Physical Systems, in *Industrial Informatics (INDIN), 2016 IEEE 17th Conference on*, 2016.
[51] M. Schleipen, S.-S. Gilani, T. Bischoff, and J. Pfrommer, OPC UA & Industrie 4.0 - Enabling Technology with High Diversity and Variability, *Procedia CIRP,* vol. 57, pp. 315-320, 2016/01/01 2016.
[52] W. Oh, A simple sensor fault detection algorithm, in *Proc. of Int. Conf. on Computer Science, Data Mining&Mechanical Engineering*, pp. 132-134, 2015.

[53] C. Wang, H. T. Vo, and P. Ni, An IoT Application for Fault Diagnosis and Prediction, in *2015 IEEE International Conference on Data Science and Data Intensive Systems*, pp. 726-731, 2015.

[54] Y. Liu, Y. Yang, X. Lv, and L. Wang, A self-learning sensor fault detection framework for industry monitoring IoT, *Mathematical problems in engineering,* vol. 2013, 2013.

[55] S. Misra, A. Gupta, P. V. Krishna, H. Agarwal, and M. S. Obaidat, An adaptive learning approach for fault-tolerant routing in Internet of Things, in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 815-819, 2012.

[56] S. Zhou, K. J. Lin, and C. S. Shih, Device clustering for fault monitoring in Internet of Things systems, in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pp. 228-233, 2015.

[57] N. Bayar, S. Darmoul, S. Hajri-Gabouj, and H. Pierreval, Fault detection, diagnosis and recovery using Artificial Immune Systems: A review, *Engineering Applications of Artificial Intelligence,* vol. 46, Part A, pp. 43-57, 11// 2015.

[58] I. E. Commission, IEC 62264-1 Enterprise-control system integration–Part 1: Models and terminology, *IEC, Genf,* 2003.

[59] I. Grangel-González, L. Halilaj, G. Coskun, S. Auer, D. Collarana, and M. Hoffmeister, Towards a Semantic Administrative Shell for Industry 4.0 Components, in *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*, pp. 230-237, 2016.

[60] H. Unver, An ISA-95-based manufacturing intelligence system in support of lean initiatives, *The International Journal of Advanced Manufacturing Technology,* vol. 65, pp. 853-866, 2013/03/01 2013.

[61] J. Virta, I. Seilonen, A. Tuomi, and K. Koskinen, SOA-Based integration for batch process management with OPC UA and ISA-88/95, in *2010 IEEE 15th Conference on Emerging Technologies & Factory Automation (ETFA 2010)*, pp. 1-8, 2010.

[62] I. E. Commission, IEC 62264-3 Enterprise-control system integration-Part 3: Activity Models of Manufacturing Operations Management, *IEC, Genf,* 2004.

[63] P. N. Muchiri, L. Pintelon, H. Martin, and A.-M. De Meyer, Empirical analysis of maintenance performance measurement in Belgian industries, *International Journal of Production Research,* vol. 48, pp. 5905-5924, 2010/10/15 2009.

[64] I. O. f. Standardization, ISO 22400-2:2014:Automation systems and integration -- Key performance indicators (KPIs) for manufacturing operations management -- Part 2: Definitions and descriptions, in *ISO 22400-2:2014*, ed. Geneva: International Organization for Standardization 2014.

[65] S. H. Huang, J. P. Dismukes, J. Shi, Q. I. Su, M. A. Razzak, R. Bodhale, *et al.*, Manufacturing productivity improvement using effectiveness metrics and simulation analysis, *International Journal of Production Research,* vol. 41, pp. 513-527, 2003/01/01 2003.

[66] R. L. Ackoff, From data to wisdom, *Journal of applied systems analysis,* vol. 16, pp. 3-9, 1989.

[67] R. Duan, X. Chen, and T. Xing, A QoS Architecture for IOT, in *2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing*, pp. 717-720, 2011.

[68] W. Rafique and M. A. Shah, Performance evaluation of IoT network infrastructure, in *2016 22nd International Conference on Automation and Computing (ICAC)*, pp. 348-353, 2016.

