Cybersecurity of industrial connected equipment : modeling, detection and temporal performance in the presence of intrusions of cyber-physical systems of the factory 4.0

Salwa Alem

Cybersecurity of industrial connected equipment : modeling, detection and temporal performance in the presence of intrusions of cyber-physical systems of the factory 4.0

Salwa Alem

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

Université Bretagne Sud

Thèse de doctorat de

L'UNIVERSITÉ DE BRETAGNE SUD
Comue Université Bretagne Loire

École Doctorale N° 601
Mathématiques et Sciences et Technologies
de l'Information et de la Communication
Spécialité : Télécommunications

Par
«Salwa ALEM »

« Cybersecurity of industrial connected equipment : Modeling, detection and temporal performance in the presence of intrusions of cyber-physical systems of the factory 4.0»

Thèse présentée et soutenue à « Lorient », le « 25 Mai 2021 »
Unité de recherche : Lab-STICC – CNRS UMR 6285
Thèse N° : 598

Rapporteurs avant soutenance :

Stéphane MOCANU	Maître de conférences et HDR, Institut Polytechnique de Grenoble
Mireille BAYART MERCHEZ	Professeur et HDR, Ecole polytechnique universitaire de Lille

Composition du Jury :

Président :	Vincent NICOMETTE	Professeur et HDR, INSA de Toulouse
Examinateurs :	Éric ZAMAI	Professeur et HDR, INSA de Lyon
Dir. de thèse :	Éric MARTIN	Professeur des universités, ENSIBS
Co-dir. de thèse :	Laurent NANA	Professeur des universités, Université de Bretagne Occidentale (UBO)
Enc. de thèse :	David ESPES	Maitre des conférences, Université de Bretagne Occidentale (UBO)
Co-enc. de thèse :	Florent DE LAMOTTE	Maitre des conférences, Université de Bretagne Sud (UBS)

Invité(s) :

Valérie VIET TRIEM TONG	HDR, CENTRALESUPELEC

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

Université Bretagne Sud

Title: Cybersecurity of industrial systems: intrusions detection system (IDS)

Keywords:

Industry 4.0, Cyber Physical System (CPS), Intrusion Detection System (IDS), ISA-95, Artificial Intelligence (AI)

Abstract: Cybercrime is eased by the emergence of the fourth industrial revolution, industry 4.0. The fourth industrial revolution is characterized by the convergence of Information Technology (IT) and Operation Technology (OT) worlds', the huge generated data, the use of Cloud as new storage means and the limitation of the security mechanisms.

All these factors increase the risk of cyber attacks in industry. Fortunately, several solutions exist to secure the industry and its equipment. Among these are firewalls, anti-virus, auditing process and IDS. Each of these securing mechanisms has a specific role such as detecting and removing malware, preventing unauthorized access, or detecting intrusions by IDS. The latter gives visibility of a system activities which in turn allows a timely detection and response to suspicious events. In literature, exist two types of IDS approaches: signatures-based and anomaly-based IDS. This last one is more efficient to detect advanced and zero-day attacks and it is composed of two other IDS types: specification-based IDS and behavioral-based IDS (see Section II). Each one of them has its own advantages and limits.

In industrial intrusion detection field, the main problematic is to distinguish between industrial process dysfunction from a real intrusion. In this thesis, the proposed approach deals with this specific point. Consequently, hybridization of two types of anomaly-based IDS remains the most efficient method to distinguish between process dysfunction from a cyber-attack in order to perform a high performance of intrusion detection. Therefore, this thesis propose an efficient IDS composed of a specification-based IDS and a behavioral-based IDS. The specification-based IDS is based on an industrial standard called ISA-95 allowing the detection of the process anomalies and the behavioral-based IDS is based on the network traffic analysis' using supervised neural network algorithm.

Titre : Cybersécurité des équipements connectés industriels : systèmes de détection d'intrusions

Mot clés : Industrie 4.0, Systèmes cyber-physique, Système de détection d'intrusions, ISA-95, Intelligence artificielle

Résumé : De nos jours, les systèmes de contrôle industriel (ICS) existent dans l'ensemble des secteurs industriels comme le secteur du conditionnement, l'industrie des produits chimiques, la construction, l'automobile, l'industrie électronique ... mais aussi dans les secteurs industriels vitaux comme l'énergie, la santé, l'armée et l'alimentation. Par conséquent, la suspension ou l'arrêt de ces systèmes pourrait être coûteux pour l'industriel et entraîner des dommages conséquents. Aujourd'hui, sécuriser un tel équipement devient plus que nécessaire.

Au cours de la dernière décennie, l'industrie est devenue la cible des attaquants et a été victime de plusieurs attaques à commencer par Stuxnet, Black Energy, WannaCry. Cette vague d'attaques a été suivie de plusieurs attaques de ransomwares en 2020 lors de la pandémie de virus corona, notamment avec l'augmentation des travailleurs à distance et un manque de sécurité Ce phénomène de cybercriminalité est favorisé avec l'émergence de l'industrie 4.0. Cette 4ème révolution industrielle est caractérisée par la convergence des mondes des technologies de l'information et des technologies d'exploitation, les énormes données générées, l'utilisation du Cloud comme nouveau moyen de stockage et la limitation des mécanismes de sécurisation. Toutes ces raisons augmentent le risque de cyberattaques dans l'industrie.

Heureusement, il existe plusieurs solutions pour sécuriser l'industrie et ses équipements. Parmi ces mécanismes, nous mentionnons les pare-feu, l'antivirus, le processus d'audit et l'IDS. Chacun de ces mécanismes de sécurisation a un rôle spécifique tel que la détection et la suppression des logiciels malveillants, la prévention des accès non autorisés ou la détection des intrusions par IDS. Ces derniers donnent une visibilité sur les activités du système qui permet une détection et une réponse à tout événement suspect en temps opportun. Deux types d'approches d'IDS existent dans la littérature qui sont les IDS basés sur les signatures et les IDS par anomalies

Dans le domaine de la détection d'intrusions industrielles, la principale problématique consiste à distinguer un dysfonctionnement du procédé industriel d'une véritable intrusion. Dans cette thèse, l'approche proposée traite ce point spécifique. Par conséquent, l'hybridation de deux types d'IDS par anomalies reste la méthode la plus efficace pour distinguer un dysfonctionnement d'un processus d'une cyber-attaque et réaliser une haute performance en terme de détection d'intrusion. Par conséquent, cette thèse propose un IDS efficace composé d'un IDS par spécifications basé sur une norme industrielle appelée ISA95 permettant la détection des anomalies de processus et d'un IDS comportemental basé sur l'analyse du trafic réseau à l'aide d'un algorithme de réseau neuronal supervisé.

Acknowledgement

I would like to express my thanks of gratitude to all those who have contributed to the success of my thesis and who have helped me in writing this manuscript.

Foremost, I would like to thank my thesis director, Mr. Eric Martin, for giving me this amazing opportunity to achieve my thesis work in excellent work conditions ; I would also to thank him for his energy and confidence which were driving forces for me. His experience and his high standards have greatly stimulated me.

I would like to express to my supervisor Mr. David Espes, for his patience, availability, permanent good humor, and above all his wise advice, which helped to nourish my reflection and to carry out this thesis. I got a great pleasure to work with him.

I am highly indebted to co-thesis supervisor, Mr. Laurent Nana, and my second supervisor Mr. Florent De Lamotte, for their attention, wise advices, and their listening throughout my work, which was a major key for the success of this thesis.

My appreciations go also to the jury members for honoring me with their agreement to evaluate and validate this thesis. A special thanks to Mrs Valérie Viet Triem Tong, and Mr. Eric Zamai, for accepting to be the reporters of my work.

I thank Mrs. Fatima Ezzahra Ezzaouit, L. Sussan and Sophie Dudley for reading and correcting my manuscript, their notes and writing tips have been priceless to me.

This work could not be done without my colleagues, Thomas Toublanc, Iehann Eveno, Fanny Guennoc and Samia Benferhat , I could not thank them enough for their help, advices, the good mood and their compassion.

Last but not the least, I would like to express my deepest thanks to those who are dearest to me and whom I have somehow neglected lately to complete this thesis. Their

attentions and encouragements have been my lighthouse throughout these years. As this journey comes to its end, I am highly indebted to my parents, Alem Abderrahman and Azza Hafida, for their moral support and their unshakeable confidence in my choices. I am also grateful to my brothers and my sisters, in particular Ayoub Alem ; for his help, his technical advice and his availability.
A special thanks to my partner for his support and patience, and my children who have been a source of love and energy for me to move forward.

Table of Contents

List of figures

List of Tables

Acronyms

6BR IPv6 border router

6LoWPAN IPv6 LoW Power wireless Area Networks

AAKR Auto Associative Kernel Regression

ACCM Ant Colony Clustering Model

ANN Artificial Neural Network

ANSSI Agence Nationale pour la Sécurité des Systèmes d'Information

AOET Actual Order Execution Time

API Application Programming Interface

APT Actual Production Time

ARP Address Resolution Protocol

ATT Actual Transportation Time

B2MML Business to manufacturing markup language

BACnet Building Automation and Control protocol

BGD Batch Gradient Descent

BIANO-IDS BI ANOmaly Intrusion Detection System

BR Border Router

CIA Confidentiality Integrity Availability

CIM Computer-Integrated Manufacturing

CNN Convolutional Neural Networks

CPS Cyber Physical System

CS Critical State

CSTH Continuous System Telemetry Harness

DDoS Distributed Denial of Service

DMS Decision Making System

DMZ DeMilitarized Zone

DNP3 Distributed Network Protocol

DPI Deep Packet Inspection

DT Decision Trees

EPO Executed Production Order

ERP Entreprise Resource Planning

FTP File Transfert Protocole

GUI Graphical User Interface

HIDS Host Intrusion Detection Systems

HSE Health, Safety and Environment

HTTP Hypertext Transfer Protocol

ICMP Internet Control Message Protocol

ICS Industrial Control System

IDE Integrated Development Environment

IDMEF Intrusion Detection Message Ex-change Format

IDPM Intrusion Detection and Prevention Mechanism

IDS Intrusion Detection System

IEC International Electrotechnical Commission

IIoT Industrial Internet of Thing

IMA Illegal Memory Accesses

ISML Industrial State Modelling Language

IPS Intrusion Prevention System

ISO International Standards Organisation

ISSP Information System Security Policy

IT Information Technology

KNN K-Nearest Neighbors

KPI Key Performance Indicators

LR Linear Regression

LSTM Long Short-Term Memory

MES Manufacturing Execution System

MITM Man In The Middle

MLP Multi layers Perceptron

MN Monitoring Nodes

MQTT Message Queuing Telemetry Transport

NB Naive Bayes

NIDS Network Intrusion Detection Systems

NIST National Institute of Standards and Technology

OCSVM One-Class Support Vector Machine

OPC-UA Open Platform Communications Unified Architecture

OS Operating System

OSI Open Systems Interconnection

OT Operation Technology

PEL Perceptual Executive Layer

PEM Process and Equipment Monitoring

PLC Programmable Logic Controler

PNMN-AKF Process Noise and Measurement Noise-Adaptive Kalman Filter

PO Production Order

PPO Planned Production Order

RFID Radio Frequency IDentification

RBM Restricted Boltzmann machine

RNN Random Neural Network or Recurrent Neural Network

RPL Routing Protocol for Low-Power and Lossy Networks

RSSI Received Signal Strength Intensity

RTT Round-Trip Time

RTU Remote Terminal Unit

SCADA Supervisory Control And Data Acquisition

SFC Sequential Function Charts

CPS Cyber Physical System

SPRT Sequential Probability Ratio Test

SVM Support Vector Machine

TLS Transport Layer Security

TTL Time to Live

UDP User Datagram Protocol

VPN Virtual Private Network

WO Work Order

XML Extended Markup Language

Chapter 1

Introduction

1.1 Cybersecurity in industry 4.0 : Overview

Today, we are facing a new industrial generation commonly known as the 4^{th} industrial generation or industry 4.0. It is based on increasing the speed of information processing, memory capacity, and also on a huge increasing of data sharing across multiple systems and participants in the manufacturing process. Industry 4.0 takes benefits from data exchange's results.

This industrial revolution was preceded by three other industrial generations. The first one began in the 18^{th} century and was characterised by the widespread adoption of steam power and the mechanisation of production.
In the second half of the 18^{th} century, the second industrial revolution was brought about by the introduction of mechanics, telegraph networks, the transport development and electricity. This latter was used as a master piece in modernisation of large-scale production mechanisms and productivity increasing. The different devices were then interconnected through primitive networks such as bus or serial networks.
In the second half of the 20^{th} century, the third industrial revolution appeared due to the evolution of the electronics, telecommunications, computers and audiovisual fields (see Figure 1.1). During this generation, ICS were isolated and supposed to be isolated and protected from the outside world until 2010 when hackers have targeted a nuclear program of Iran with Stuxnet worm causing consequent damage. Other attacks have followed Stuxnet such as Shamoon in 2012 targeting a Middle Eastern oil company in Saudi Arabia and Dark Seoul in 2013 damaging thousands of computers in South Korean media and financial companies. These attacks have shown that isolation is not the appropriate solution to be protected against attacks.
The 4^{th} industrial revolution of automation has appeared relying on the massive introduction of cyber-physical systems (CPS), which can be defined as complex embedded systems designed to interact with their environment continuously through the combina-

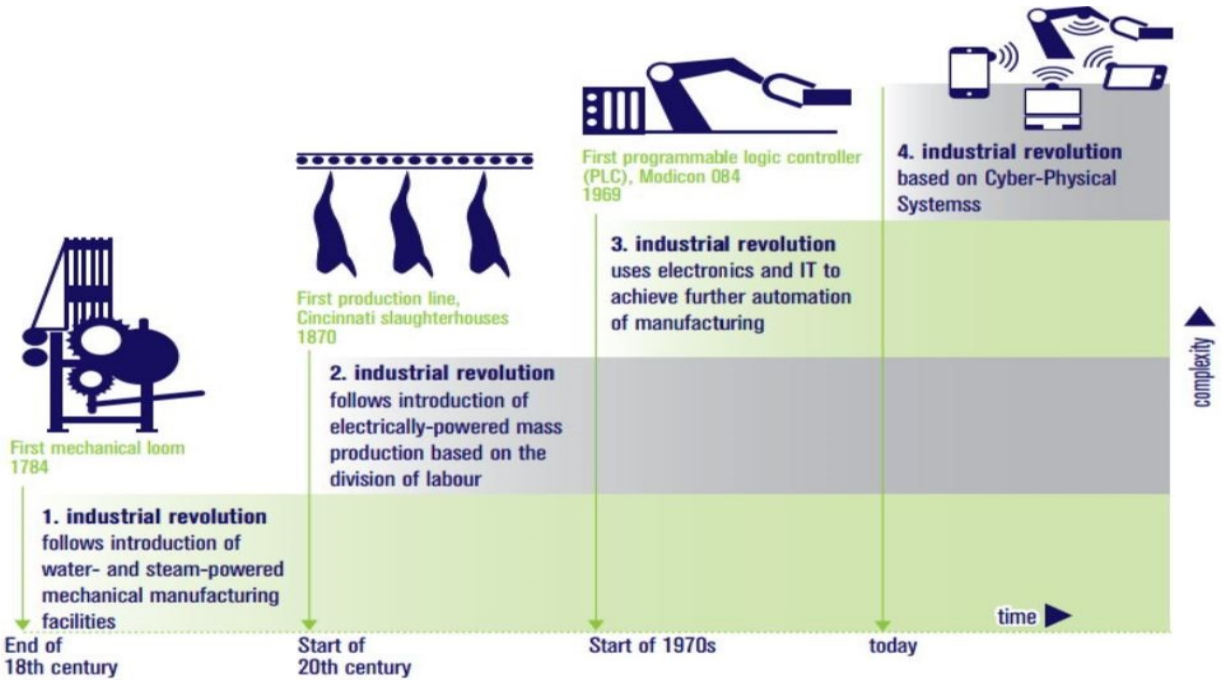


Figure 1.1 – The fourth industrial generations

tion of physical, computers and communication elements. This smart industry is based on many pillars such as Machine to Machine (M2M), Big data, Cloud computing and Information Technology (IT)/Operations Technology (OT) worlds' convergence.
Today with the advent of This 4^{th} generation, the industrial world and the outside one border is removed especially with the emergence of the industrial internet in factories. The opening of industrial systems to external world and the democratization of IP make all the attacks that were carried out in the past in IT world possible to be replayed in OT world. The risk of intrusions into industries has increased and the vulnerability of industrial control systems has grown. With the emergence of this new industrial generation, the number of attacks has been multiplied which could also cause a fatal damage on Health, Safety and Environment (HSE).Hackers hit more than ever industry with more sophisticated attacks such as German steel factory cyberattack in 2014 causing massive material damage, BlackEnergy and Killdisk that have targeted an Ukrainian power plant in 2015 depriving hundreds of people of power and NotPetya ransomware causing $10 billions damage in 2017 in the worldwide.

Today, the industrial control system (ICS) networks face cyber threats from a wide range of actors-state-sponsored hackers, terrorist groups, hacktivists, professional criminals, and disgruntled employees. The introduction of disruptive technologies such as the Industrial

Internet of Things (IIoT) and as well as the factory 4.0 pillars brought more complexity, by widening the risk area of the fragile ICS networks to attacks. This kind of equipment (e.g. IIoT, M2M,...) is connected to people and to machines through internet. Therefore, another door is opened for hackers. Some security experts even claim that the IoT (Internet Of Things) means "Internet of Threats" [Meany 2017].
In addition, in this fourth industrial generation, most data is deported and is no longer stored in premises' storage media such as local hard drive but in the Cloud. Unfortunately, today OT world does not have all the necessary tools to protect its equipment and its exchanged data.
Nowadays, in industry, IEC-62443-3-2 standard introduces “zone” and “conduits” concepts. “zone” designs a set of physical or logical assets with the same "criticality" which must to respect and meet common security requirements. The physical or logical border between these zones has to be clearly determined. The connections between these zones are called conduits. Due to this industrial structure, the security mechanisms used in the industry are mainly segregation mechanisms such as firewalls or Virtual Private Network (VPN).
To detect intrusions, industries use either Host Intrusion Detection Systems (HIDS) which analyse and monitor logs and audits of an operating system or a machine [Zhengbing, Zhitang, and Junqi 2008] such as anti-virus or they implement Network Intrusion Detection Systems (NIDS) that capture, manage and analyse packets from the network. HIDS and NIDS handle only packets for a specific host or a set of machines connected to network [Kazienko and Dorosz 2004]. NIDS could also be deployed in an active network element such as router.

The main issue with this securing mechanism is the fact that in industry, IDS are deployed in the IT workstations and not in the industrial devices such as Programmable Logic Controller (PLC) which limits the intrusions detection.

1.2 Industrial cybersecurity problematic

1.2.1 Reverse security priority in the industrial world

The cybersecurity field is based on three CIA triads which are confidentiality, integrity and availability [Ross 2020] [Lee and Jang 2009]. These three security properties are

defined below:

- Confidentiality means that the access to data or equipment has to be restricted. Equipment or information have to be accessible only for authorized persons [Kumar et al. 2015]. The violation of this security requirement could be exploited by hackers who can disrupt services and steal data [Yin et al. 2002].

- Integrity implies that data remains accurate and consistent throughout its life cycle. Users' requests to services have to be made correctly. The violation of this characteristic could alter, lose and compromise data [Kumar et al. 2015]. Any alteration or modification in the data is made by authorised persons and according to a defined policy.

- Availability signifies that data or equipment have to be accessible and services have to be provided without interruption [Yin et al. 2002]. This securing characteristic depends sometimes on some conditions related to time slots and the people authorised to use them.

To propose an efficient securing mechanism, these three requirements have to be fully considered. The priority and the significance of these requirements differ according to the context (application field and the data inside the information system) [P. Williams 2001]. Between IT and OT worlds, the priority is reversed. Confidentiality is the priority number one in IT world. Ensuring confidentiality in a company allows not only to have a good image and high level of credibility, but also to assume legal responsibility to avoid the leak and theft of sensitive customer data. The violation of this security property could expose a company to consequent legal proceedings. In the IT world, ensuring confidentiality is not an option but an obligation.
While confidentiality is the primary concern for IT engineers, availability or integrity seems to be the top priority for ICS systems according to activity sector. Industry kind is decisive to prioritize either availability or integrity. For instance, availability of material and data is of a major importance in automotive sector instead of integrity and confidentiality. However, pharmaceuticals industries are concerned that the proportions of raw materials used in their products may be disrupted which could compromise the quality of their products and their notoriety.

1.2.2 Real time constraint

IT and OT control systems have different requirements. Latency and delay are not allowed in industry and if they occur, it could be expensive for manufacturers. Unlike IT equipment that requires high throughput and allows a response delay, ICS are time-critical equipment and in some particular cases, a real-time response is required [Stouffer et al. 2015].
In Industry 4.0, real-time aspect is very important. This notion encompasses other concepts which are presented below :

- Real-time alerts : Alerts can be raised, machines can be proactively maintained and operators can do remote maintenance [i-scoop 2017]. Today, monitoring and predictive diagnostic and maintenance have became possible.
- Real-time capability : This characteristic consists of being able to acquire data, transmit and evaluate it in a short time frame [Basler 2020]. This specificity could be major for applications where speed is required. All advanced analytic, both IIoT and smart industrial production participate all to the development of real-time capability's notion.
- Real-time information : Thanks to the acquisition and analysis of real-time production information, today we are able to optimize a production process and reduce unnecessary tasks. This real-time data could also help predicting dysfunction problems and plan proactive maintenance. It also contributes to a good decision-making about particular situations. Therefore, it allows an efficient production in the smart factory.

1.2.3 Convergence IT/OT

Industry 4.0 is based on the connection of the numerical world to the physical worlds through factory 4.0 that contains cyber-Physical Systems (CPS). Before the advent of the 4^{th} industrial generation, Industrial Control Systems (ICS) were isolated and more protected from the outside world. Today that this border is opened with the emergence of the industrial internet in the factories, the opening of industrial systems to external world and the democratization of IP, the convergence of Information Technologies (IT) and Operation Technologies (OT) becomes a reality. This convergence allows to industries the automation of the entire life cycle from design step to production along with several

challenges.
OT operators are well aware of physical security threats and have implemented safety measures in industrial systems for decades. However, they face today threats that are potentially beyond their control. Since machines and control systems are no longer close and isolated, the hacking threat is introduced, with the endangerment of employees such as through overheating or cancellation of emergency stops.
To face these threats, OT world has to work with IT world, but some concerns arise. IT engineers generally have little experience with industrial systems and their traditional security solutions are often incompatible with existing control systems. In [19] author says that OT and IT tend to use different approaches to problem-solving. IT implements solutions using a top-down approach while in OT professionals are using a bottom up approach solutions, starting from the individual components to build a more complex system.
In addition, the attacks that were played in the past in IT system could be replayed today in ICS. Furthermore, industrial concurrency have increased during this last decade. Therefore, data violation has to be avoid and confidentiality of the transmitted data must be guaranteed with the securing mechanisms proposed by IT engineers.

1.2.4 Equipment and protocols heterogeneity

Another challenge is observed in industry 4.0 which is the heterogeneous equipment. While IT world is used to frequent and regular patches and software updates, industrial equipment are too obsolete and outdated tending to take a more systemic approach which make their configuration incompatible with the standard security offers of IT teams. Furthermore, in industry we find both wired and wireless equipment. Some of them are still based on the traditional protocol such as IP/TCP Modbus/TCP and other are using some new protocols related to Industrial Internet Of Things (IIoT) such as OPC-UA, MQTT, Lora, zigBee, Sigfox...
In this context, industry has to put in place a good strategy for an adaptive combination between equipment and interfaces of all of these components [Zhou, Taigang Liu, and Lifeng Zhou 2015].
This heterogeneity raises more challenges to industry 4.0. These challenges consist of equipment connectivity management, analysis of a huge heterogeneous data sources but above all, the main challenge is the proposal of an adaptive cybersecurity strategy to face the cyber threats.

IT and OT convergence complicates this task since some proposed securing techniques are suitable for some IT equipment such as encryption but this technique generates a delay which is not allowed in OT environment. The interoperability of the proposed securing mechanisms has to be guaranteed.

1.3 Thesis motivations and positioning

1.3.1 Thesis Motivations

Industries need more than ever a strong security strategy based on IEC-62443-3-2 standard which recommends defense in depth. Within this context of defense in depth, the central information systems security department has drawn up a guide for developing an information system security policy (ISSP). This guide contains 160 measures to build an efficient and a robust ISSP to deal with vulnerabilities and the risks of attacks or intrusions. These principles are involved in 16 areas of information systems security (ISS). This guide covers a large scope to protect information system and it has inspired us to determine our thesis perimeter. Among the 160 measures proposed by this guide, this thesis focus on those mentioned below:

- Establishing an intrusion detection system
- Mechanisms of logging intrusions or fraudulent use
- Implementation of an alerts system
- Predicting reflex reactions to emergency situations
- Reduction of the vulnerabilities

Focusing on this guide and on the existing security mechanisms in industry, we have found that there are a several solutions to secure the industry and its equipment. Among these mechanisms, we mention firewalls, anti-virus, auditing process and IDS. Each of these securing mechanisms has a specific role such as detecting and removing malware, preventing unauthorized access or detecting intrusions by IDS. These latter give a system activities visibility which allows a timely detection and response to any suspicious events [Arshad et al. 2020]. Two kinds of IDS approaches exist in literature which are signatures-based and anomaly-based IDS. This last one is more efficient to detect advanced and zero-day attacks and it is composed of two other IDS types which are specifications-based IDS and

behavioral-based IDS (see below). Each of these IDS has advantages and limits.
In industrial intrusion detection field, the main problematic consists in how to distinguish a dysfunction of the industrial process from a real intrusion. Thanks to the hybridization of two anomaly-based IDS, this thesis proposes an approach which allows the distinction of a dysfunction process anomalies from the real intrusions and classify these latter into temporal or sequential. Therefore, this thesis proposes an efficient IDS composed of a specifications-based IDS and a behavioral-based IDS. Our specifications-based IDS is focused on Manufacturing Executive System (MES) exchanging data and it is based on an industrial standard called ISA95 allowing the detection of the process anomalies and the behavioral-based IDS is based on the network traffic analysis using supervised neural network algorithm.

1.3.2 Thesis positioning

1.3.2.1 IEC-62443 standard

IEC 62443 focuses on the security of the industrial control system (ICS). The main purpose of this standard is to ensure that all industrial suppliers, operators, integrators or engineers opt for an effective method to ensure environment security and the personnel safety. It must also ensure the availability, efficiency and quality of the production of the ICS [Mickael 2019].
This standard is composed of 4 parts which are "General", "Management System", "Industrial IT Security" and "Embedded Security Components" (See 1.3).

The third part proposes many security strategies to protect ICS. In industry, after implementing a strong network architecture, a security architecture has to be established. This architecture includes all control means and their placement inside network and hosts in the aim of putting in place layers of security-Defense in Depth [us-cert 2016]. This security strategy consists of setting up a series of defense levels based on the intrinsic characteristics of installation, material, organizational and human measures as well as the procedures intended to prevent accidents.
Among the security techniques proposed by the defense-in-depth, there are zones and conducts as it is explained above and the implementation of an IDS in each zone. IDS represents an integral part of defense-in-depth strategy since IDS is automated mechanism to monitor a network or a system and responds to the unexpected events.

Figure 1.2 – IEC 62443 Series

Based on this standard, this thesis proposes a hybrid IDS composed of two anomaly-based IDS kinds: Specifications-based IDS and Behavioral-based IDS.

1.3.2.2 Towards anomaly-based intrusion detection systems

Anomalies are part of the industrial cycle. In industry, the risk of anomaly is inherent to any manufacturing process, but efficiency must come first. It is necessary to acquire the capacity to protect against any break in production continuity by anticipating possible failures, detecting anomalies sufficiently upstream so as avoid the damages that could be costly, especially in vital sectors. These anomalies could be as well intentional as unintentional origin. The border between these both could be very thin.
Currently, in industrial intrusion detection system field, the main difficulty is how to distinguish dysfunction anomalies from real intrusions to reduce the false positive alerts. For this purpose, we combine in the proposed approach two anomaly-based intrusion detection systems:

- Specification-based IDS: it is one of the two kinds of the anomalies-based IDS. A specifications-based IDS builds a system reference model from its properties which constructs patterns and rules [Caselli et al. 2016]. In the proposed specifications-based IDS, MESA model extracted from ISA-95 standard is used to define these

specifications. This model is described in details in the next chapter. It takes as input, 13 rules that we identified from MESA model (see chapter 3). To check the nominal function of our production line, a rules set is defined. The IDS raises an alert as output if one of these rules is not respected and the error is reported in the IDS logs for more investigations by the operator.

- Behavioral-based IDS: this one represents the second type of anomalies-based IDS. It defines a behavior reference model from features and observes their changes. Any activity which differs from this defined reference behavior is considered as abnormal [Salwa Alem et al. 2019] [Hodo et al. 2016]. Differently from specification-based IDS, behavior of features is derived by a learning phase.
  This IDS takes as input the network traffic features that we pre-processed and labeled. It is an intelligent IDS based on neural network model. From an extracted dataset, we train a neural network model to recognize normal activity from the malicious one. The IDS pushes an alert if the activity deviates from the reference defined model. The alerts are listed in the IDS logs to be used by the third module which is responsible of a decision making regarding the nature of the detected anomaly.

1.3.2.3 Positioning: Manufacturing Executive System (MES)

According to Computer Integrated Manufacturing (CIM), factory is composed of 5 levels as shown in the left side of 1.3:

- Enterprise Resource Planning (ERP): A software package that allows the management of all operational processes in a factory by integrating several management functions [choisirmonerp 2020]. It is the highest factory's level.

- Manufacturing Executive System (MES): A software package which fills the gap between ERP and supervisory level [S. Alem et al. 2019]. It is responsible of the orders planning and scheduling, resource management, quality and maintenance controls and inventory management [Choi and Kim 2002]. It centralizes all information concerning production, maintenance, inventory and product quality.

- Process control level or Supervisory control and data acquisition (SCADA): A system which allows the monitoring and controlling of all sensors and actuators using

a monitoring system. This latter could be either a PC or Programmable Logic Controller (PLC) [Igure, Laughter, and R. D. Williams 2006].

- Control level or Programmable Logic Controller (PLC): An industrial computer which is responsible of a real-time measurement and control of industrial processes [Van Aubel et al. 2017].
- Ground field level: It contains sensors, actuators, robots and collaborative robots (cobots).

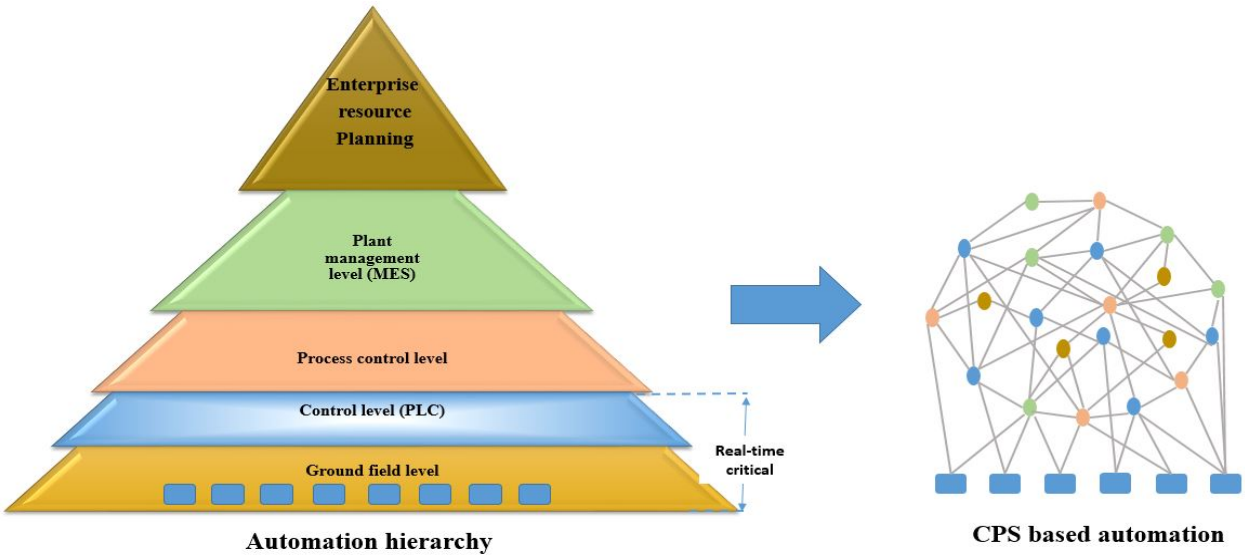


Figure 1.3 – Conventional industrial hierarchy Vs. industrial hierarchy 4.0

Since the emergence of industry of the future, the smart manufacturing ecosystem has been evolved to become more agile and fully connected (See the right side of 1.3). All the manufacturing functions in CIM pyramid can be virtualized and hosted as services, except those time-critical that have to stay at the shop floor level. In these both hierarchies, MES remains present since it represents the main industrial information system that has to be protected against industrial cyber-attacks due to its important role which consists of filling the gap between IT and OT worlds.
During the study and analysis phase of the state of the art (see chapter 2), we observed that the whole proposed intrusion detection systems are targeting either PLC or SCADA levels. None of the proposed approach is focused on the MES level. Thanks to the MESA standardization applied to MES, it possible to define an interesting behavioral model for

the anomalies detection : this model is linked to the manufacturing field and standardization makes it suitable and adaptive for all MES systems, new or old ones. Therefore, This standardization makes possible to reduce the efforts to adapt the detection.

For all these presented reasons, we have specifically targeted this industrial level by proposing a hybrid intrusion detection approach that we expose all its modules through the different chapters of this dissertation whose organization is presented in the next Section.

1.4 Organisation of the dissertation

This dissertation is organized into 5 chapters:

- Chapter 2 - Existing industrial IDS: This chapter gives a global view of the existing research works related to industrial IDS. They are classified into four IDS kinds which are signature-based IDS, specifications based IDS, behavioral based IDS and hybrid IDS. This chapter draws up also a state of the art regarding the existing datasets used as a basis in behavioral based IDS approaches.

- Chapter 3 - behavioral-based IDS: It is the first component of our global approach. In this part, a behavioral-based IDS is exposed by using a neural network algorithm. This IDS works in addition with the below IDS. Thanks to the MES database and the MESA model, today it is possible to discriminate industrial dysfunction from a real intrusions and classify these latter into several kinds of intrusions. It is presented, as well as its performance evaluation using a new industrial dataset was captured and built during this thesis. The used dataset is built from the network traffic features related to Modbus protocol and containing 133 features. This chapter ends by a discussion on the results of the performance evaluation of the IDS, followed by conclusions.

- Chapter 4 - Specification-based IDS: This chapter presents the second component of our proposed approach which is a specification-based IDS. It details the principle and the basis of this IDS which is MESA model from the industrial standard called ISA-95. Specifications based IDS framework is presented at the end of this chapter.

- Chapter 5- Global approach of BIANO-IDS: This chapter presents the global approach of the proposed IDS in addition to its third module which is a Decision

Making System: DMS. This latter makes a decision regarding the nature of the detected anomalies. The results obtained are presented and discussed.

- Chapter 6 - Conclusions and perspectives: This chapter ends this dissertation by a summary of the work performed in this thesis, its limitations and future works.

Chapter 2

State of the art of industrial IDS and datasets

2.1 Intrusion Detection System (IDS)

Intrusion Detection Systems (IDS) is security detection mechanisms put in place to monitor network traffic, suspicious activity, policy violations and to alert the system administrator when such activities or violations are detected. They have been in existence since the 1980s with Anderson research [29]. IDS could use either the machine log and are called Host Intrusion Detection System (HIDS). In the second case, IDS use network traffic lofs and are called Network Intrusion Detection System (NIDS). According to the approach used, two IDS kinds are distinguished: signature-based IDS and anomaly-based IDS:

- Signature-based IDS: also called misuse IDS, they are based on a set of attack descriptions or signatures [11]. It consists of defining attack scenarios and looking for traces of these scenarios. Signature-based IDSs are the most commonly used in industry.

- Anomaly-based IDS is composed of two types:

  - Behavioral-based IDS: These IDS are related to models of normal behavior of a computer system [12]. The principle of this approach is to define a behavior reference representing the normal behavior, then any activity which deviates from this reference behavior is considered as an intrusion.

  - Specification-based IDS: This type builds a system reference model from its properties forming patterns and rules. These behavior features are extracted directly from documentation [Caselli et al. 2016].

In the following sections, both conventional industrial IDS (see Section 2.2) and industrial IDS for IoT (see Section 2.3) are presented. In addition, the conventional ones are classified into signatures-based and anomaly-based IDS.

2.2 Conventional industrial IDS

In this section, a state of the art related to the exiting IDS used in the traditional industry is established. These IDS target either SCADA or PLC or the automation ground level. These existing research works are classified into either signature-based IDS (see Sub-section 2.2.1) or anomaly-based IDS (see Sub-section 2.2.2).

2.2.1 Signature-based IDS

The first signature-based IDS were developed and implemented a long time ago in [Hochberg et al. 1993][Anderson, Frivold, and Valdes 1995][Paxson 1998]. Since their emergence, several researchers have enhanced and improved their detection techniques by proposing diverse and varied approaches: distributed, innovative or basic ones. A new distributed approach based on Mobile Agent (MA) is exposed in [Barika, Hadjar, and El-Kadhi 2009]. Their distributed IDS is composed of four agents: Sniffer, Filter, Analyser and Decision Agents. The Sniffer Agent intercepts and logs the real-time network events; the Filter Agent which collects, treats and categorises the detected events; the Analyser Agent which processes then the filtered event by using a pattern matching method to check if there are similarities between the filtered and malicious packets and; finally, the Decision Agent which uses the administrator knowledge for a deep detection. The authors simulated four attack kinds (Probe, R2L, U2R and DOS) and captured a real traffic. They evaluated their approach by measuring the detection delay, false positives, and detection rate. Their performance seemed good.
Another innovative approach is detailed in [Fovino et al. 2010]. In this research work, the authors propose an innovative signature-based IDS which combines a conventional detection method using known attack signatures and the system state checking methods. After analysing the packets in the known signatures database, the IDS keeps in its memory a system state (a numerical representation of the system). Therefore, an alert is raised when the system reaches a critical state. For their implementation, the authors put in place a set of signatures for the DNP3 and Modbus protocols, and devices states. According to Evaluation results showed their IDS detects the simulated attacks with a high accuracy.
In [Diallo and Feuillet 2014], the authors provide a set of rules to be integrated in Suricata signature-based IDS. Their approach targets SCADA level and enriches the IDS Suricata rules with those specific to the Modbus protocol. The performance tests carried out show

the feasibility of detection probes with inexpensive industrial equipment. The criticisms made against of this approach is that the fact that a failure cannot be distinguished from a real intrusion. In addition, new attacks whose signature does not match with none of the proposed rules cannot be detected.
Some other researchers have chosen another signature-based IDS approach such as in [Sicard, Zamai, and Flaus 2018]. The proposed approach is composed of two modules. The first one is a filter, and the second one an IDS. To detect intrusions, the latter uses the distance notion which leads an Industrial Control System (ICS) to a critical state. Therefore, by controlling this distance, an industrial system is prevented leading a system to this critical state.

2.2.2 Anomaly-based IDS

Another type of IDS was in the focus of the researchers: anomaly-based IDS. The latter contains two other sub-categories : specification-based and behavioral-based IDS.

2.2.2.1 Specification-based IDS

In [Carcano et al. 2011], the authors propose a critical state-based IDS which manages and monitors the evolution of SCADA states. This approach is composed of three main modules to detect abnormal activity and lead a SCADA system to a critical state. These modules are: "System Description and Critical State Representation" where the system is described using Industrial State Modelling Language (ISML). ISML formalises the condition-action of an industrial system. "A State Evolution Monitor" which tracks the system states and how they evolve. Using ISML, a virtual image of the managed system is created as an in-memory model. The latter is fed by network traffic to behave like a real system. And the third is "A Critical State Detector" which establishes whether goes one of the previous monitor states through a Critical State (CS) or not by checking the CS rules defined using ISML language.
To the previous modules, the authors have added multidimensional metrics to predict the critical state. These metrics are: state-state distance, state-critical states distance and distance evaluation metrics. Their performance seems good but their proposed IDS targets a particular class of attacks aiming to lead an industrial system to a critical state.
In [Parvania et al. 2014], the authors propose a new hybrid IDS approach. The first step of this approach is the definition of a set of rules describing the normal behavior. They define

five rules between rules related to network, like master IP, and others related to process, like operation time. After defining these rules, the authors model the normal behavior and consider any behavior which deviates from the normal model as an intrusion. They applied their approach on the Modbus protocol and use Bro to implement their rules. They test their approach in an emulated environment with a real PLC. The authors rate the detection attack of their IDS as accurate and that it gives a good performance.
In [Caselli et al. 2016], a new specification-based IDS approach is proposed using the documentation of the networking system. The documentation is automatically used to reduce the human effort required to define the specification rules. The approach is composed of three steps: system discovery, features lookup, and rule definition. This IDS is implemented in a real environment using Building Automation and Control protocol (BACnet) and identifies process control errors and the level of danger caused by misconfiguration.
In [Kabir-Querrec et al. 2015], the authors present a new resilient IDS implemented in the Ethernet layer. It analyses the vertical flows exchanged between the devices in the ground level and SCADA system. The latter receives an alert if an intrusion is detected. This IDS is implemented for an electrical station using GOOSE protocol. The authors simulated Ethernet storms and usurped GOOSE attacks. After receiving alerts, the Intelligent Electronic Device (IED) control is rewritten to consider the alerts.
In [Koucham et al. 2018], the authors propose an approach to mine specifications from attack execution traces and detect violations for a sequential process. They use Sequential Function Charts (SFC) which is suitable for this kind of process. Their specifications consist of events and states of the evolution of both sensors and actuators. The proposed approach is composed of two steps. The first one is the mining and the inference of the specifications from traces captured between supervisors and PLC. The second step is a monitoring phase which raises an alert when a specification violation is detected. The authors propose a novelty which eliminates the redundancy to increase the alerts pertinence for a quick reaction by the operator.
In [Monzer, Beydoun, and Flaus 2019], a model-based approach for an IDS is proposed to protect a CPS. Their approach contains three stages: model construction, model transformation, and online implementation of the IDS. In the first step, the researchers define the system model using a formal modelling language, then this model is transformed into rules by using a model converter and finally, the generated rules are embedded in an open source IDS. Their results are not conclusive enough.
In [Lin et al. 2013], the authors propose a specification-based IDS using Bro. They modi-

fied the latter by adding a DNP3 protocol analyser to generate specific SCADA operation events. The semantics for each event are stored in the corresponding event handler. To analyze these semantics, they set up the protocol validation policies by defining the event handler in terms of Bro scripting. The policies interpreter script runs code to produce scan results, such as abnormal network activity alerts. The approach is tested in an experimental simulation. The authors believe that the proposed DNP3 analyzer holds promise to work in real SCADA systems.