[69] M. Aazam, I. Khan, A. A. Alsaffar, and E. N. Huh, Cloud of Things: Integrating Internet of Things and cloud computing and the issues involved, in *Proceedings of 2014 11th International Bhurban Conference on Applied Sciences & Technology (IBCAST) Islamabad, Pakistan, 14th - 18th January, 2014*, pp. 414-419, 2014.

[70] Z. Zhu, J. Tan, H. Ren, W. Ni, and Q. Guan, RFID application in manufacturing: A case study on a pilot RFID project in household appliance production, *International Journal of Computer Integrated Manufacturing,* vol. 25, pp. 3-10, 2012/01/01 2012.

[71] M. Ma, P. Wang, and C. H. Chu, Data Management for Internet of Things: Challenges, Approaches and Opportunities, in *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, pp. 1144-1151, 2013.

[72] W. Miao, L. Ting-Jie, L. Fei-Yang, S. Jing, and D. Hui-Ying, Research on the architecture of Internet of Things, in *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, pp. V5-484-V5-487, 2010.

[73] A. Gyrard, S. K. Datta, C. Bonnet, and K. Boudaoud, Standardizing generic cross-domain applications in Internet of Things, in *2014 IEEE Globecom Workshops (GC Wkshps)*, pp. 589-594, 2014.

[74] M. Momotko and B. Nowicki, Visualisation of (distributed) process execution based on extended BPMN, in *14th International Workshop on Database and Expert Systems Applications, 2003. Proceedings.*, pp. 280-284, 2003.

[75] W. Mahnke, S.-H. Leitner, and M. Damm, *OPC unified architecture*: Springer Science & Business Media, 2009.

[76] J. Imtiaz and J. Jasperneite, Scalability of OPC-UA down to the chip level enables “Internet of Things”, in *2013 11th IEEE International Conference on Industrial Informatics (INDIN)*, pp. 500-505, 2013.

[77] L. Dürkop, J. Imtiaz, H. Trsek, L. Wisniewski, and J. Jasperneite, Using OPC-UA for the autoconfiguration of real-time Ethernet systems, in *2013 11th IEEE International Conference on Industrial Informatics (INDIN)*, pp. 248-253, 2013.

[78] OPCfoundation, OPC unified Architecture for ISA-95 Common Object Model Companion Specification, in *OPC foundation*, ed 2013.

[79] A. Gırbea, S. Nechifor, F. Sisak, and L. Perniu, Design and implementation of an OLE for process control unified architecture aggregating server for a group of flexible manufacturing systems, *IET Software,* vol. 5, pp. 406-414, 2011.