2.2.2.2 Behavioral-based IDS

In [Tsang and Kwong 2005], the authors present an anomaly-based IDS using an unsupervised multi-agent learning model. Their approach uses a new improved technique called Ant Colony Clustering Model (ACCM). The authors use several techniques to refine clustering. First, it combines the entropy of information and the density similarity mean to identify the spatial regions of the clusters. Secondly, it uses cluster-pheromone to look for the clusters compact and object-pheromone to help the lost ant to be recovered. This mechanism helps the optimal formation of clusters. Third, the model uses a selection scheme to control the diversity of the agent population. In this work, Principle Component Analysis (PCA) is used to get the optimal cluster structure and reduce dimensionality for a better clustering. The tests show a good performance in terms of attack detection and false-positives rate reduction thanks to increasing the principle components (PCs) number.
Another behavioral-based IDS is described in [D. Yang, Usynin, and Hines 2006]. This IDS uses the pattern matching technique built from the normal behavior of network traffic and hardware operation statistics such as CPU usage connection failure. The authors use a previously developed condition monitoring technique, the Continuous System Telemetry Harness (CSTH), originally designed by Sun Microsystems. They use it to monitor server activity and model the profile of normal behavior (called a base). This base is then incorporated into MATLAB using Process and Equipment Monitoring (PEM) to monitor future operations and detect the malicious behavior. This detection is made by comparing the current activity with the learned normal behavior using AutoAssociative Kernel Regression (AAKR) model. After this normal behavior modeling, a prediction is generated. To decide about the nature of these prediction, Sequential Probability Ratio Test (SPRT) module is used according to a confidence interval. This work is potentially extensible and can be used for other intrusion scenarios. However, setting the threshold

value in the SPRT to determine false alarms and false negatives seems arbitrary. To test their approach, the authors build a local network for a simulated SCADA system.
In [Cheung et al. 2007], another behavioral-based IDS is proposed by using a model-based approach. The main basis of this approach is building patterns that characterise the expected behavior of a system and detecting attacks on a system that runs off these models. In this approach, the authors use three techniques which consist of protocol-level models that define normal behavior using features characterising the Modbus/TCP protocol. Then they use Snort tool to define the rules allowing the model violation detection. The authors specify then the expected communication patterns between network components and their access policies. Finally, they use a last detector based on a heuristic approach to detect any first time observed changes in servers or services. This last detector keeps a log of the whole components state to avoid duplicated alerts. The authors mention that the proposed approach is effective to monitor SCADA networks and detect attacks.
In [Barbosa, Sadre, and Pras 2016], the authors present a PeriodAnalyser tool which uses an automated NIDS approach. This tool is based on message repetition and temporal information to define periodic cycles in the network traffic. Their approach is composed of three modules called multiplexer module which analyses the application headers to filter the Modbus/TCP and MMS packets and multiplexes them into different flows. Two flow kinds are identified at this stage which are long-lived and short-lived connections. The tokenizer module which transforms the filtered flows into a standard format independent of the protocol. And finally, a learner module is in charge of identifying all the cycles in the previous pre-processed flows. The authors use three traces of a real environment to evaluate the approach which shows a good modeling accuracy.
In [Maglaras and Jiang 2014], the authors propose a new IDS approach to protect a SCADA system. They use a One-Class Support Vector Machine (OCSVM) classifier which uses offline network traffic for its training step. The architecture is hierarchical since all IDS refer to a Security Information and Event Management (SIEM) database. Alerts are sent to the latter using the Intrusion Detection Message Exchange Format (IDMEF) standard. Their IDS is composed of four steps: classification, training, anomaly detection and sending of the IDMEF messages containing the incident source, their time and the alert classification. The accuracy of this IDS is good but still needs to be improved.
In [Shang et al. 2015], the authors propose an industrial IDS using an improved OCSVM PSO-OCSVM model. They use particle swarm optimisation to train their model with

only one class. The approach is tested in a simulated environment. The efficiency of the IDS is high with 96% detection accuracy and a short training time.
In [Qian et al. 2020], the authors propose an IDS for a SCADA system aiming to detect both attacks against the process and those coming from the network. To detect attacks targeting the industrial process disruption like MITM, replay and zero-day, they use a process-state validation. For that, they use a white list and correlation scores to validate respectively numerical and non-numerical parameters. For other attacks like DoS and buffer overflow, a nonparallel hyperplane-based fuzzy classifier is implemented. Their evaluation shows good accuracy comparing to both SVM and fuzzy-based SVM classifiers. However, their approach does not provide the type of the attack nor its location.
In [Liu et al. 2020], the authors outline a new hierarchically distributed intrusion detection scheme for industrial CPS composing of three layers: the Perceptual Executive Layer (PEL), the Data Transmission Layer (DTL) and the Application Control Layer (ACL). The authors propose a detection technique for each layer. For PEL, they use process noise and measurement noise-adaptive Kalman filter (PNMN-AKF) with spare Bayesian and relevant vector machine (RVM) classifiers to detect attacks. PNMN-AKF uses an estimation of the distributed system states. In the DTL layer a network communication anomaly based detection system is proposed. It consists of the measuring of distance between the actual network distribution characteristics and the normal ones. Kolmogorov–Smirnov (KS) divergence is used for this measurement. Finally, for the Application Control Layer (ACL), the authors use cyclic redundancy check (CRC)-based identity authentication for an accurate detection since compromising the application layer could be critical. This technique is coupled with a Regularised Sparse Deep Belief Network (RSDBN) model using the Restricted Boltzmann Machine (RBM) for its training step. Using an OPNET simulated environment, the efficiency of the authors IDS is demonstrated by a low false positives rate.

2.3 Industrial IDS for IoT equipment

In [Raza, Wallgren, and Voigt 2013], the authors expose a new IDS called SVELTE, which is mainly aimed at routing attacks but it can also be extended to types of attacks such as DOS. SVELTE has been designed to detect routing attacks as well as Sybil and clone ID attacks.
SVELTE uses Routing Protocol for Low-Power and Lossy Networks (RPL) as the rout-

ing protocol and consists of three components: the 6LoWPAN Mapper (6Mapper), the intrusion detection system IDS and a mini-firewall. First, the 6MAPPER placed in the 6LoWPAN Border Router (6BR) collects information from the RPL network and allows the building of the Destination-Oriented Directed Acyclic Graph (DODAG). Mapping requires that mapping packets should not be distinguished from others to avoid selective forwarding attacks. This is done by encrypting the data and ensuring that the headers do not reveal that this package is being used by the Mapper. Secondly, the Intrusion Detection (IDS) component detects three types of intrusions: data altering attacks (consistency checking), sinkholes (checking the routing graph) and selective forwarding attacks (verification of the node availability). Additionally, the mini firewall is integrated into the 6BR and in nodes with limited capacity for attacks coming from outside. In conjunction with protecting from the well-Known external attackers, it can also block the external malicious hosts specified in real-time by the nodes inside a IPv6 LoW Power wireless Area Networks (6LoWPAN).

SVELTE uses a hybrid intrusion detection approach (signature-based and anomaly-based detection), employing IPv6 as network protocol, RPL as routing protocol, and finally IPsec and DTLS as security protocol.

The approach uses the hybrid placement strategy in which the intensive-process modules are put in the 6BR including a mini firewall and the least intensive ones in the low-resource nodes. It has been evaluated for selective forwarding attacks and sinkholes.

In [Sedjelmaci, Senouci, and Feham 2013], the researchers introduce a framework where the intrusion detection process is carried out in three levels, as outlined subsequently. In the low level, a set of nodes called IDS agents monitor the communication of their neighbors and report their feedback to their Cluster Header (CH) for further detection. Firstly, nodes gather the packets within their radio range and pass them to the intrusion detection module for analysis. Then agents identify any suspicious behavior, using the specification-based detection technique. This technique relies on a set of rules to detect and prevent the malicious behavior. In the medium level, the detection procedure is divided into three steps: the first one, known 'features selection', uses Received Signal Strength Intensity (RSSI) and PDR as input data, the second one is SVM training, and the last one is binary classification protocol. A powerful CH uses the Support Vector Machine (SVM) training technique to detect any anomaly. This approach allows separating data into two classes (normal and anomalous). It is called a binary classification given that no node is assumed to be trustworthy, a reputation mechanism is applied at the CH

in order to evaluate the trustworthiness of their IDSs membership. In the high level, each CH monitors its CH neighbors on the basis of a specification detection technique and sends a ballot form to the base station containing the suspected CH. The base station is used as the counter to collect the votes that are generated by CHs in order to make a final decision on any suspected node that may be found. Their results are encouraging, depicting a low number of false positives, a high detection rate, and fast detection time.
In [Pongle and Chavan 2015], a novel IDS is proposed based on the wormhole attack detection system for resource constrained devices using IPv6 border router (6BR). The placement for the IDS uses a hybrid approach, in which centralised modules on 6BR and distributed modules on the sensor nodes cooperate to detect attacks. They use a 6BR in which four modules: the Neighbor Validation (NV), the RSSI Distance, the RSSI collection and, the Attacker Detection.
The NV is responsible for collecting the data and checking the consistency with the distance by referring to the transmission range. The RSSI Distance collects distance and converts it. The RSSI Collection compares received RSSI values, compromised nodes, and neighbor nodes and destroys packets sent twice. Finally, the Attacker Detection module compares the value of the RSSI with the range of distance to detect the compromised node. The authors also use a distributed architecture in which all the nodes participate and consist of four modules. The first one is the Send Neighbor Info module' in which, if information changes, the nodes send an update to the BR. Secondly, the 'Packets Forwarding module' which avoids losses due to the User Datagram Protocol (UDP) protocol. Finally, the 'Monitoring RSSI module', refers to the instance when a packet is exchanged between two compromised nodes, a 3rd node calculates their RSSI, then with a broadcast of compromised packets, the attacker is located. Their approach is designed for the wormhole attack and their results in terms of packets overload, power consumption, memory consumption and true positive rate are encouraging.
In [Saeed et al. 2016], the authors propose an approach called Intrusion Detection and Prevention Mechanism (IDPM). The IDMP consists of two layers, the first one is based on learning the normal behavior of the system using a Random Neural Network (RNN) taking diverse datasets, and covering both valid and invalid cases, as input parameters. The trained RNN model is then embedded in the base station of the IoT system to detect any anomalous behavior and to prevent its propagation. The second one is designed to detect a wide range of Illegal Memory Accesses (IMA) bugs and data integrity attacks. The proposed solution also acts as a health monitoring system for the IoT sensor nodes

by analyzing data transmitted to the base station. As in the case of any malfunction, the valid sensor node may either stop its operation and/or transmit invalid data to the base station. The RNN model has been trained to detect such cases as an intrusion and report them to the main server. The proposed solution effectiveness and performance overhead are measured for an existing IoT system consisting of sensor nodes transmitting data to a base station. Through an experimental setup, it is shown that without a proper security mechanism, it is possible to intrude into the application running on the base station. Furthermore, it is also demonstrated that the base station successfully detected the presence of the malicious sensor node when the given IoT device is enabled with the proposed IDPM.

In [Thanigaivelan et al. 2016], the IDS principle approach is to find any anomaly in the network by monitoring the characteristics of the neighboring nodes at one hop. The system learns and derives the normal behaviors of the monitored information. They use a distributed approach in which all of the observed information is locally managed in the nodes and the exchanges with the parent node appear only if an event is reported. The system has a configurable profile in which the criteria for critical tasks such as detection process interval, memory, threshold and grading parameters are defined. Their approach consists of four modules: the first one is in charge of the data collection and analysis. The second one processes the release of the report to the parent via the DPO (distress propagation object), the third one rejects the packets based on the status of the neighboring node, isolating the compromised node once the intrusion is detected and the fourth one is the BR module that, in the event of a positive anomaly decision, alerts the user. The authors did not disclose any details neither about their implementation nor their results.

The authors propose a hybrid T-IDS for Sybil attacks in [Medjek et al. 2017], it is a distributed, a cooperative and a hierarchical IDS, invloving Border Routers (BR), 6LoWPAN Border Router (6BR) and Monitoring Nodes (MN). Each actor has a specific goal. The BR holds the list of all of the Network's Nodes (NN) and their respective states and maintains the list of nodes allowed to access the network. In NN, each node is associated with a Trusted Platform Module (TPM) identifier, a node identifier associated with the TPM-ID, the status flag of the node (Mobile, Static), and the 6BR prefix associated with the node after deployment. When a node wants to join the network, it must first be registered to the NNs list. BR also has a list of malicious nodes for all 6BR subnets. 6LoWPAN Border Router (6BR) which maintains three dynamic lists: the first list contains 6BR (6BRAN) area nodes. 6BRAN is developed and updated by the BR and

transferred to 6BR in a secure channel. The second contains mobile nodes (MON) and the third list contains malicious nodes (MAN). The 6BR is responsible for setting the maximum delay response field in the DODAG Information Object (DIO) message and the last actor is Monitoring Nodes (MN) which maintains a suspicious node list (SUN) and a malicious node list (MAN). They also keep a copy of the MON list developed by 6BR.
T-IDS consists of three modules: IdentityMod, MobilityMod and IDSMod. For IdentityMod module, after deploying the nodes, and before starting the construction of the RPL topology, the BR uses IdentityMod to configure the 6BRAN list of each 6BR in the network. This list will be used to control access and authenticate nodes. The MobilityMod module, manages hierarchically the mobility with the collaboration of BR, 6BR and nodes in the network. MobilityMod is used by different actors to maintain the network state regarding mobile nodes. In fact, 6BRAN contains the mobility status of each node. Intrusion Detection Module (IDSMod) detects attacks, whenever the IDSMod queries the IdentityMod and the MobilityMod to check if the node belongs to network and if it is a mobile node. Therefore, with minimum knowledge, and observing, by collaborating, the nodes can detect the nodes behaving in a suspicious way based on certain RPL rules. Even if the authors show how T-IDS can deal with a SybM attack, the resources appear to be costly.
In [McDermott and Petrovski 2017], the authors have chosen the anomaly-based detection as a method of intrusion detection in wireless sensor networks. To test and compare detection rates, a Multi-Layer Perceptron Backpropagation Neural Network (BPN) was chosen from the architectures and was compared to a Support Vector Machine (SVM) classifier. In their approach, the authors use NSL-KDDTrain+_20Percent as a dataset. Before starting the training step, the authors have processed and normalized the data to convert the raw input data into an appropriate format which the machine learning the algorithms could use for subsequent analysis. The authors then trained the model with the chosen dataset. Finally, they employed the SVM for the classification and the detection phase. In this work, the authors have focused on DOS (Smurf, Neptune, Back, Teardrop, Pod, Land) attacks. Their results show a very good performance with the neural network method for the majority of attack types but the results are not significant when they use a few samples as for "Pod" and "Land" attacks. The SVM results are better than the BPN which was also able to detect attacks for very low samples such as "Pod" and "Land".
In [Becker and Vester 2017], the author proposes an approach where one of its compo-

nents can be described as an outer shell that acts as a wrapper for the entire application. Another component is a program simulating the normal activities of a sensor node in terms of collecting sensor values and sending them to the network. This method reads the energy consumption of the node and stores the values of the variation of energy consumption over time in an appropriate data structure. An algorithm analyzes the collected values to detect attacks. If the energy consumption of the node is uniform and somewhat linear, the detection method can take advantage of the linear regression and the history of energy consumption to predict the value of the upcoming energy consumption value. Due to the uncertainty regarding the power consumption profile of a sensor node, if the power consumption is not linear, another algorithm is used to collect and compare the variation of the average energy consumption during a period of energy consumption (node history). This method uses a past history to calculates an average value. If the computed average values increase for a consecutive period for a node, the method concludes that the node is under attack. A second checking is proposed using RPL routing protocol involving three roles: the child, the parent and the grandparent.
To test and evaluate behavioral-based IDS especially those based on machine learning algorithms, a reliable dataset is required. In the following section, a state-of-art of the existing datasets is established by classifying them into public, non-public and industrial ones.

2.4 Existing datasets

2.4.1 Public datasets

The DARPA 98 and DARPA 99 datasets from the MIT Lincoln Laboratory are amongst some of the most well-known. Both datasets use network traffic captured from a simulated environment and lack actual attack data records [Brown et al. 2009][Hettich and Bay 1999]. The KDD CUP 99 dataset improves the DARPA98 version and is one of the most widely used datasets for NIDS evaluation. However, one of the criticisms made against the KDD CUP 99 dataset is the use of data duplication [Tavallaee et al. 2009].
In [Tavallaee et al. 2009], the authors propose the NSL-KDD dataset based on KDD CUP 99 which mitigates the weaknesses of the latter.
In [S. Garcia et al. 2014], the authors proposed a malware dataset. It contains botnet, normal and background traffic generated from several malware scenarios. The authors labeled the malicious traffic based on the IP addresses used by the botnets.

In [Ring et al. 2017], the authors depict a dataset called CIDDS-002. They emulated a small business environment using the OpenStack software platform and subsequently simulated the internal, the real and up-to-date attacks from the internet. The main criticism of this dataset is the fact that it is not captured in a real environment and there is no heterogeneity and diversity in the simulated attacks as it is mentioned in [Gharib et al. 2016].
In [Sharafaldin et al. 2018], the authors propose a reliable and publicly available dataset called CICIDS2017 that was captured in a real environment over the course of five days. The dataset contains more than 80 features extracted and calculated from normal and malicious traffic. The authors use the CICFlowMeter software published by the Canadian Institute for Cyber-security website [Lashkari et al. 2017] for the extraction. They provide a more exhaustive review of their dataset, however it lacks the updated attacks.
In [Maciá-Fernández et al. 2018], the UGR'16 dataset is presented. This dataset is designed for the anomaly-based detection algorithms that consider the cyclostationary nature of traffic data. The netflow traces are collected over four months in a real environment.

2.4.2 Non-public datasets

Researchers propose a flow-based dataset called SANTA in [Wheelus et al. 2014]. Flow is defined by traffic that has the same 5-tuples (source IP, destination IP, protocol, source port, and destination port). This Dataset contains real traffic and different attack scenarios. Its labelling was done by manual analysis and heuristics. In [Zuech et al. 2015],the authors present IRSC, which is another flow-based dataset, for IDS. The authors collected network flows as well as full packets. These datasets have been criticised for not being publicly available.
In [Sharma, Singla, and Guleria 2018], the authors detail a new labeled flow-based DNS dataset viz called PUF in order to detect compromised hosts in a network. The data was captured in a real environment over the course of three days, and the labelling process was performed using logs generated by an Intrusion Prevention System (IPS). The criticism of this dataset is the lack of a variety of attacks.
In [Bhattacharya and Selvakumar 2014], the researchers discuss a labeled dataset containing 28 attributes divided into host-based and network-based attributes. It contains 200000 labeled data points. Its data volume duration is four hours, performed in an emulated environment. Unfortunately, the dataset is not publicly available.

2.4.3 Industrial datasets

Other researchers have proposed industrial datasets. In [Morris and W. Gao 2014], the authors propose two datasets: the first dataset contains transactions from the gas pipeline system, and the second one contains transactions from the water storage tank system. Both of them contain network traffic information and the current state of the process control system defined from the payload content. Both datasets are labeled.
Researchers generate in [Lemay and Fernandez 2016] a labeled dataset for electric infrastructures that relies heavily on SCADA networks. Their system is implemented in a SCADA sandbox. The authors simulate some attacks and label the captured traffic. The environment configuration is detailed in their paper and the dataset is publicly available. The criticism made to this work is the lack of diversity of attacks and their unrealistic experimentation environment.
Other researchers have presented their industrial evaluation dataset in [Hijazi, El Safadi, and Flaus 2018] but they do not give the details about the configuration environment, the simulated attacks or the capture duration. In addition, the dataset is not publicly available.
In [Almalawi, Fahad, et al. 2015], the authors use eight datasets, five of which are proposed in other works and are publicly available and three of which are captured from a real urban waste water treatment plant. The latter are composed of raw sensors measurements related to the water level readings of a tank and the status of three pumps correlated with the temperature and the humidity. These measurements are transformed into a set of distributions to find the limit between the normal and critical states. The datasets are labeled and consist of 38 process parameters and are not publicly available.
To propose an IDS based on neural networks, in [Linda, Vollmer, and Manic 2009], the authors record five datasets composed of 20000 packets. The dataset was captured only for normal traffic containing 100000 packets in a simulated environment. Certain information related to the capture duration, simulated attacks and extracted features are missing, and there is no information whatsoever about the labelling process. Additionally, this dataset is not publicly available.
In [J. Gao et al. 2019], the authors use an industrial dataset composed of 19 features. Some of the features are related to the network and others are a sensory data about a tank system. It is extracted in a simulated environment and contains normal and malicious traffic. The authors have labelled their dataset.