[80] T. Hannelius, M. Salmenpera, and S. Kuikka, Roadmap to adopting OPC UA, in *2008 6th IEEE International Conference on Industrial Informatics*, pp. 756-761, 2008.
[81] W. Mahnke, A. Gössling, M. Graube, and L. Urbas, Information modeling for middleware in automation, in *ETFA2011*, pp. 1-7, 2011.
[82] B. Dennis Brandl, P. H. DSInteroperability, and W. M. ABB, OPC UA and ISA 95, 2013.
[83] O. Palonen, Object-oriented implementation of OPC UA information models in Java, *Master's Thesis, Aalto University, School of Eletrical Engineering, Espoo,* 2010.
[84] K. Ni, N. Ramanathan, M. N. H. Chehade, L. Balzano, S. Nair, S. Zahedi, *et al.*, Sensor network data fault types, *ACM Trans. Sen. Netw.,* vol. 5, pp. 1-29, 2009.
[85] Y. Zhang, N. Meratnia, and P. Havinga, Outlier Detection Techniques for Wireless Sensor Networks: A Survey, *IEEE Communications Surveys & Tutorials,* vol. 12, pp. 159-170, 2010.
[86] S.-S. Wang, S.-C. Wang, and K.-Q. Yan, Reaching Trusted Byzantine Agreement in a Cluster-Based Wireless Sensor Network, *Wireless Personal Communications,* vol. 78, pp. 1079-1094, 2014.
[87] N. Lekhi and M. Mahajan, Outlier Reduction using Hybrid Approach in Data Mining, *International Journal of Modern Education and Computer Science,* vol. 7, p. 43, 2015.
[88] V. Hautamäki, S. Cherednichenko, I. Kärkkäinen, T. Kinnunen, and P. Fränti, Improving K-Means by Outlier Removal, in *Image Analysis: 14th Scandinavian Conference, SCIA 2005, Joensuu, Finland, June 19-22, 2005. Proceedings*, H. Kalviainen, J. Parkkinen, and A. Kaarna, Eds., ed Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 978-987, 2005.
[89] K. A. Yoon, O. S. Kwon, and D. H. Bae, An Approach to Outlier Detection of Software Measurement Data using the K-means Clustering Method, in *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*, pp. 443-445, 2007.
[90] G. Münz, S. Li, and G. Carle, Traffic anomaly detection using k-means clustering, in *GI/ITG Workshop MMBnet*, 2007.
[91] S. Chawla and A. Gionis, k-means–: A unified approach to clustering and outlier detection, in *Proceedings of the 2013 SIAM International Conference on Data Mining*, ed, pp. 189-197.
[92] L. Sasu, D. Puiu, and S. Nechifor, Fault recovery mechanism for smart city environments, in *2016 IEEE 20th Jubilee International Conference on Intelligent Engineering Systems (INES)*, pp. 57-62, 2016.
[93] S. Wang and J. Cui, Sensor-fault detection, diagnosis and estimation for centrifugal chiller systems using principal-component analysis method, *Applied Energy,* vol. 82, pp. 197-213, 11// 2005.
[94] Z. Shen and Q. Wang, Failure Detection, Isolation, and Recovery of Multifunctional Self-Validating Sensor, *IEEE Transactions on Instrumentation and Measurement,* vol. 61, pp. 3351-3362, 2012.
[95] T. H. Guo and J. Nurre, Sensor failure detection and recovery by

neural networks, in *IJCNN-91-Seattle International Joint Conference on Neural Networks*, pp. 221-226 vol.1, 1991.

[96] M. R. Napolitano, C. Neppach, V. Casdorph, S. Naylor, M. Innocenti, and G. Silvestri, Neural-network-based scheme for sensor failure detection, identification, and accommodation, *Journal of Guidance, Control, and Dynamics,* vol. 18, pp. 1280-1286, 1995/11/01 1995.

[97] G. Campa, M. L. Fravolini, M. Napolitano, and B. Seanor, Neural networks-based sensor validation for the flight control system of a B777 research model, in *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, pp. 412-417 vol.1, 2002.

[98] B. Kádár, W. Terkaj, and M. Sacco, Semantic Virtual Factory supporting interoperable modelling and evaluation of production systems, *CIRP Annals - Manufacturing Technology,* vol. 62, pp. 443-446, // 2013.

[99] Y. Hao, R. Karbowski, A. Shamsuzzoha, and P. Helo, Designing of Cloud-Based Virtual Factory Information System, in *Advances in Sustainable and Competitive Manufacturing Systems*, A. Azevedo, Ed., ed: Springer International Publishing, pp. 415-426, 2013.

[100] J. Fang, G. Q. Huang, and Z. Li, Event-driven multi-agent ubiquitous manufacturing execution platform for shop floor work-in-progress management, *International Journal of Production Research,* vol. 51, pp. 1168-1185, 2013/02/15 2013.

[101] J.-S. Yoon, S.-J. Shin, and S.-H. Suh, A conceptual framework for the ubiquitous factory, *International Journal of Production Research,* vol. 50, pp. 2174-2189, 2012/04/15 2012.

[102] N. Jufer, Performance Factory–A new approach of performance assessment for the Factory of the Future, *Estonian Journal of Engineering,* 2011.

[103] D. Yuniawan, T. Ito, and E. Mohamad, Characteristic Mapping Of Overall Equipment Effectiveness Elements Through Simulation And The Taguchi Method, *Journal of Human Capital Development (JHCD),* vol. 6, pp. 1-12, 2013.