Table 2.1 – Overview of the existing IDS

Year	Approach type	IDS type	Attacks type	IDS placement	Protocols	Techniques	Resp type	Metrics Evaluation	Tools used	Ref
2005	Behavioral	NIDS	DOS U2R, R2L, Probe	Decentralised (MAS)	TCP, UDP, ICMP	Multi-agent system with the ACCM model	Passive	Detection accuracy (average attack detection r...	ICA algorithms: Infomax ICA, Extended Infomax ...	[Tsang and Kwong 2005]
2006	Behavioral	HIDS	DoS, ping flood, Jolt2	Centralised (SCADA)	SNMP	Matching patterns	Passive	Parameters of audit server	CSTH and MATLAB	[D. Yang, Usynin, and Hines 2006]
2007	Behavioral	NIDS	Zero day and DOS	Centralised (SCADA)	Modbus / TCP	Matching patterns	Passive	Not specified	EMERALD Bayes sensor, expert-Net SNL and Snort	[Cheung et al. 2007]
2011	Specif-based	NIDS	system critical state	Centralised (SCADA)	Modbus / TCP	Matching patterns	Passive	Not specified	Not specified	[Carcano et al. 2011]
2013	specif-based	NIDS	DNP3 malformed packets	Centralised (SCADA)	DNP3	deterministic	Passive	Detection accuracy, latency, and analysis of the flow	VMware virtual machine with a single logical p...	[Lin et al. 2013]

Table 2.2 – Overview of the existing IDS

Year	Approach type	IDS type	Attacks type	IDS placement	Protocols	Techniques	Resp type	Metrics Evaluation	Tools used	Ref
2014	Specif-based	NIDS	Injection, insulation of an electrical transformer, controller behavior imitation	SCADA and automates	Modbus / TCP	deterministic	Passive	Attack detection capability	Bro	[Parvania et al. 2014]
2014	Behavioral	NIDS	MITM, SYN Flooding and honeypot	Centralised (SCADA)	DNS, FTP, MODBUS, TCP, UDP	Probabilistic (one-class SVM)	Passive	Accuracy of the classification	Not specified	[Maglaras and Jiang 2014]
2014	Behavioral	NIDS	MITM	Centralised (SCADA)	Modbus / TCP	Probabilistic (unsupervised)	Passive	FPR , F-score, accuracy, detection time	Matlab	[Almalawi, Yu, et al. 2014]
2014	Signature	NIDS	Not specified, SYN Flooding and honeypot	Centralised (SCADA)	Modbus	deterministic	Passive	Accuracy of the detection	Not specified	[Diallo and Feuillet 2014]

Table 2.3 – Overview of the existing IDS

Year	Approach type	IDS type	Attacks type	IDS placement	Protocols	Techniques	Resp type	Metrics Evaluation	Tools used	Ref
2014	Behavioral	NIDS	MITM	Centralised (SCADA)	Modbus / TCP	Probabilistic (unsupervised)	Passive	FPR , F-score, accuracy, detection time	Matlab	[Almalawi, Yu, et al. 2014]
2015	Behavioral	NIDS	suspicious code injection	Centralised (PLC)	Modbus / TCP	Probability (one-class SVM)	Passive	Detection Time	Wire- shark, Unity Pro	[Shang et al. 2015]
2015	Specif-based	NIDS	GOOSE and usurped Ethernet storm	Centralised	GOOSE, Modbus / TCP	deterministic	Passive	TPR, destroyed packet number, analysis time GOOSE	Bro, Suricata, Snort, Metasploit, Scapy, Simulink, Matlab	[Kabir-Querrec et al. 2015]
2016	Behavioral	NIDS	Injection data, DOS, network attacks	Centralised (SCADA)	Modbus / TCP and MMS	Probabilistic (machine learning, patterns)	Passive	Traffic periodicity	Not specified	[Barbosa, Sadre, and Pras 2016]
2016	Misuse	NIDS	malformed packet , DoS , ARP spoofing and MITM	Centralised (SCADA)	IEC 61850 GOOSE	deterministic	Passive	Detection accuracy and porcess time	Wireshark, ITACA(C/C++)	[Y. Yang et al. 2016]
2018	Specif-based	HIDS	Injection suspicious execution traces	Decentralised (SCADA, PLC)	Modbus	deterministic	Passive	Detection accuracy, TPR, FPR	OpenModelica2	[Koucham et al. 2018]
2018	Specif-based	NIDS	Injections of false data	Decentralised (sensors, actuators, PLC)	Not specified	Statistics	active	accuracy	Petri Net	[Sicard, Zamai, and Flaus 2018]

Table 2.4 – Overview of the existing IDS

Year	Approach type	IDS type	Attacks type	IDS placement	Protocols	Techniques	Resp type	Metrics Evaluation	Tools used	Ref
2019	specif-based	NIDS	Disturbing industrial process and MITM	Decentralised (sensors, actuators, controllers)	Modbus / TCP	deterministic	Passive	Detection accuracy	Zeek	[Monzer, Beydoun, and Flaus 2019]
2020	Misuse	NIDS	DoS and Replay attack	Sensors	Not mentioned	Probabilistic (RVM, GMM)	Passive	accuracy, FP	Opnet, Simulink, Matlab	[Liu et al. 2020]
2020	Behavioral	NIDS	DoS (SYN Flood), MITM	SCADA	Modbus / TCP	Probabilistic (SVM, non-parallel hyperplane)	Passive	Accuracy	Matlab	[Qian et al. 2020]

Table 2.5 – Overview of the existing datasets

Research work	Year	Public	Industrial	Duration	Packet	Flow	Environment	label
DARPA98/99 [Brown et al. 2009], [McHugh 2000]	98/99	yes	no	7/5 weeks	yes	no	Emulated	yes
KDD CUP 99 [13]	99	yes	no	n.m	other	other	Emulated	yes
NSL-KDD [Tavallaee et al. 2009]	2009	yes	no	n.m	other	other	Emulated	yes
CIDDS-002 [Ring et al. 2017]	2017	yes	no	14 days	no	yes	Emulated	yes
CICIDS2017 [Sharafaldin et al. 2018]	2017	yes	no	5 days	no	yes	Emulated	yes
UGR'16 [Maciá-Fernández et al. 2018]	2016	yes	no	4 months	no	yes	real	yes
Dataset I,II, III, IV [Morris and W. Gao 2014]	2014	yes	no	n.m	yes	no	Emulated	yes
Modbus_dataset [Lemay and Fernandez 2016]	2016	yes	no	n.m	yes	no	Emulated	yes
kyoto [Song et al. 2011]	2009	yes	no	3 years	other	other	real	yes
UMASS [Prusty, Levine, and Liberatore 2011]	2011	yes	no	7/5 weeks	yes	no	Emulated	n.m
ISCX2012 [Shiravi et al. 2012]	2012	yes	no	n.m	yes	yes	Emulated	yes
SANTA [Wheelus et al. 2014]	2014	no	yes	n.m	other	other	real	yes
IRSC [Zuech et al. 2015]	2015	no	yes	n.m	yes	no	real	yes
PUF [Sharma, Singla, and Guleria 2018]	2018	no	yes	3 days	no	yes	real	yes
[Hijazi, El Safadi, and Flaus 2018]	2018	no	yes	n.m	yes	no	Emulated	yes
[Almalawi, Fahad, et al. 2015]	2015	no	yes	n.m	other	others	Emulated	yes
[Linda, Vollmer, and Manic 2009]	2009	no	yes	n.m	yes	no	Emulated	yes
[J. Gao et al. 2019]	2019	no	yes	n.m	other	others	Emulated	yes

n.m: not mentioned

2.4.4 Conclusions and discussion

The state of the art shows drawbacks , that can be summarized as follows:

- Limitation of the targeted attacks class: some of the proposed IDS target only some particular types of attacks.
- Lack of works targeting MES: In 4.0 industrial systems, ERP, MES, SCADA and some smart Human Machine Interface (HMI) exist as industrial information systems. In the industrial cyber-security field, most researchers focused on the proposal of some approaches to secure either the SCADA system [D. Yang, Usynin, and Hines 2006],[Cheung et al. 2007],[Lin et al. 2013],[Maglaras and Jiang 2014], [Carcano et al. 2011] or the PLC [Koucham et al. 2018], [Huh 2017][L. Garcia et al. 2016]. However none of the proposed works target MES level. The latter represents a sensitive level since it is responsible for the whole planning, execution and management of the productions orders (POs). Stopping, altering or compromising this system could be very costly for a factory.
- Lack of hybrid anomaly-based approaches to discriminate between industrial dysfunction and real intrusions.
- Lack of industrial datasets: due to the complexity of the datasets construction and their time consumption, a salient lack of a reliable datasets is observed. Most of the proposed datasets in this field are typically IT-built and captured in and for IT equipment. Few of them are related to the industrial field. In addition, the existing ones are not reliable enough and do not respect the eleven reliability criteria mentioned in [Gharib et al. 2016] and summarized in Figure 3.5.

To respond to and fill these various gaps, the thesis proposes a new approach based on two sub-categories of anomaly-based IDS that are discussed in the next chapter. One of them is a specification-based IDS detailed in Chapter 4 through its principle, its basis, and all details regarding its conception and results. The second one is a behavioral-based IDS explored in Chapter 3 with more details related to its functioning, its basis and its experimental results. In addition to these IDS, a decision-making system has been designed and developed to provide the nature of the detected anomalies. It is examined in detail at the end of the dissertation.

Chapter 3

Intelligent behavioral based IDS

3.1 Introduction

Today, and more than ever, industry needs an efficient means to protect its equipment and its communications. Malicious attacks have become more elaborate. Intruders have improved their operating mode and use different evasion techniques to prevent detection by an Intrusion Detection System (IDS) [Khraisat et al. 2019]. Recently, hackers organise themselves and have improved their cyberattack techniques by performing more zero-day attacks than a simple malware. Therefore, the critical challenge today is the design and the development of an efficient IDS to protect the ICS (Industrial Control System) rather than conventional security solutions.

Among the existing defences, we note anomaly-based IDS which are efficient enough to be used in either industry or IT environments. Unlike signature-based IDS, they can detect new threats because all events that differ from a normal behavior are considered as an attack.

For these reasons, a behavioral-based IDS is used in this thesis, which is one of the sub-categories of anomaly-based IDS used for its efficiency and robustness against advanced and sophisticated attacks.

The main purpose of this IDS is to learn the normal behavior of the industrial platform presented in Section 3.2 and to consider all that deviates from this behavior as an intrusion. To implement this IDS, one of the artificial intelligence techniques called the neural network has been used.

Testing this kind of IDS requires a reliable dataset. In this thesis, we have captured and built our own industrial dataset that is exposed in Section 3.3. This chapter is concluded with the results of this IDS detection using some conventional performance metrics.

3.1.1 Intelligent behavioral based IDS: principle

This behavioral-based IDS aims to analyse network traffic to detect intrusion inside our industrial platform. The IDS learns from normal network traffic, the nominal behavior of our system and considers any activity which differs from this normal learned behavior as malicious activity. To achieve this purpose, several machine learning algorithms were possible to be used. However, according to our requirements and the study context, the neural network algorithm was chosen as basis for this behavioral-based IDS. The motivations of this choice are explained latter in this chapter.

3.1.2 Intelligent behavioral based IDS: assumptions

The approach of this IDS is valid whilst respecting several assumptions presented below:

- This IDS uses network traffic. Therefore, the industrial platform has to be connected to the network.
- This IDS uses the neural network algorithm which requires a huge amount of data for its training step to obtain good classification results. Therefore a large dataset is used in this work.
- Most attacks go through the network layer of the Open Systems Interconnection (OSI) model.

3.1.3 Intelligent behavioral based IDS: Neural networks basis

An Artificial Neural Network (ANN) is inspired by biological neuron behaviour and the structure of the human brain [Schuman et al. 2017], [Agatonovic-Kustrin and Beresford 2000]. The biological neuron is a cell composed of a cellular body and a kernel. The cellular body branches out to form what are called dendrites (Figure 3.1). Due to this latter, information is routed from outside to the neuron. Then it is processed by the neuron and it goes through the axon to be transmitted to the other neurons [Igor Kononenko et al. 2007]. ANN imitates the biological neuron functioning with a better accomplished performance.

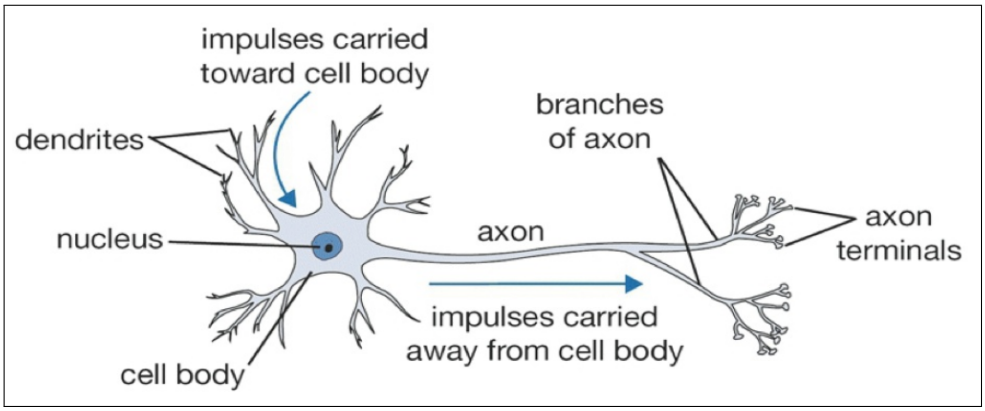


Figure 3.1 – Biological neuron schema

An artificial neural network is a set of elementary objects that we call formal neurons. ANNs are distinguished from each other by theirs complexity levels, their neuron types and their objectives.
In the literature, several kinds of ANNs can be found. However, in this research work, MultiLayer Perceptron (MLP) is used. ANN is organized into three kinds of layers (see Figure 3.2):

- The input layer, which is a layer from which, data is transmitted to the next layers.
- The hidden layers, which interface with the input layer and output layer. The whole computation is carried out inside them.
- The output layer, which is in charge of producing the result for given inputs.

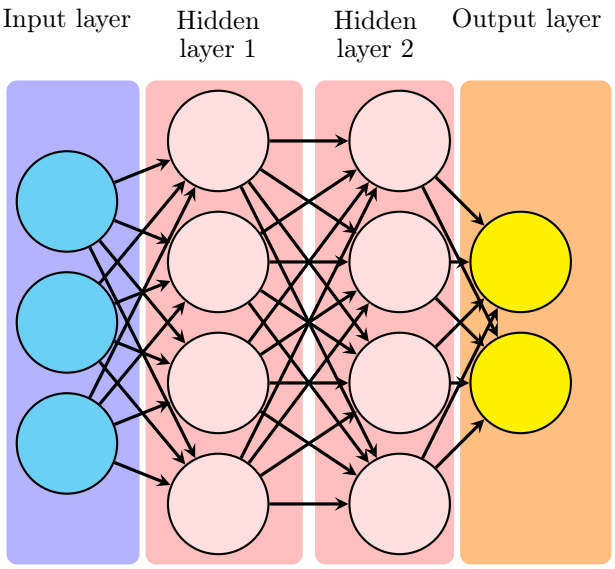


Figure 3.2 – Artificial Neural Network (ANN) structure

In ANN, two learning procedures exist: Supervised Learning and Unsupervised Neural Network Learning [Hodo et al. 2016]. The first one is used when input data are already labelled and categorized. This learning method is often used in feedforward or MultiLayer Perceptron (MLP) models [Dalal 2020]. The second one is much more complex since the system has to receive unlabelled data, detect similarities in the received data during the training process and organize them into categories. Self-Organizing Maps (SOM) is one of ANN which uses the unsupervised learning technique for its training process [Hodo et al. 2016].

3.1.4 Neural network : motivation

In the beginning of this thesis, a multitude of other machine learning models were encountered such as SVM, CNN, RNN... However, following a more in depth study, it was found that these models were not suitable for our study due to the reasons explained below:

- Support Vector Machine (SVM): It is not suitable for a large dataset such as ours since it requires too much time for its training stage, due to the computation of a distance function between each point in the dataset. In addition, this model could be too costly in terms of memory since each computed distance has to be stored.
- Other forms of neural networks such as RNN, LSTM, CNN... exist:

  - Convolutional Neural Networks (CNN) : are more suitable for images data. A one dimensional CNN also exists which is for signal or sequential data.
  - Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) which is a special RNN: These kinds of neural networks are suitable for series time data and sequential data which is not the case with our dataset.

- Decision Trees (DT), Linear Regression (LR) : These models have been tested to some extent in our experimentation. A comparison of ANN and these model's in terms of precision and accuracy is given in the results Section. However, due to shortage of time, they were not explored deeply. Consequently, it is intended to test and develop them in future works.

After studying and comparing the artificial neural network (ANN) to other machine learning models, ANN was chosen for its simplicity, its good efficiency in the intrusions detection field [Saeed et al. 2016][Hodo et al. 2016][McDermott and Petrovski 2017] for its suitability for our study context which consists in detecting anomalies in an industrial process and determining their nature with a high accuracy. So, for the reasons that we summarised below:

- This industrial process deals with a huge amount of real-time data and does not allow for any delay in the execution of its steps. Due to ANN's parallelism characteristic (neuron parallelism and neuron connections parallelism), very fast results could be obtained. Therefore, it managed to overcome the real-time's constraint [BOUROUH and KANOUN 2017].

- Due to its strong discrimination and generalisation capabilities, our classification could be performed with significant success rates.

The proposed approach is generic and can use other algorithms of machine learning. In our case, neural networks model is used for all the reasons previously presented.

3.2 Experimentation platform

The experimentation platform used in this thesis has an educational objective and provides a concrete approach, from commissioning to the final used of a material using a flexible dosing line as those used in the pharmaceutical field. It puts the researcher in

real industrial conditions.
This academic production line entails filling vials with balls and then capping them.

This flexible dosing line consists of two parts: a "Pick and Place"(1) associated with a "Productis" system (2):

1. The "Pick and Place" is a manipulator with two digital axes and one switching pneumatic axis, it is intended for tubes and caps handled in order to be filled by the "Productis" system. After filling and capping of the vials, Pick and Place proceeds to put the vials in cardboard boxes, then to move them to a storage area. Pick and Place cannot be used alone, it must be linked to the "Productis" system.

Figure 3.3 – Experimental platform

The operative part of the equipment consists of three stations and a 3-axis Cartesian robot:

- A station for loading vials and caps.
- A cardboard loading station where the finished vials will be stored.
- A station for unloading boxes to the store.

- A 3-axis Cartesian robot.

2. The Productis system enables the operation, management, maintenance, adjustment and control of an integrated packaging system. It also allows the implementation of multi-product production management and the study of the workshop organizing methods. The operative part or Productis consists of 6 stations:

   - Station 5: Pick and Place loading station.
   - Station 1: distribution of balls by counting.
   - Station 2: closure of the vials by capping, and removal of products declared defective.
   - Station 6: replaces the charging station (station 5) when stations 3 and 4 are in maintenance (in manual mode).
   - Station 3: distribution of balls by counting (identical to station 1).
   - Station 4: closure of the vials by capping, without the removal of the products declared to be defective, this station allows the capping of several sizes of vials.

   The pallets, placed on the conveyor move from one station to another.

3.3 IDS basis: industrial dataset

3.3.1 Methodology

In order to build the dataset, firstly the attacks are performed, then the traffic is captured and finally the features are extracted as shown in Figure 3.6. All these steps have been performed on the academic platform presented in the previous subsection. This platform is hosted in Lab-STICC laboratory at the University of Brest. The architecture used is presented in Figure 3.4.

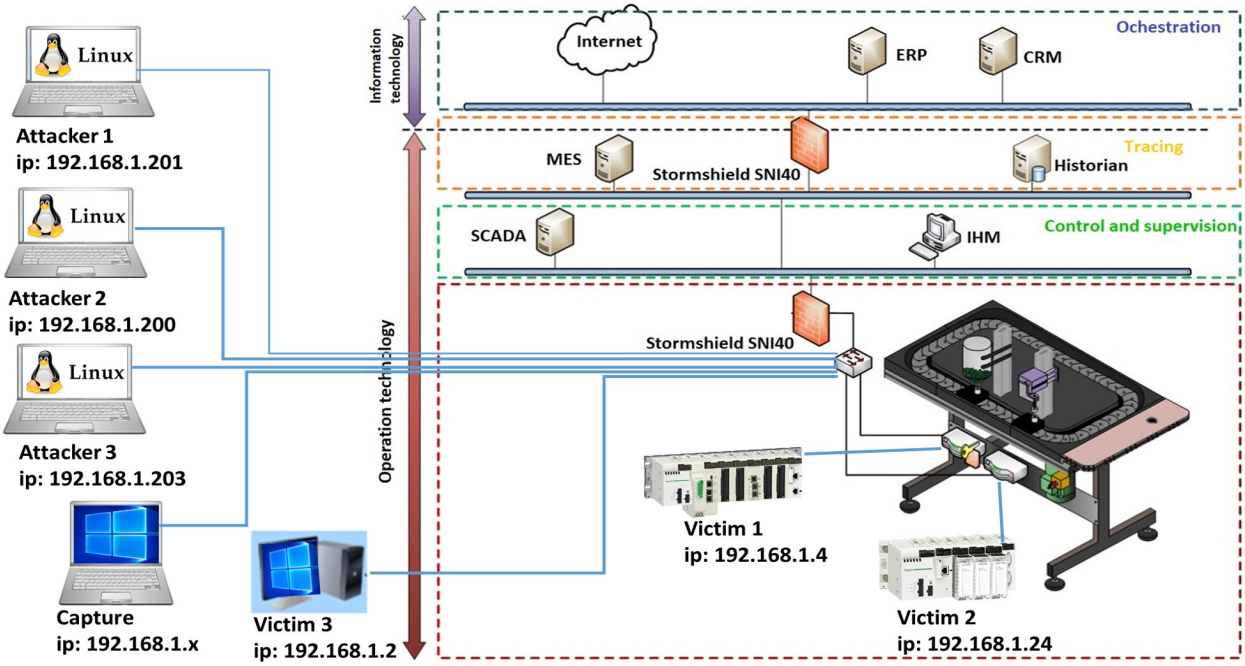


Figure 3.4 – Experimental platform

This experimental platform is composed of:

- Material: Schneider Electric PLC (M340 and M580), three attackers' laptops, one laptop dedicated to traffic capture and a victim computer.
- Software: Unity Pro, Ettercap, Metasploit, Wireshark.
- Manufacturing Executive System (MES): LINA.
- OS: Linux for the attacker, Windows 2012 server for the server, and Windows 10 for the port mirroring capture.
- Switch: one Alcatel-Lucent switch and two industrial switches with three Ethernet ports (resp. five Ethernet ports) embedded in the M340 (resp. M580) PLC.
- Protocols: TCP/IP, Modbus/TCP.

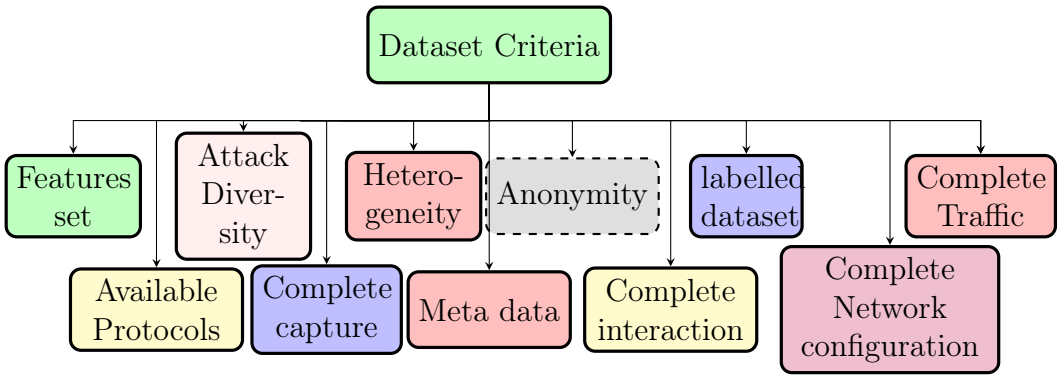
Depending on the attack, the victims could be either a PLC or a PC. A port mirroring setting is put in place to capture all the traffic.

IT attacks such as DoS/DDoS, FTP and MSSQL brute force were carried out, as well as attacks that are typically industrial such as disturbing the production line and the MITM attack applied to a PLC (see Sub-section 3.3.5.2).

3.3.2 Design criteria

According to [Gharib et al. 2016], a reliable dataset should satisfy the criteria as shown in Figure 3.5. Our dataset meets most of them.

Figure 3.5 – Criteria of a reliable dataset



- Complete Network configuration: our network topology (Figure 3.4) includes various types of network equipment such as switches and different operating systems. This topology also integrates industrial-specific equipment such as industrial switches, a Programming Logic Controller (PLC) such as M340 and M580 PLC and a real industrial Manufacturing Executive System (MES).
- Complete Traffic: There are three victim machines and three attacker machines which perform attacks on a real academic industrial platform.
- Labelled Dataset: the dataset is completely labelled with 9 labels. Each sample is labelled as normal or with the type of attack or equipment reaction against an attack. The labelling procedure is described in Sub-section 3.3.7.
- Complete Interaction: internal and external flows are captured in the architecture.
- Complete Capture: probes are positioned at strategic places in the architecture in order to capture and record all the traffic.
- Available Protocols: this dataset deals with all common available protocols, such as TCP, ICMP, Modbus/TCP, HTTP and FTP.
- Attack Diversity: the simulated attacks are for the converging industry. Therefore, the dataset includes the most common attacks related to the Information Technology

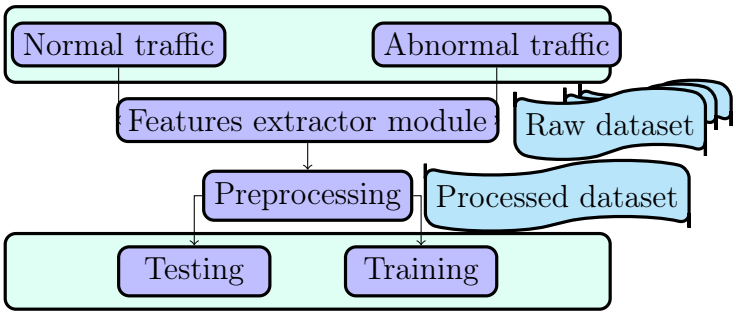
(IT) world such as DoS, DDoS, Botnet, FTP Brute force, http flooding and others related to the Operational Technology (OT) world such as a disturbance of the industrial process, Man In The Middle (MITM) against PLC and altering the data in the MES database through a MSSQL server attack.

- Heterogeneity: malicious and non-malicious network traffic are captured from all victim machines.
- Feature Set: 133 features were extracted to build this dataset from network traffic using the extractor developed during this work. These features will be described in Sub-section 3.3.4.
- Metadata: this dataset contains application and transport layer features. Some of them are basic and others are statistically computed. More details on the type of attacks, the pre-processing and labelling processes are given later in this chapter.
- Anonymity: All traffic will be provided with the payloads and the IP addresses used.

3.3.3 Dataset generation process

In order to build the dataset, firstly the attacks are launched, then the traffic is captured and finally the features are extracted as shown in Figure 3.6. All these steps have been performed on the previously presented academic platform.
Two kinds of attacks have been simulated to build this dataset. The first one is typically IT such as DoS/DDoS, FTP and MSSQL brute force. The second one is typically industrial such as disturbing the production line and the MITM attack applied to a PLC.

Figure 3.6 – Dataset extraction process



In the literature, datasets are either by flow (the traffic that has the same 5-tuples: source IP address, destination IP address, protocol, source port, and destination port) or by packet. Since the dataset is intended to be used in a machine learning algorithm which requires a huge amount of data, every millisecond, a flow is split into sub-flows to generate enough samples to train any kind of machine learning algorithm.

3.3.4 Dataset model

Our dataset model is composed of both transport and application features. It is composed of basic features such as destination and source IP addresses, destination and source ports, protocols..., temporal features such as RTT, TTL.. and expressive features that require statistical calculation such as avg, min, max, std...

The networking features are very helpful in our case because most attacks go through the network. Therefore, the nature of remote attacks can be established based on features such as RTT, duplication and retransmission of packets to detect DoS attacks or MAC addresses and RTT to detect MITM attacks. Or they are based on application features such as registers values, and Modbus time to detect industrial attacks such as a disturbance process. The application features related to Modbus protocol are illustrated in Table 3.1.

Metrics name	Meaning
Request_cnt_funcx	Number of the function code X request = 1,2,3,4,5,6,15 or 16 for the different Modbus function code
Response_cnt_funcx	Number of the function code X response 1,2,3,4,5,6,15 or 16 for the different Modbus function code
Command address	Device ID in command
Response address	Device ID in response
Command_memory_avg	Memory start position in Modbus request: includes internal memory addresses for read and write commands

Response_memory_avg	Memory start position in response packet: includes internal memory addresses for read and write responses
Command_memory_count_avg	The average of the number of memory bytes for R/W response:includes field size for read and write responses
Response_memory_count_avg	The average of the number of memory bytes for R/W response:includes field size for read and write responses
Command_length_avg	The average of the total length of the request packet: the lengths of the Modbus response
Response_length_avg	The average of the total length of the response packet: the lengths of the Modbus response
Time_avg	The average of the time interval between two packets: the time between a Modbus query and its response
Register_number	List of Modbus register numbers
Register_value	List of Modbus register values
Modbus_err_code	Modbus error code
Time_modbus_std	The standard deviation of the time interval between two packets: the time between a Modbus query and its response
Exception_code	Exception code of Modbus protocol
Modbus_err_count	Number of the error of Modbus protocol

Table 3.1 – Application metrics for the Modbus protocol

During this thesis, the ISA-95 standard has been analysed from which 13 industrial anomalies that could be caused by attacks. They are divided into sequential, temporal or content as explained hereafter and, in more detail, in Chapter 4:

1. Sequential

- PO (production order) request/response control: the attacker sends a response that does not match the request.
- WO (word order) sequencing control
- Overlapping PO control

2. Temporal
   - Planned/used total time control
   - Planned/used time per post control
   - Planned/used time between posts control
3. Content
   - Planned/used PersonnelClass control
   - Planned/used EquipementClass control
   - Planned/used MaterialClass control
   - Produced quantity control
   - Equipment and data control
   - Sufficiency resources control
   - Consistency of launching PO control

Due to the Modbus application's features, an IDS can be trained to detect attacks on the integrity of the system (i.e., sequential and content anomalies) and on its availability (i.e., temporal anomalies).
For each category, some attacks are launched in our dataset to offer the possibility of learning the specific pattern.

3.3.5 Attacks simulation for dataset generation

3.3.5.1 Attacks choice motivations

The choice of attacks was motivated by several elements which are detailed below:

- The variety of attacks with respect to the OSI model and the intrusion techniques: attacks targeting the different layers of the OSI model using several intrusion techniques.

- The variety of protocols: attacks targeting several protocols namely: Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), Transmission Control Protocol/Internet Protocol (TCP/IP), Modbus...
- IT / OT Convergence: Since this study targets industry 4.0 which leads to IT / OT convergence, simulations were made on both typical IT attacks such as DDoS, FTP, http attacks and typical OT attacks such as Man-In-The-Middle (MITM) attacks on OT part or disturbing an industrial process which targets the modification of the industrial platform registers.

3.3.5.2 Attack scenarios

As explained previously, the dataset model includes normal and attack traffic. For this work, twelve attacks are simulated on the experimentation platform. Among these attacks, some are typically IT and others are OT as described below:

1. IT attacks:
    - DoS/DDoS: A DDoS attack aims to make a server, service or infrastructure unavailable. The attack can take different forms: saturation of the server's bandwidth to make it unreachable, or exhaustion of the machine's system resources, thus preventing it from responding to legitimate traffic (See Figure 3.7).

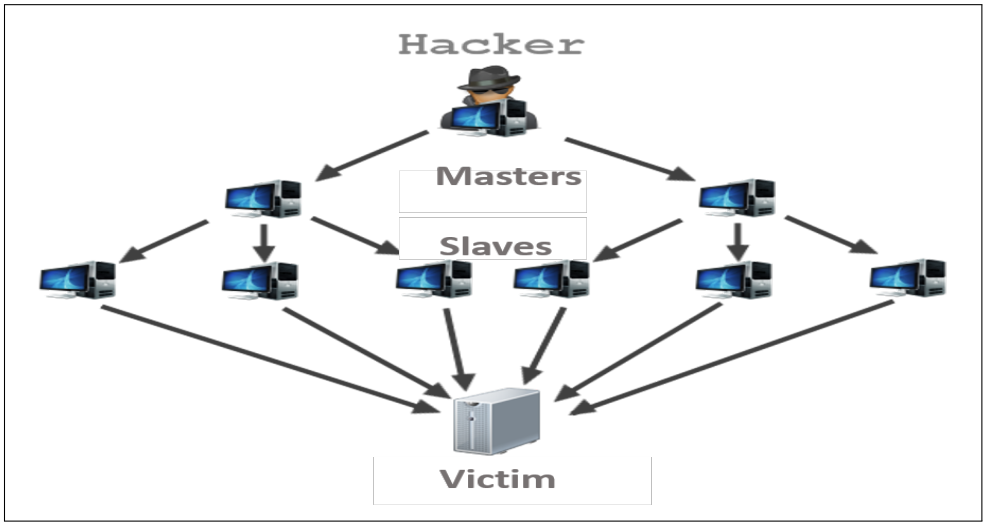


Figure 3.7 – Denial of Service (DoS) attack principle

To perform these attacks, the following were sent; a TCP SYN flood, TCP ACK (Acknowledgement) flood, TCP RST (Reset) flood or Xmax flood by setting all TCP flags (CWR, ECN, URG, ACK, PSH, RST, SYN, FIN). UDP flood packets were also simulated. The script implementing these attacks uses hping3. If hping3 is not found, it attempts to use the nmap-nping tool instead.

- FTP brute force: Brute force attack is a method used in cryptanalysis to find a password or key. It consists in testing, one by one, all the possible combinations. For it to be performed, the dictionary method is used to crack the FTP server connection.
  For the latter, firstly the login and the password of the ftp server are obtained by using either a hydra or Metasploit tool then a connection to the server to retrieve some files and delete others.

- Web http DoS: The principle of this attack is to paralyze the Apache server which uses http protocol to make it unavailable by sending a large number of requests.
  In this attack, a script written in Python called slowhttp-test is used to perform an application layer Denial of Service attacks (DoS). Its principle relies on the fact that requests are not processed before being completely received by the server. If the data is not complete or the speed of sending packets is too low, the resource is kept busy till receiving the complete data. In this attack, Apache is targeted by causing very significant memory and CPU usage on it.

- Botnet: A botnet is a network of computers infected with malware so that they can be controlled remotely, forcing them to send spam, spread viruses or launch DDoS attacks without the knowledge of the real computer's owners and without their approval.
  Ares tool, which is a written remote administration tool RAT in Python, is used to perform this attack.
  It is composed of two programs:
  - Command and Control server (C&C), which is a graphical Web interface for the administration of agents (victims).
  - An agent (or backdoor), which runs on the compromised host ensuring communication with the C&C server.

2. OT attacks:

   - MSSQL attack: It is composed of two steps. The first one is a MSSQL brute force to retrieve the MES database's login and password. The principle is the same as for the FTP brute force but uses the mssql_login module of Metasploit. Then, the second step consists in dropping or altering data by taking control of the machine hosting the MES database.
   - PLC disturbing process: Two attacks are launched to disturb the PLC process. Their principle consists of disrupting the production line functioning by modifying either the recipe scheduled by the operator, or by provoking a remote emergency stop. After studying and analysing the registers' configuration of the platform PLC, we decide to modify the registers responsible for the worst and the most widespread scenarios in industrial attacks, namely the complete shutdown of the production line and the modification of the production line receipt. In our case, register 2 of M580 PLC is responsible for remotely stopping the conveyor, and register 120 of M340 PLC is in charge of seamlessly modifying the number of balls. Therefore, the vials are filled, capped and removed to destocking without the operator noticing the modification of his planned PO. To perform this, we correctly connected to the PLC line and modified the Modbus registers. For this attack, the Metasploit attack tool and its Modbusclient module was used.
   - MITM: This is an attack whose principle consists of intercepting communications between two machines, hosts, servers..., without either of them suspecting that the communication channel between them has been compromised (Figure 3.8).

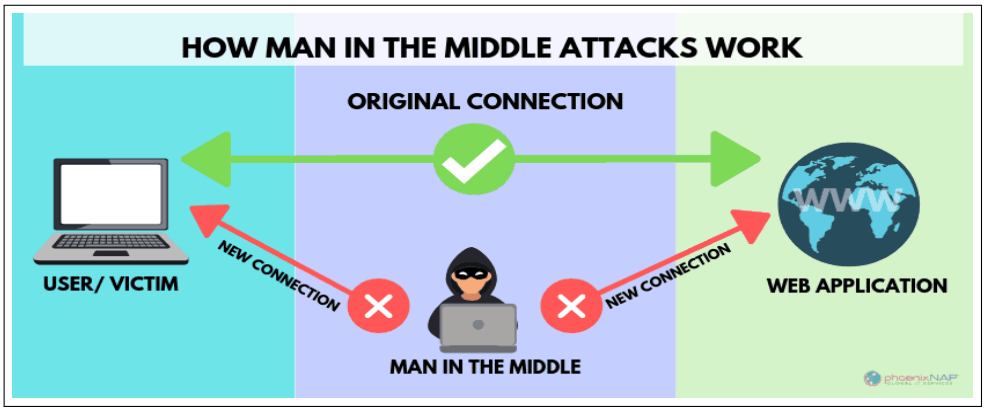


Figure 3.8 – Man-In-The-Middle (MITM) attack principle

For this attack, we entered into the middle, between the PLC and the server using the Ettercap tool then the content of the registers as modified to disturb the production line's nominal functioning. During this attack, either the remote emergency stop is triggered or the operation recipe is modified.

3.3.6 Data acquisition: Extractor

An extractor tool has been developed in Python 3. Scapy, which is a Python library, is used for packet dissection. Scapy deals with packets using a class per protocol. For each protocol, a class is defined in a library which can be customized on an as needed basis.
A customized full duplex session extractor was written to separate protocol sessions. Server and clients are then chosen based on their addresses after the separating session. It is assumed that sessions are initiated by clients.
The protocol's data fields are processed after the extraction to give the required results such as avg, max, min, etc. The processed protocols are TCP, HTTP, TLS, ICMP and Modbus.
This tool extracts more than 100 features every one millisecond, some of them are extracted directly from the network packets and others are computed mathematically. These features are inspired from two other tools which are tstat [Mellia, Carpani, and Cigno 2003] and tcptrace [Ostermann 2005] that only extract the features related to the transport layer of the OSI model. The novelty of the tool, as proposed here, is the fact that it also extracts the application features related to the Modbus protocol. The explanation of all these features is given in Table 3.1.

3.3.7 Labelling and pre-processiong data

The first step for performing this work is the preprocessing process. The raw dataset is composed of several kinds of data such as the IP address, categorical data, digital data, and a list of values. Due to the variety of the data types, a pre-processing process is required to make data usable by any machine learning algorithm. Consequently, non-digital features such as protocols are replaced by numerical values using the LabelEncoder class from the sci-kit learn library. The IP address dots are omitted to get one number instead of an eight-digit number. The features which are in list format such as register numbers or values are split in one value per row. Then the dataset is normalized and formatted to the same scale.
In the literature, three options were found to label the data: manual labelling [Zuech et al. 2015], semi-automatic labelling [Pereira and Nunes 2015], and automatic labelling [Aparicio-Navarro, Kyriakopoulos, and Parish 2014]. In view of the data simplicity (malicious and non-malicious data), the manual labelling was used in this dataset. -1 was assigned to the normal traffic. For malicious traffic, each attack was distinguished by assigning it a different label between 1 and n, where n is the number of attacks simulated in this work. The main novelty of this dataset is the fact that another label was added to characterize the equipment response to the attacks. This label was set to 0.
The dataset was captured so that normal traffic was separated from malicious traffic. This facilitated the normal traffic labelling process. It remained to label malicious packets and distinguish them from those representing the reaction to an attack. To this end, the different extracted features were used. Taking the example of a SYN flood attack, after several attempts and without receiving any ACK flag, communication is cut, the RST flag is set at 1 and State_cnx_S0 and State_cnx_S1 are set at 1. Finally, all packets meeting these conditions as a reaction to the SYN flood attack were labelled. Therefore, all attacks and reactions to attacks packets based on the suitable features were labelled.

3.3.8 Dataset extension approach to other protocols

The methodology used in this chapter is related to the Modbus protocol which is the most widely used protocol in the industry today. However, this approach could be extended to other new emerging protocols such as OPC Unified Architecture (OPC-UA) and, Message Queuing Telemetry Transport (MQTT)... For instance, considering the OPC-UA protocol, this latter uses two transport protocols which are OPC TCP and Simple Object

Access Protocol (SOAP)/HTTP(S). Therefore, we can extract some features related to the OPC-UA subscription step and others related to OPC-UA communication such as the MaxAge feature which provides the operation that the server has to perform and the NodeId Identifier feature used in the operation... OPC-UA is also a protocol which works in request/response mode as for the Modbus protocol. Thus, we can extract some features related to OPC UA request and others related to the OPC-UA response such as the elapsed time between the OPC-UA request and response. Once the features are extracted, the next steps of our approach are the same.