[104] F. D. Carlo, M. A. Arleo, and M. Tucci, OEE Evaluation of a Paced Assembly Line through Different Calculation and Simulation Methods: A Case Study in the Pharmaceutical Environment, *International Journal of Engineering Business Management,* vol. 6, p. 27, 2014.

[105] J. A. Garza-Reyes, From measuring overall equipment effectiveness (OEE) to overall resource effectiveness (ORE), *Journal of Quality in Maintenance Engineering,* vol. 21, pp. 506-527, 2015.

초 록

스마트 공장은 생산 라인 전반에 걸쳐 기계 장비, 통신 기기, 무인 자동화 기기들에 사물인터넷이 적용되어 있는 곳으로 언제 어디서든 실시간으로 공장 상황을 모니터링 할 수 있고, 상황에 따라 통제 가능하도록 모든 생산 요소들이 연결되어 있는 생산 환경을 의미한다. 스마트 공장은 사물인터넷 기술을 통해 생산 현황 모니터링의 범위를 넓히고, 관련 데이터 취득 수준을 높이기 때문에 기존의 생산 환경보다 더 높은 생산성 및 유연성을 가질 수 있으며, 효율적인 자원 관리가 가능하다는 특징을 갖고 있다. 생산 공정 현황 모니터링은 성과 측정을 통해 가능한데, 새로운 형태의 사물인터넷들이 실시간으로 취득 가능할 수 있게 사물인터넷 데이터의 특성에 맞춘 새로운 성과 측정 모델이 적용되어야 할 필요가 생기게 되었다. 이에 본 연구는 생산 관리의 핵심 시스템인 제조실행 시스템의 국제 표준인 ISA-95와 성과 측정 지표 국제 표준인 ISO-22400 표준에 사물인터넷을 적용하여 새로운 성과 측정 모델을 제시하고자 한다. 또한 다양한 생산 정보시스템이 연계되어 있는 스마트 공장 환경에 맞추어 데이터 호환성을 높이기 위해 OPC-UA 표준을 활용하여 성과 측정 모델을 구현하도록 하였다.

생산 정보시스템에 많은 사물인터넷 기기들이 연결되면서 데이터 취득 시 혹은 데이터들이 연계되는 과정 중에 데이터 오류가 발생하는 문제가 빈번해졌다. 데이터 기반 성과 측정에서 데이터 오류는 성과 측정 결과물의 신뢰도를 저하시키며 관리자로 하여금 생산 현황을 정확하게 인지하는 데에 방해가 되는 요소이다. 본 연구는 사물인터넷 데이터 오류에 대해 모니터링하고, 오류를 탐지하였을 경우 이를 추론 기법을 통해 실제 현황에 맞춘 데이터로 변환 시켜 성과 측정 결과물에 정확도를 높이는 모델을 제시하고자 한다. 또한 시뮬레이션을 통해 데이터 오류가 성과 측정 결과에 미치는 영향력을 파악하고, 제시한 오류 탐지 및 완화 알고리즘의 성능에 대해 검증하고자 한다.

주요어 : Performance measurement; ISA-95; Internet of Things; OPC-UA; Fault tolerance; Data anomaly analysis
학 번 : 2013-30318

알 림 : 해당 학위 논문은 황규선 저자가 주 저자로 있는 아래의 게재된 논문을 기반으로 작성되었음

[1] G. Hwang, S. Han, S. Jun, and J. Park, Operational Performance Metrics in Manufacturing Process: Based on SCOR Model and RFID Technology, *International Journal of Innovation, Management and Technology*, vol. 5, 2014.

[2] G. Hwang, J. Lee, J. Park, and T.-W. Chang, Developing performance measurement system for Internet of Things and smart factory environment, *International Journal of Production Research*, vol. 55, pp. 2590-2602, 2017.

상위 논문을 제외하고, 학위 논문 가이드라인을 준수하여 관련 연구들을 인용하였음