3.3.9 Digital description of the dataset

The used dataset is large which gave a good training of our neural network. It contains around 9 millions lines in its csv file and has a size of 16 GB for its network traffic files. It was captured over five days during which seven attacks have been simulated. It contains 133 features (Figure 3.2).
This dataset was split into training and testing datasets according to respectively 70% and 30% of the whole dataset. More details are given below in the following table and graphic:

Table 3.2 – Dataset digital description

Dataset characteristics					
Lines number	Pcap files size	Capture duration	Attacks	Labels	Features
8735711	16 GB	5 days	7 attacks	9 labels	133

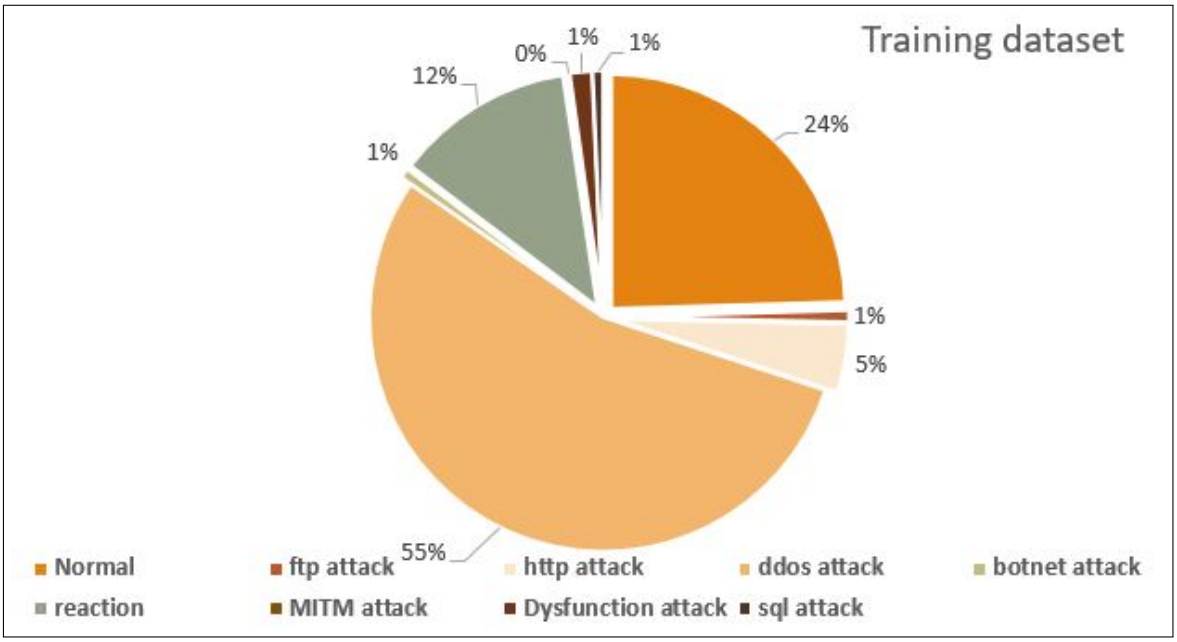


Figure 3.9 – Training dataset division traffic

3.4 Results

3.4.1 Neural network: Experimental parameters

To obtain the optimal neural network architecture, several tests were performed by changing either the layer's number, the layer's neuron number, the activation function, epoch's number or the batch size. In the beginning of the neural network development, either over-fitting or under-fitting phenomenon were encountered. The first one occurs when the neural network is very good at learning its training set, but cannot generalize beyond the training set. The second one occurs when the network is not able to generate accurate predictions on the training set.
To overcome these both problems, firstly the correct splitting percentage between training and test datasets had to be found. After using several splitting configurations, 70% of the whole data was kept for the training dataset and 30% for the testing one. More than 70% led to the over-fitting phenomenon. Secondly, dropout function but the neural network became too complicated and the results was added were not conclusive. Thirdly, we manipulated the layers and the layer's neuron numbers by increasing them. However, the results were noisy and some class predictions were lost.

Finally, a model was kept with 102 neurons in the input layer corresponding to the retained features and 9 neurons in the output layers corresponding to the labels (seven labels for the simulated attacks in addition to one for normal traffic and another for the traffic related to the reaction of the equipment against an attack).
For the training stage, Keras was used with the TensorFlow backend. The Adam optimizer was used because it is the most frequent and it gives better results than the Batch Gradient Descent (BGD). All of the models were trained for 50 epochs. Batches of size 1024 were used because the dataset is large and smaller batches led to slower training with worse results. Table 3.3 gives the chosen model:

Table 3.3 – Neural network parameters

Parameters	Details
Learning	Supervised
Input layer	One input layer with 102 neurons (Used features)
Hidden layer	One hidden layer with 12 neurons
Output layer	One output layer with 9 neurons (9 class)
Number of epochs	50
Activation function	Sigmoid
Performance metrics	Accuracy, Loss, Precision, F1-Score, Recall

3.4.2 Neural network: Graphical interface

As for behavioral-based IDS training and testing development, behavioral-based IDS GUI was developed with Python taking into consideration the previously trained model. This GUI enables the downloading of the suspected traffic, then it prepossesses it and finally it gives a prediction regarding the suspected traffic as shown in Figure 3.10. On the left-hand side, a prediction of normal traffic is given and on the right-hand side, a dysfunction process attack is detected.

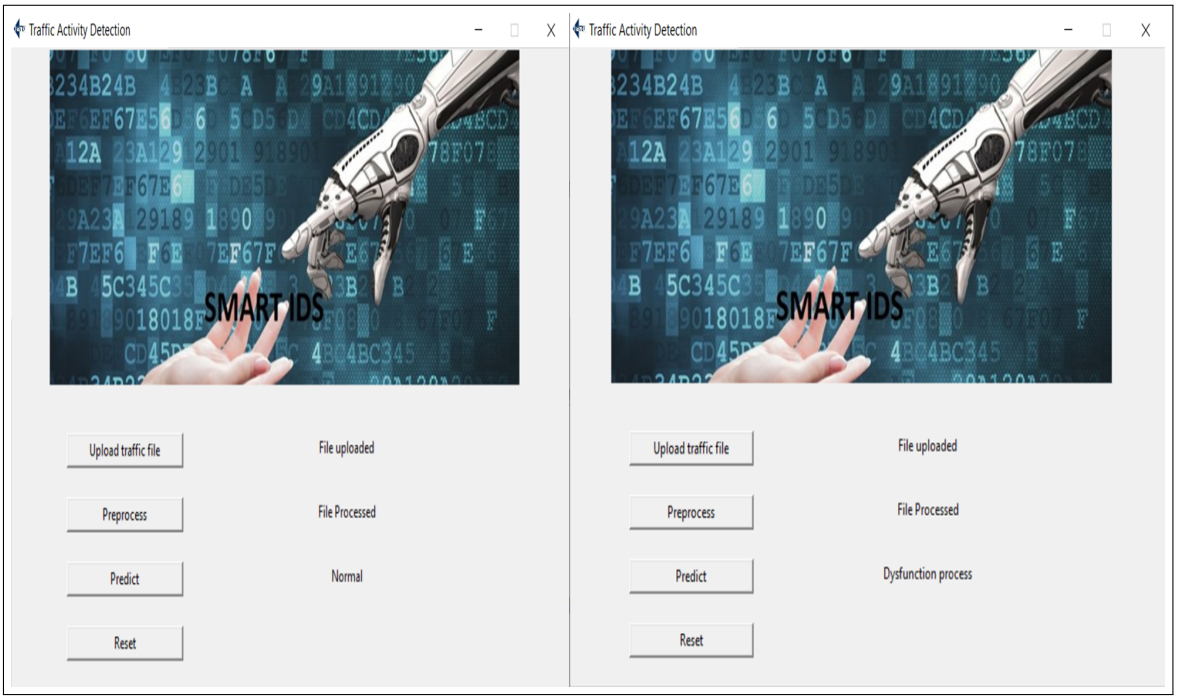


Figure 3.10 – Behavioral-based IDS graphical interface

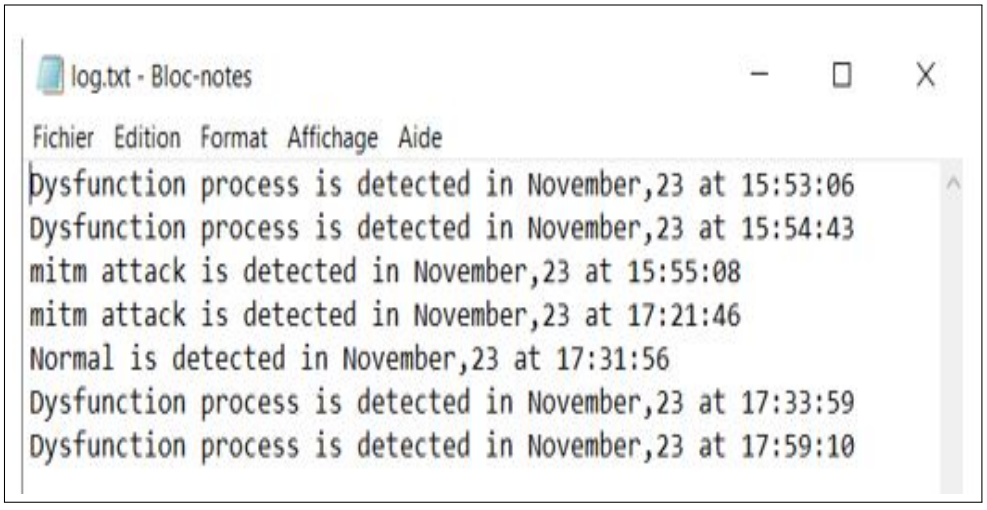


Figure 3.11 – Behavioral-based IDS log

After its prediction, the behavioral-based IDS provides logs (Figure 3.11) which will be used later in the last module of the proposed approach. The log file contains the nature of the detected intrusion and the date and time of the detection. Figure 3.11 gives an example of this log.

3.4.3 Neural network: Performance results

In the evaluation phase, one of the main concepts of the machine learning field which is a confusion matrix (Figure 3.12) has been used. A confusion matrix is a tool for measuring the performance of a machine learning model. It checks in particular how often its predictions are accurate, compared to reality. In its x-axis, true labels are presented and its coordinate axis shows the predicted labels. This confusion matrix is computed on testing dataset. According to these results, all traffic types are correctly predicted as shown in the diagonal line of this confusion matrix. The entire attacks have been classified with a high detection accuracy.

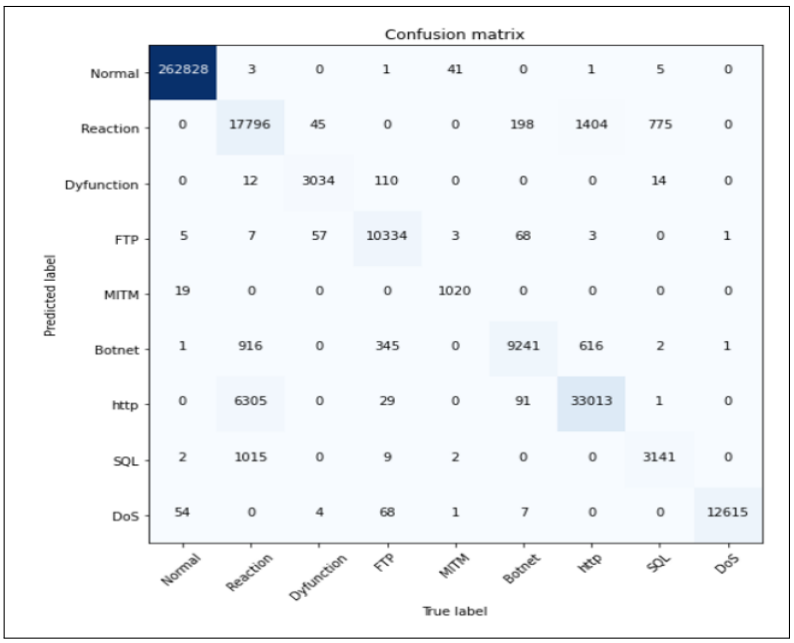


Figure 3.12 – Confusion matrix

To evaluate the IDS performance in more depth, three metrics have been used: precision, recall and F1-score whose formulas are given below:

1. Precision: a metric that gives the number of the correct computed positive predictions. It calculates the accuracy for the minority class. It is calculated as the ratio of correctly predicted positive examples, divided by the total number of positive examples that have been predicted.

$$Precision = \frac{True\ \ Positive}{True\ \ Positive + False\ \ Positive}$$

- True Positive: the truth is positive, and the test predicts are positive. There is an intrusion, and the test accurately reports this.
- False Positive: the truth is negative, but the test predicts are positive. There is no intrusion, but the test inaccurately reports that there has been one.

2. Recall: a metric that calculates the number of the correct computed positive predictions from all positive predictions that could have been made. It gives an indication of missed positive predictions.

$$Recall = \frac{True\ \ Positive}{True\ \ Positive + False\ \ Negative}$$

- False Negative: the truth is positive, but the test predicts are negative. There is an intrusion, but the test inaccurately reports that there has not been one.

3. F1-score: a metric that combines the previous two scores. It is the most used on imbalanced data, such as precision and recall, it has to be near to 1.0 for a perfect and best classification.

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

The neural network model has firstly been trained using the training dataset and tested later with the testing dataset. Training step has taken about 4 minutes. Once the training is done, the detection time is about few milliseconds. Other performance metrics have been used to measure the efficiency of the model: precision, recall and F1-score. Their results are good as shown in Figure 3.4:

as shown in Figure 3.4:

Table 3.4 – Neural network performance

Traffic type	Precision	Recall	F1-score
Normal	1.00	1.00	1.00
Reaction	0.68	0.88	0.77
Dysfunction	0.96	0.95	0.96
FTP	0.95	0.99	0.97

MITM	0.96	0.98	0.97
Botnet	0.96	0.83	0.89
HTTP	0.94	0.84	0.89
SQL	0.80	0.75	0.77
DoS	1.00	0.99	0.99

Normal and DoS traffic are correctly classified. For other types of traffic, precision is more than 90%. For reaction traffic precision is only 68%, this is probably due to incorrect labels. Reaction traffic was built from all other network traffic captures and was labelled manually according to our expert information. This task was not easy since some flows looked very similar and had to be distinguished from others. Consequently, some flows were mistakenly labelled as "Reaction". Otherwise, the other metrics are also good for almost the whole class.

To test and compare the neural network model to other machine learning algorithms, a quick comparison was performed regarding all the performance metrics for the following models: K-Nearest Neighbors (KNN), Linear Regression (LR), Naive Bayes (NB), Random Forest (RF), Support Vector Machine (SVM), Decision Tree (DT). Table 3.5 shows the results of this comparison. According to this comparison, some models are good in terms of accuracy but poor in terms of the other metrics such as DT, LR, NB except KNN which is good in term of accuracy, Recall and F1-score. ANN is the best one in terms of the whole performance metrics.

Table 3.5 – Comparative of machine learning algorithms performance

ML model	Accuracy	Precision	Recall	F1-score
KNN	0.93	0.80	0.82	0.93
LR	0.93	0.73	0.70	0.67
NB	0.90	0.68	0.64	0.62
RF	0.88	0.70	0.80	0.71
SVM	0.88	0.71	0.65	0.61
DT	0.87	0.70	0.67	0.67
ANN	0.91	0.92	0.92	0.92

3.5 Conclusions and discussion

In this chapter, a behavioral-based IDS is presented. This IDS analyses network traffic to detect intrusions in our industrial platform using a neural network model. The detection accuracy rate is high in addition to the different performance metrics. This IDS has correctly classified each attack.

Due to this IDS, it is possible to distinguish industrial faults from a real intrusion which is one of the difficulties that this type of IDS faces. However, this IDS has some weaknesses which consist of these three points:

- This IDS is unable to provide information about the features which have allowed the classification attack.
- This IDS uses 133 features which are not all useful for our intrusion detection. Therefore, the use of the right number of the suitable features could reduce the training time and improve the performance metrics.
- This IDS is unable to provide information about the attack impact on an industrial system.

Consequently, a study will be performed to determine the useful features. This is in addition to a reduction or selection technique study to determine which are the correct features to use.

To determine the impact of an attack on an industrial system, another type of anomaly-based IDS is used in the global approach. Chapter 4 presents a specification-based IDS which helps an operator to determine the impact of an attack on a system and to classify it into three categories: temporal, sequential or content.

Chapter 4

Specification-based IDS

4.1 Introduction

This chapter presents the second module of our global approach which is a specification-based IDS. The latter is the other sub-category of an anomaly-based IDS which purpose is to detect anomalies related to the industrial defaults and dysfunctions. The IDS aims to detect anomalies located at Manufacturing Executive System (MES) level. Its main basis is the MESA model provided by the ISA-95 standard. This model is a state machine that enables the planning, executing and tracing of a Production Order (PO). The use of this model brings another novelty in this detection field since it gives a reference model regarding the life cycle PO inside a factory.
MES fills the gap between Enterprise Resource Planning (ERP) and the automation level and it is based on a database in which all information related to a particular PO are saved and traced.
The idea of this specification-based IDS is to use this MESA model and the MES database data to compare the information regarding a planned PO with those of an executed one. A set of rules is established to detect the difference between both these PO. If a difference is detected, an alert is raised and logged in a log file for further investigation.
Using this specification-based IDS alone enables the detection of industrial faults inside a production line (see Section 4.4). These faults could be identified as temporal, sequential or content.
However, using the detection results of this specification-based IDS and those of the behavioral-based IDS, The nature of the anomaly can be determined and classified them into temporal, sequential or content intrusions. This discrimination follows an algorithm which helps to make an accurate decision.
In this chapter, a global view and the basis of the specification-based IDS are respectively given in Sections 4.2 and 4.3. Additionally, the identified anomalies in the industry are detailed and illustrated with an use case in Sections 4.4 and 4.5. Finally, the chapter is

ended by the IDS formalism, the technical specifications and the obtained results.

4.2 Specification-based IDS: Global view

4.2.1 Specification-based IDS principle

The main purpose of the IDS principle is process dysfunction detection. Based on the previously presented model, thirteen common anomalies were identified and are described in the next sub-section.To detect these anomalies, a set of rules was established allowing their detection which were then applied to the MES database. In addition to this dysfunction detection role, the specification-based IDS has one other role which consists of classifying the intrusions detected by the behavioral IDS into temporal and/or sequential.

4.2.2 Specification-based IDS assumptions

The approach of the IDS is valid while respecting several assumptions presented below:

- Industrial platform behavior is cyclic and deterministic.
- The proposed specification IDS targets the ISA95 compliant MES.
- The MES database is supposed to be accessible.
- Communications from the industrial control system are generally regular over time.
- PLCs are connected to the network.

4.2.3 Specification-based IDS: Motivations

The MESA model presented in 4.1 is a state machine which allows all the transactions to plan, execute and trace a Production Order (PO) through its performance metrics. This model lacks the security aspects that we have tried to add through this specification-based IDS. The thirteen anomalies listed in Table 4.1 focus on thirteen kinds of industrial dysfunctions (safety). Pairing these IDS detection results with behavioral-based IDS detection results (see Chapter 3), allows a reinforced and robust detection. Therefore, the thirteen anomalies could be qualified as intrusions (security) such as Distributed Denial of Service (DDoS), Man-In-The-Middle (MITM) and Brute force attacks as shown in 5 depending on the behavioral-based IDS detection results.

4.3 Specification-based IDS: basis

The main basis of the proposed specifications-based IDS is provided by the industrial ISA-95 standard. This basis is called the MESA model described in Sub-section 4.3.1. The MESA model manages the whole production line starting by the planning of the production order (PO) until the traceability of the latter in the MES database. For the entire industrial systems which are compliant with the ISA-95 standard, MESA model maps the whole schemas of their MES. In the following subsections, both these bases are highlighted specifically and are those from which our specification-based IDS was inspired.

4.3.1 The MESA Model

The MESA model defines a generic model of operational activities [Johnsson 2004]. It is applied to either production, maintenance, quality or stocks. It was particularly developed in the third part of the ISA-95 standard. The MESA model is a finite state machine that gives a succession of transactions through the various entities of the MES (see Figure 4.1). The Definition, Capability, Planning, Performance objects are the exchange objects level 4: ERP 1.3.

A Production Order (PO) originates from the ERP, then it goes through a detailed scheduling function which requires resources management. The latter contains information about available and engaged personnel, equipment, material and assets. Simultaneously, the definition management function is checked to obtain the product's requirements (recipes, procedures, ranges).

The detailed scheduling function contains all the planned information about the PO (planned start time, planned end time, planned PO duration, planned produced quantity...)

The PO is then dispatched and split by a dispatching management function into the Work Order (WO). Finally, the PO is sent sequentially or in parallel with other PO to the production execution function to be performed and executed. These last operations interact directly with the process control function which can be either manual or automated (supervision, control-command).

After the execution step, the data is collected, analysed and a performance report is sent to the ERP (right side of this model). The performance management function contains all the real information about the PO (actual start time, actual end time, actual PO duration, actual produced quantity...)

These planned and execution notions are the main basis of the proposed specification-based IDS since the latter will check that there is no difference between the planned and the actual executed PO by using the MES database.
MESA defines information flows by categories. The data structures are detailed in 8 models:

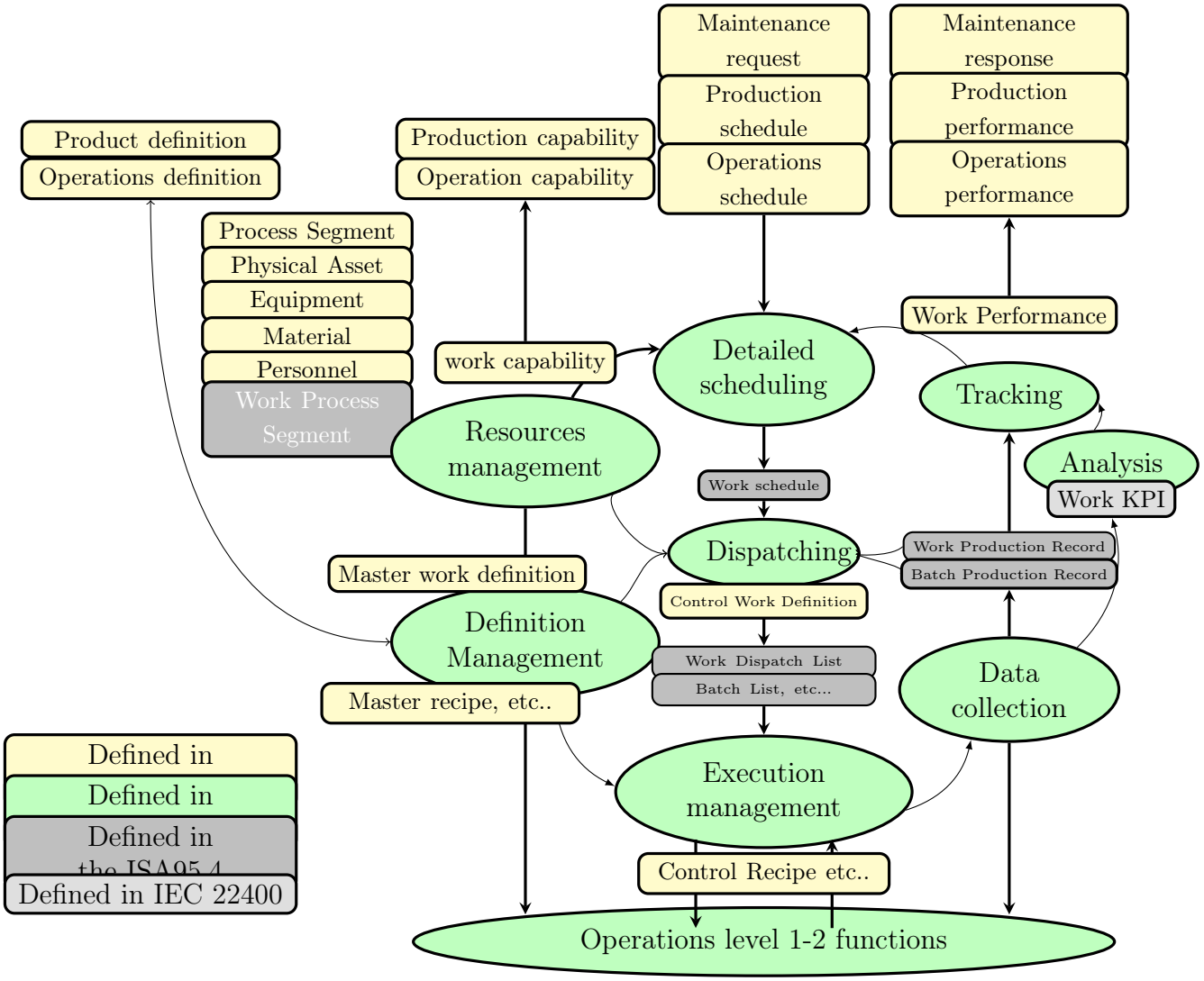
- 4 resource data models (personnel, equipment, materials and energy, process segment).
  - Personnel model: describes the personnel used to produce a PO (personnel information, skills and qualifications...)
  - Equipment model: describes the equipment used to produce a PO (equipment information, category of equipment, area and location...)
  - Material model: describes the material used to produce a PO (material information, reference, Unit of measurement...)
  - Energy model: describes the Energy used to produce a PO (Energy information...)
- 4 operational data models (production capability, product definition, production schedule and production performance).
  - Production capability model: provides all the requested resources (personnel, material, equipment,...) to plan and execute a PO.
  - Product definition model: specifies all the production recipes. It describes entirely a product, all the necessary steps to be performed and all the required resources.
  - Production schedule model: specifies what is needed to schedule a PO. After the scheduling step it provides the optimized schedule.
  - Production performance model: gives all the tracing information and the performance metrics regarding the execution of a PO.

The OPC Foundation, the ISA95 committee, and the MESA merged their efforts to make a new model which adds the ISA-95 object model representations of equipment, personnel, material, and physical assets to an OPC UA 95 specifications. The aim of this model is to show how the OPC Foundation, the ISA95 committee, and the MESA

standards can be used together in a federated system structure [Brandl 2016]. More details related to both the ISA-95 standard and the MESA model are given in Appendix A.

The MES database defines products, material, personnel and processes information. This information is implemented into the MESA model which generates data on a the physical level. In this new structure, every material is a data publisher and provides the selected information using standard exchange models. This model has been designed to take into account exhaustively, all the hazards that can occur in a factory.

Figure 4.1 – The MESA Model



4.3.2 MES database: tables

The Manufacturing Executive System (MES) is the main industrial information system that enables the planning, execution and management of a production order (PO). It manages four activities: production, maintenance, quality and inventory.
A MES which is compatible with the ISA95 standard must be based on the MESA model presented previously. To implement the latter, a Business to manufacturing markup language (B2MML) is used in order to provide provides an extended markup language (XML) schemas corresponding to the MESA model. These schemas are used to build the MES database. In this thesis, only XML schemas used to identify the thirteen anomalies detected by the specification-based IDS are presented below:

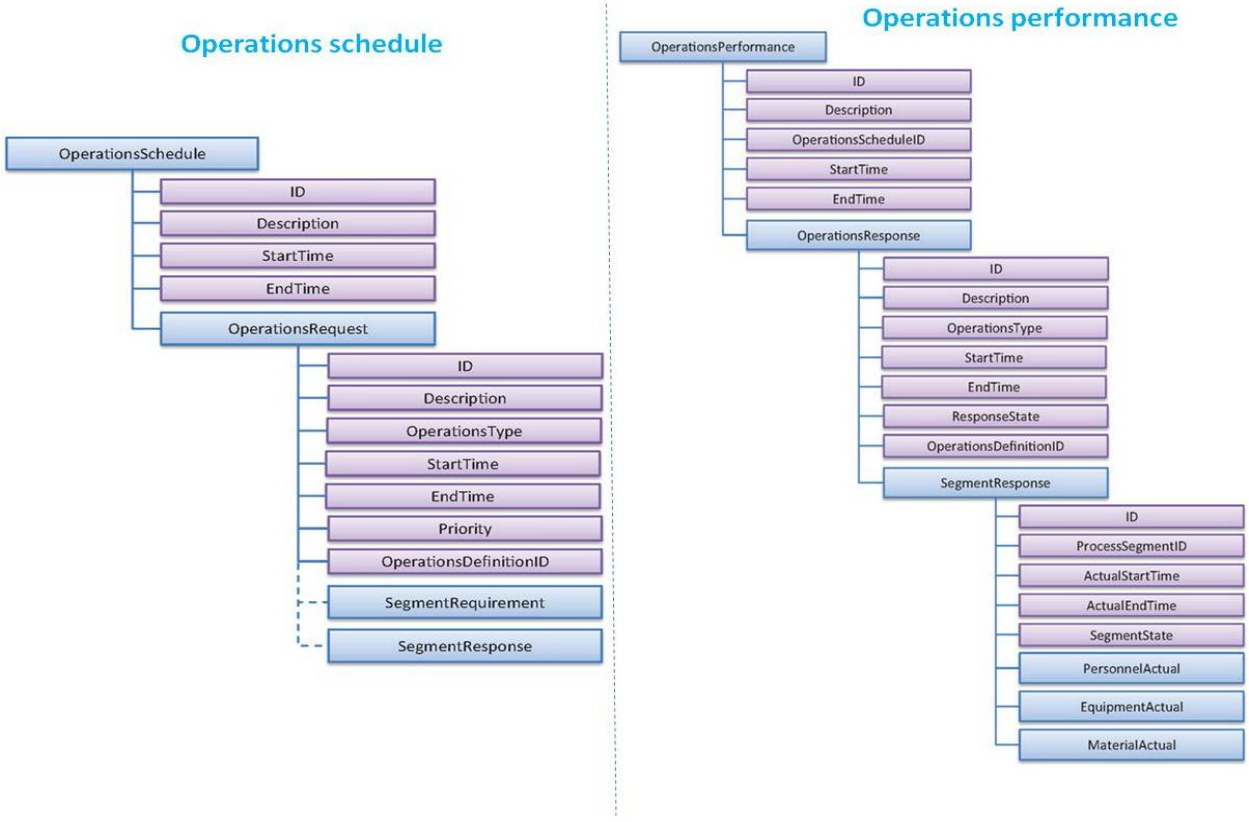


Figure 4.2 – Operation schedule and operation performance XML schemas

The operation schedule diagram (see left-hand side of Figure 4.2) gives all the necessary information related to a scheduled PO with a unique ID, start time, end time, description (optional), operation requested with a unique ID, segments requests.... After the PO planning, the optimized schedule is proposed. All of this information should be filled into the underlying segment with the right resources and all timing details to allow a proper dispatching operation.
The purpose of the operations performance is to follow-up and track the PO after planning, dispatching and executing it. This structure (see right-hand side of Figure 4.2) provides data regarding the actual information about delays? earlier execution PO, resources, timing, deviation time and duration. This diagram is also composed of a mandatory and optional fields. For a tracking purpose, this structure contains the same ID as the operation schedule. It also contains the actual start time, the actual end time, description (optional), operation response, segments response (start time, end time, state, etc.)
The operation definition structure (see left-hand side of Figure 4.3) provides a production recipe, that has a goal to define a product, the steps it must go through to be manufactured, the needed resources for each step... This structure is concerned with one recipe and it is composed of many segments. For each segment, it provides required resources regarding personnel, equipment, material, etc. This information also follows other structures that we exposed below.
The standard also provides schemas for resources. On the right-hand side of Figure 4.3, personnel structure is presented. It contains generic ID and an optional description, personclassproperty which could contain information regarding operators including names, addresses, dates of birth, etc. PersonnelClass groups together all individuals with the same skills such as mixer operators, moulders or testers.
In addition to personnel, equipment and material diagrams are described similarly using a unique ID to be identified clearly, description (optional), EquipmentClass and MaterialClass refers respectively to a class of equipment or material to which they belong.

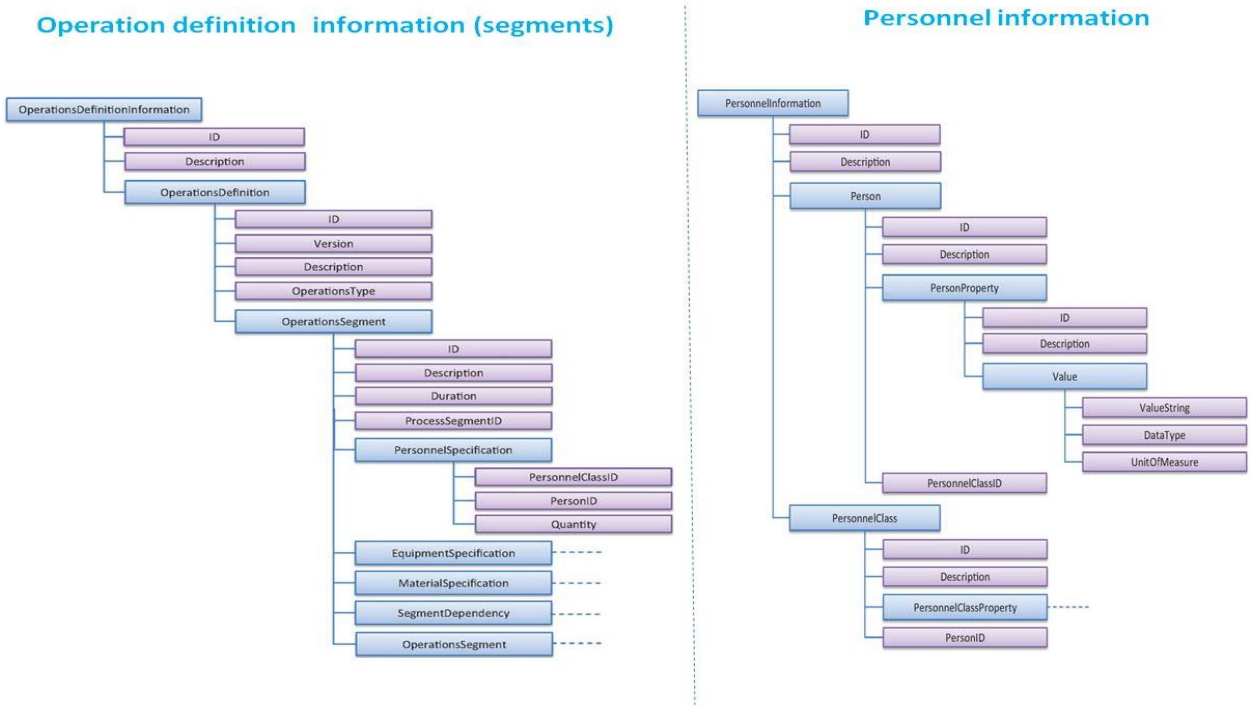


Figure 4.3 – Operation definition and personnel XML schemas

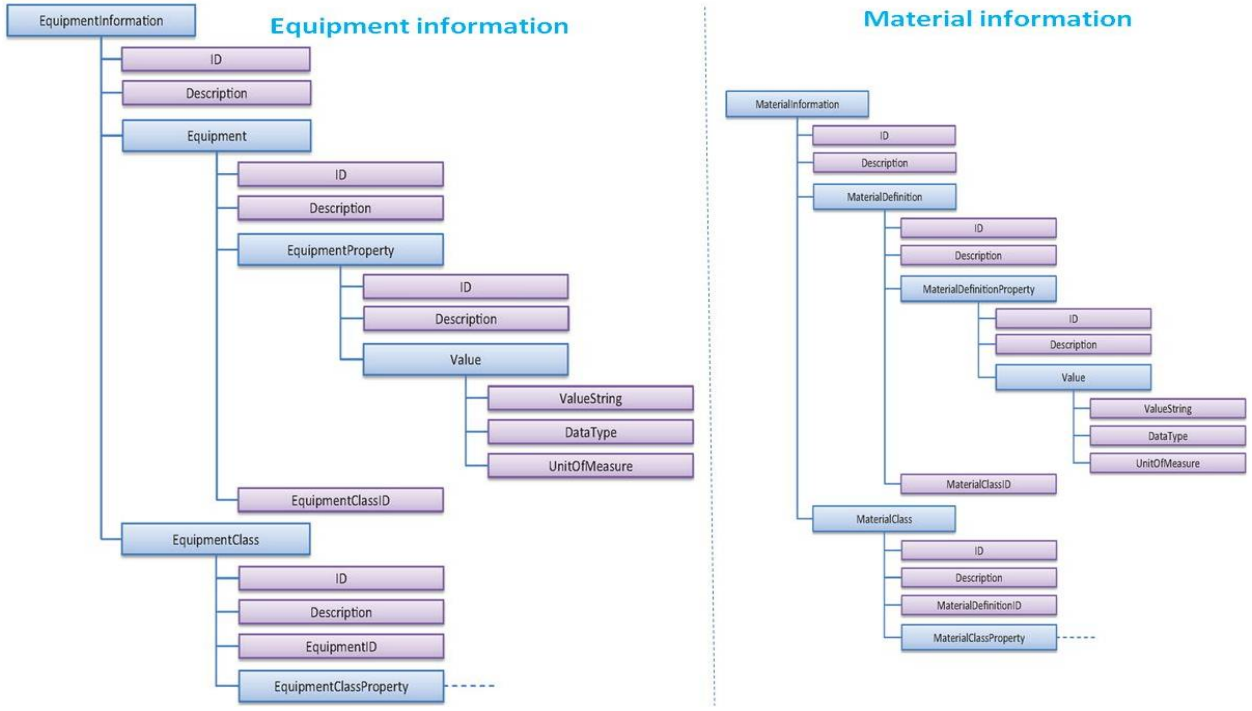


Figure 4.4 – Operation schedule and operation performance XML schemas

4.4 Identified anomalies in the industry

13 anomalies have been identified and are summarised in Figure 4.1. These anomalies are divided into temporal, sequential and content anomalies. As previously mentioned, the MESA model allows the planning and the execution of the tasks sequence. Therefore, thanks to this model property, the sequential anomalies are identified. To enrich this model, some KPIs (Key Performance Indicators) have been added to monitor the temporal aspect. These KPIs are provided by the ISO 22400 standard and they were chosen according to our research requirements.

4.4.1 Sequential anomalies

Among the 13 identified anomalies, we have sequential anomalies which detect non-respect of the execution sequencing. either for segments (work order: WO) or for production orders (PO).

- Compliance of the sequencing of segments execution: IDS checks that the order of the execution segments is respected.
- Compliance of the sequencing of the PO execution : IDS checks that the first planned PO is indeed executed first and the last one is executed at the end.

4.4.2 Temporal anomalies

Three temporal anomalies have been identified from the previously presented standard regarding our study requirements. These anomalies are summarized below:

- Compliance between the expected Actual Production Time (APT) and the APT actually consumed. APT is the actual time spent on a machine. For instance, if the production line is composed of four machines, four APTs are computed.
- Compliance between the planned Actual Transportation Time (ATT) and the Actual Transportation Time (ATT) which is actually consumed. ATT is the actual time spent between stations, such as assembling, separating, final checking and unloading time [Kang et al. 2016].
- Compliance between the planned or the expected Actual Order Execution Time (AOET) and the AOET actually consumed. AOET is the total time spent to execute an order, from the start to its completion on a machine.

To detect these anomalies with a high degree of accuracy, a statistical study was done to define a tolerable delta time that we can accept for our experimental platform. After computing the average and the standard deviation for every station, a delta of 10 seconds for APT and ATT and 90 seconds for AOET were accepted.
The advantage of this tolerated difference value is firstly, to avoid the false positive alerts and secondly, to make the IDS compatible with all industrial experimental platforms by adapting this value according to their requirements. The risk of this tolerated difference is that it can be modified by an attacker by using a modification of data attack. However, this implies that the attacker knows that the IDS tolerates a threshold and knows its value. The probability that all of these conditions are met remains low.

4.4.3 Content anomalies

In addition to temporal and sequential anomalies, attackers can also try to modify the data. It could be directly related to PO data such as the personnel, equipment or material used. It could also be related to equipment state or sufficient resources. All of these content anomalies are detailed below:

- The compliance between the segment request in the Planned Production Order (PPO) and the segment response in the Executed Production Order (EPO): the action is to check that the EPO response is the expected one i.e., the response is related to the PPO request and not forged by an attacker.
- Compliance between the personnel class in the PPO and the one in the EPO which checks that the staff used are the ones that were planned to be used.
- Compliance between the equipment class in the PPO and the one in the EPO which checks that the machine used is the one that was planned to be used.
- Compliance between the material class in the PPO and the one in the EPO which checks that the materials used are the ones that were planned to be used.
- Compliance between the expected produced quantity and the quantity that is actually produced.
- Equipment break-down control: IDS checks whether the equipment has broken down or is compromised whilst continuing to send data.

- Compliance of resources: IDS checks resources before launching the PO.
- Compliance between the PO launch and request: which checks if the request arrives while the PO is not launched.

The entire list of identified anomalies are detailed below:

Anomalies	Ref	Technique
1: Check that the response matches with the request segment	MESA	OP_ID (Planned) = OP_ID (Executed)
2: Check that the order of the segments is respected	MESA	StartTime (WO1) $\leq$ StartTime (WO2) $\leq$ StartTime (WO3) (planned) and StartTime (WO1) $\leq$ StartTime (WO2) $\leq$ StartTime (WO3) (executed)
3: Check the personnel skills planned/used	MESA	ID_PersonnelClass (planned) = ID_PersonnelClass (executed)
4: Check that the PO is executed with the right equipment	MESA	ID_EquipmentClass (planned) = ID_EquipmentClass (executed)
5: Check that the PO is executed with the correct material	MESA	ID_MaterialClass (planned) = ID_MaterialClass (executed)
6: Check that the total expected time (AOET) is correct	KPIs	AOET (planned) = AOET (executed) +/- 90 SEC
7: Check that the duration per segment (APT) is correct	KPIs	APT1 (planned) = APT1 (executed) +/- 10 SEC APT2 (planned) = APT2 (executed) +/- 10 SEC APT3 (planned) = APT3 (executed) +/- 10 SEC APT4(planned) = APT4 (executed) +/- 10 SEC

8: Check if times between segments (ATT) are correct	KPIs	Check if : ATT1 (planned) = ATT1 (executed) +/- 10 SEC ATT2 (planned) = ATT2 (executed) +/- 10 SEC ATT3 (planned) = ATT3 (executed) +/- 10 SEC ATT4 (planned) = ATT4 (executed) +/- 10 SEC
9: Check for overlapping between PO	MESA	Check PODateEnd (executed) (Previous PO)< PODatStart (executed) (next PO)
10: Check that the quantity requested is the one manufactured	MESA	Check if PQ (planned) = PQ (Executed)
11: Check if resources are available before launching the PO	MESA	Check if QuantitePersonnelReady > QuantitePersonnel (planned) and QuantiteEquipmentReady > QuantiteEquipment (planned) and QuantiteMaterialReady > QuantiteMaterial (planned)
12: Check the launching order of the PO	MESA	EndTime (Planned) (PO1) ≤ StartTime (Planned) (PO2), EndTime (Executed) (PO1) ≤ StartTime (Executed) (PO2)
13: Check if the equipment is down while continuing to send data	MESA	Check if PO_state (planned) = release and PO_state (executed)≠ release and PODateStart (executed) = 0 PODateEnd (executed) = 0

Table 4.1 – Anomalies identified from the MESA model and KPIs

4.4.4 Added metrics : ISO 22400 standard

Key Performance Indicators (KPIs) are used to determine the factors taken into account in order to measure the overall effectiveness of sales, a marketing device or a particular action. They are largely used in the management field. In this particular case, they have been used in the industrial management of a PO. We have used them to fill the gaps related to the temporal aspect of the MESA model which meet only the requirements related to the sequential aspect. Consequently, the anomalies or intrusions detection will

be more efficient by taking on consideration these temporal aspects.
The notion of KPIs was created by the International Standard Organization (ISO) which has developed ISO 22400. The latter was drafted and prepared by the technical committee ISO/TC184/SC 5, Automation systems and integration, Interoperability, integration and architectures of automation systems and applications. It was published in October, 2014 [Dennis Brandl 2016]. The ISO 22400 standard "Automation systems and integration — Key Performance Indicators (KPIs) for manufacturing operations management" is composed of four parts [Fukuda 2014] which are:

- Part 1: Overview, concepts and terminology
- Part 2: Definitions and descriptions of KPIs
- Part 3: Templates and categories of KPIs
- Part 4: Exchange and use of KPIs

This standard presents 34 KPIs to test and evaluate the efficiency, reliability and performance of a production line. These KPIs refer to work units, as it is specified in IEC 62264.
For each KPI, a formula, corresponding elements, tolerable values, unit of measure, timing and other characteristics are given. These KPIs are related to the activities described in IEC 62264: Production, Inventory, Maintenance or Quality. For each activity, it was decided to perform a set of KPIs by activity [MESA 2014]. The 34 KPIs are thus related to the third industrial level which is Manufacturing Operations Management (MOM) as it is described in IEC 62264. The proposed specification-based IDS targets this level in particular which justifies our use of this standard.

4.5 Anomalies illustration: use case

4.5.1 Context

the platform presented in Section 3.2 provides clarity on aforementioned anomalies. The example used illustrates the different identified anomalies and shows their impact on a production line. This example demonstrates the execution of a PO and consists of filling two vials with two balls and their capping through the platform Human Machine Interface (HMI) as shown in Figure 4.5.

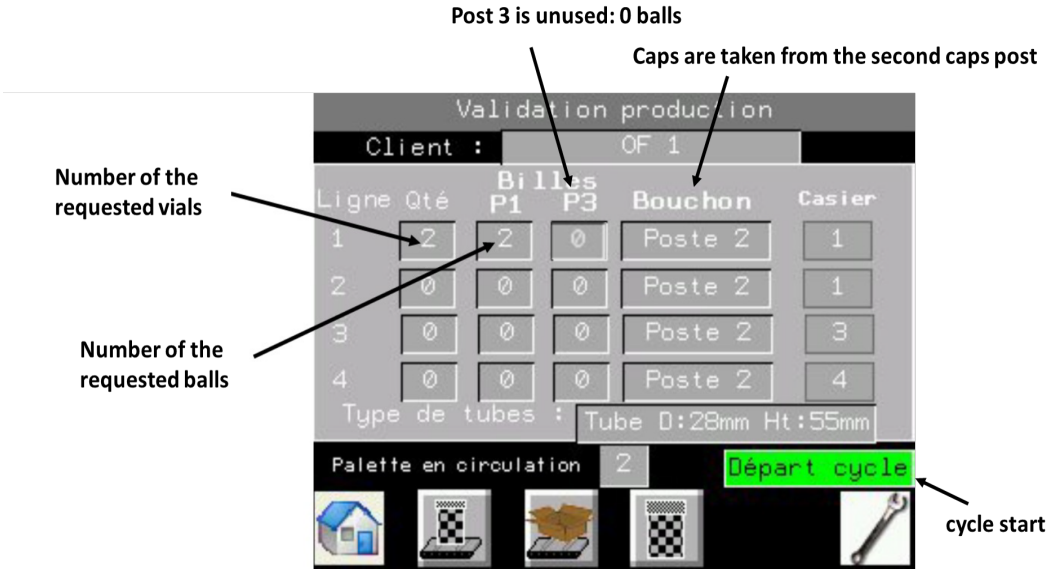


Figure 4.5 – Planned production order (PO)

To perform this PO, materials depicted in Figure 4.6 is needed: vials, caps and balls.

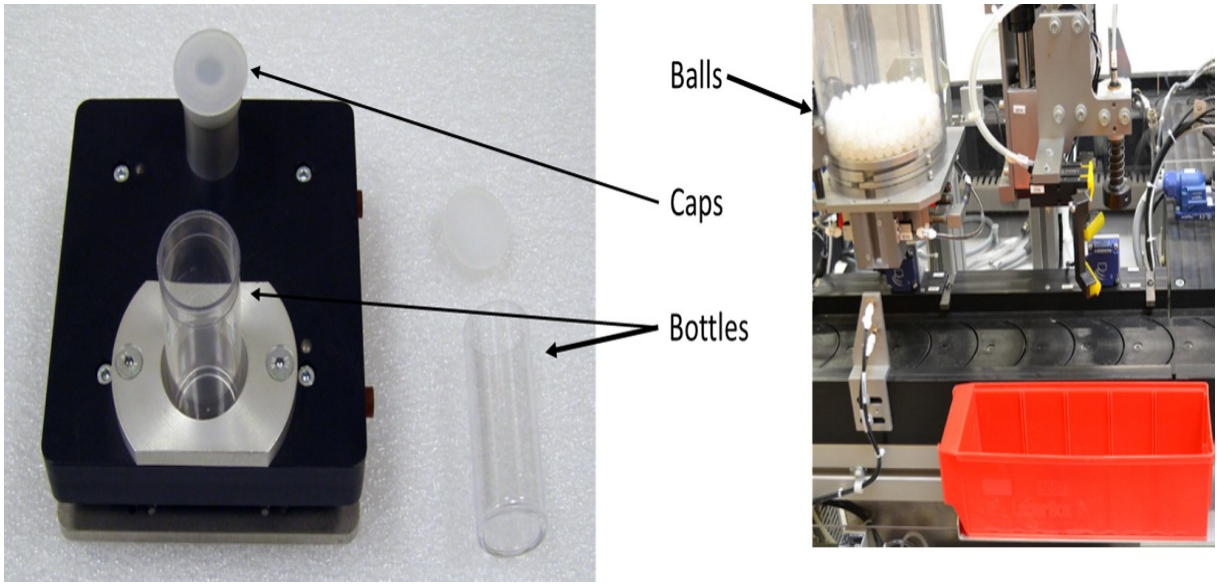


Figure 4.6 – Production order (PO) items

This PO is called an "operation" as it is used in the ISA-95 jargon. This operation is split into several segments as outlined below:

- Vials and caps loading on the pallet.

- Vials filling with balls.
- Vials capping with caps.
- Vials unloading.

After launching a PO through the HMI, a pneumatic arm loads one vial and one cap in the pallet, which then moves along the conveyor. Its position is known thanks to the Radio Frequency IDentification (RFID) card that the pallet carries and the different RFID card readers implemented on the conveyor.
The pallet arrives at the "vials filling" post where the vials filling operation starts. A counting system allows the planned number of balls to come down from the hopper into the vials. The pallet then continues on the capping post. Finally the pallet arrives at the unloading post where the filled vial is unloaded to the storage zone.

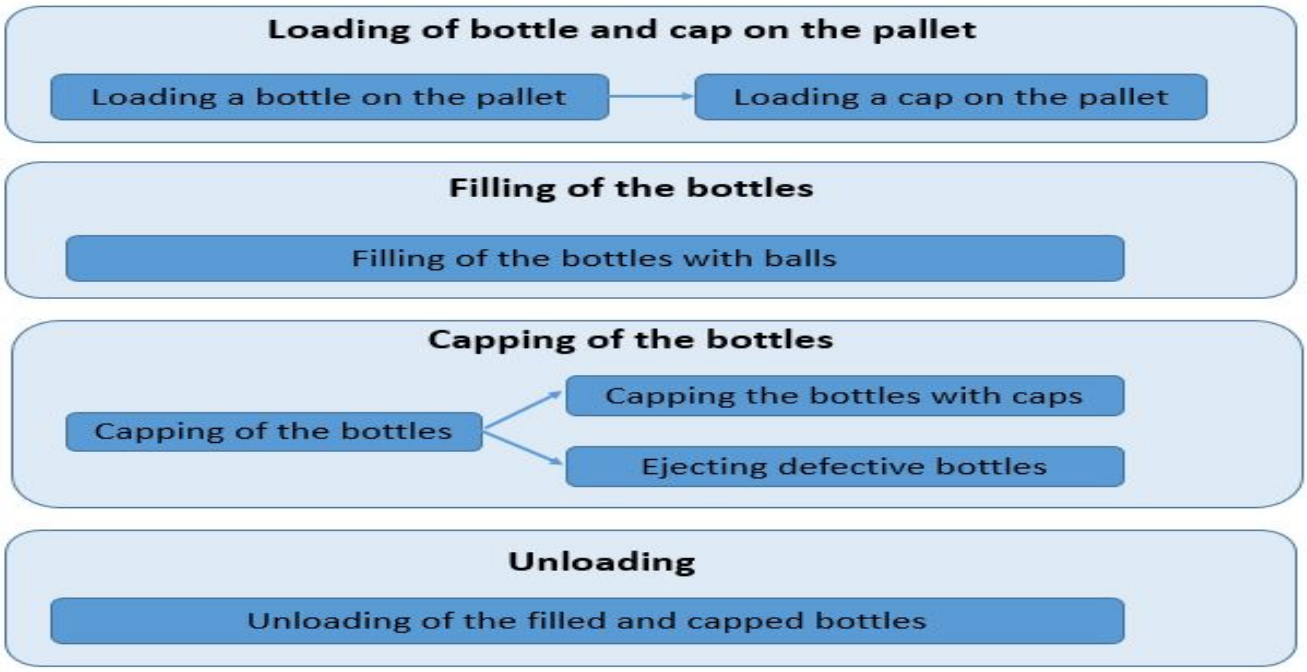


Figure 4.7 – PO segments and sub-segments

In accordance with the ISA-95 standard, "Operation" in our example refers to the filling of two vials with two balls and their capping in our example, this operation is split into many segments as shown in Figure 4.7.

These segments are divided into sub-segments as shown in Figure 4.8.

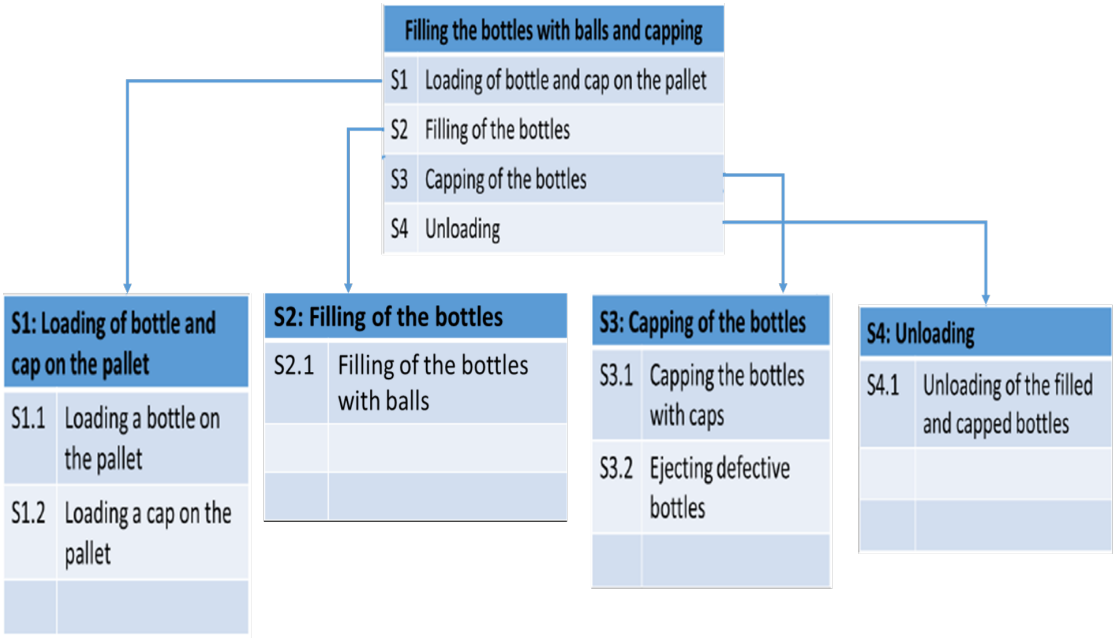


Figure 4.8 – Product segments structure

Figure 4.9 denotes the operations definition class diagram, outlining the system's classes, their attributes, operations, and the relationships among objects. One "Operation definition" corresponds to one recipe and could be composed of many segments. Each of these segments requires a number of staffs or personnel with specific skills, a set of equipment with defined specifications and capabilities, a quantity of raw material with specific requirements and some adjusted setting related to the PO to be correctly performed and executed.
Equipment, personnel and material follow specific models, as can be seen in the same figure.

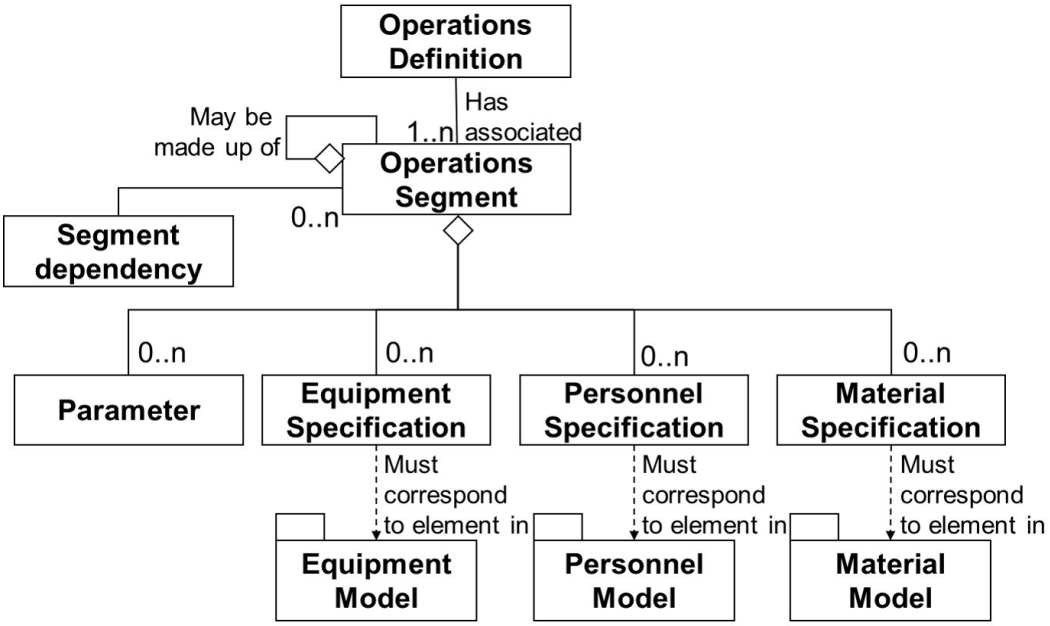


Figure 4.9 – Class diagram of operations definition

4.5.2 Use case

To highlight the importance of the thirteen identified anomalies, it is important to understand the meaning of each of them with an example, measure their impact on an industrial production line, and to be aware of the frequency of their occurrence.

Meaning and example	Industrial impact	Security impact	Frequency
Control 1: Allows to check that the executed PO corresponds to the planned one through the identifier (ID): For instance, the performance report established for filling two vials with two balls corresponds to the scheduled PO to fill two balls with four balls (See Figure 4.2)	Waste of time, Modification of recipe	Integrity	Medium

Control 2: Allows to check that all the operation segments are executed in the right order. For instance, "Capping" (S3) has to be executed after "Filling with balls" (S2) and the latter after "loading caps and vial" (S1) (See Figure 4.8)	Disruption to the production process, Waste of time and money	Integrity	High (With MITM attack)
Control 3: Allows to check that the PO is executed with the right personnel specifications. For example, it checks that the planning and the execution of the PO for filling two vials with two balls are done by a production operator and not a maintenance operator as demonstrated in Figure 4.9	Disruption to the industrial process, waste of time and money due to reprogramming the PO	Integrity and availability	Medium (With MITM attack)
Control 4: Allows to check that the PO is executed with the right equipment specifications. For example, capping the vial must be done with suction cup and not with evacuation clamp (Figure 4.7)	Disruption to the industrial process, waste of time and money due to reprogramming the PO	Integrity and availability	Medium (With MITM attack)
Control 5: Allows to check that the PO is executed with the right material specifications. For example, to fill vial with balls (Figure 4.7) the right balls color has to be used (this platform contains two balls colors: white and green)	Disruption to the industrial process, waste of time and money due to reprogramming the PO	Integrity and availability	Medium (With MITM attack)

Control 6: AOET corresponds to the total time spent to execute a PO. For instance, the AOET or total time to load or unload a vial	Production too long and too costly for industrialists	Integrity and availability	High (with DDoS or MITM attacks)
Control 7: APT corresponds to time spent in a workstation. For instance, APT for "loading" (S1.1), APT for "capping" (S3.1), APT for "Filling of the vials" (S2.1) (See Figures 4.7 and 4.8)	Production too long and too costly for industrialists and risk of overlapping between segments	Integrity and availability	High (with DDoS or MITM attacks)
Control 8: ATT corresponds to time spent between workstations. For instance, ATT spent between "Loading" (S1.1) and "Filling of the vials" (S2.1), ATT spent between "Filling of the vials" (S2.1) and "Capping" (S3.1)	Production too long and too costly for industrialists and risk of overlapping between segments	Integrity and availability	High (with DDoS or MITM attacks)
Control 9: Checks that the sequencing of the planned PO execution is respected.	Risk of overlapping between PO, waste of money	Availability	High (with with disrupting process or MITM attacks made in the experimentation Section)
Control 10: Checks that the planned quantity is produced. For instance, if the PO contains two vials to fill at the end of the production, two vials have to be filled up (no more and no less)	Risk of modification of the recipe, waste of time and money	Integrity and availability	High (with disrupting process, MITM or SQL force attacks made in the experimentation Section to modify the planned or manufactured quantity)

Control 11: Checks that the needed resources (personnel, equipment, material) are available. For instance, to perform a PO with this platform, one operator has to be available, with the available platform, using the required quantity of vials, caps and balls (as material)	Risk of overlapping between PO waiting for resources which was falsely declared as available	Integrity and availability	High (with MITM attack to launch the PO remotely or SQL force attacks to make resources available when in reality they are not)
Control 12: Checks that the first scheduled PO is finished before launching the next one	Risk of overlapping between PO, waste of money	Integrity and availability	High (with with disrupting process or MITM attacks made in the experimentation Section)
Control 13: Checks that the equipment is working when the performance report is sent. For instance, checks that when we receive performance metrics regarding " Filling" workstation (S2.1), it is indeed working	Disrupting process, waste of time, unreliable performance report	Integrity and availability	Medium (SQL force brute to modify data in the performance report)

Table 4.2 – 13 controls performed by specification-based IDS

4.6 IDS formalism

The principle of the IDS is to compare the scheduled PO with the actual executed one. This information is saved and traced in the MES database. By adding KPIs to this information, the database gives an inclusive view regarding the PO.
A set of rules is established to be used by the specification-based IDS in order to detect anomalies as it is mentioned in Figure 4.10. When an anomaly is detected, an alert is raised and logged.

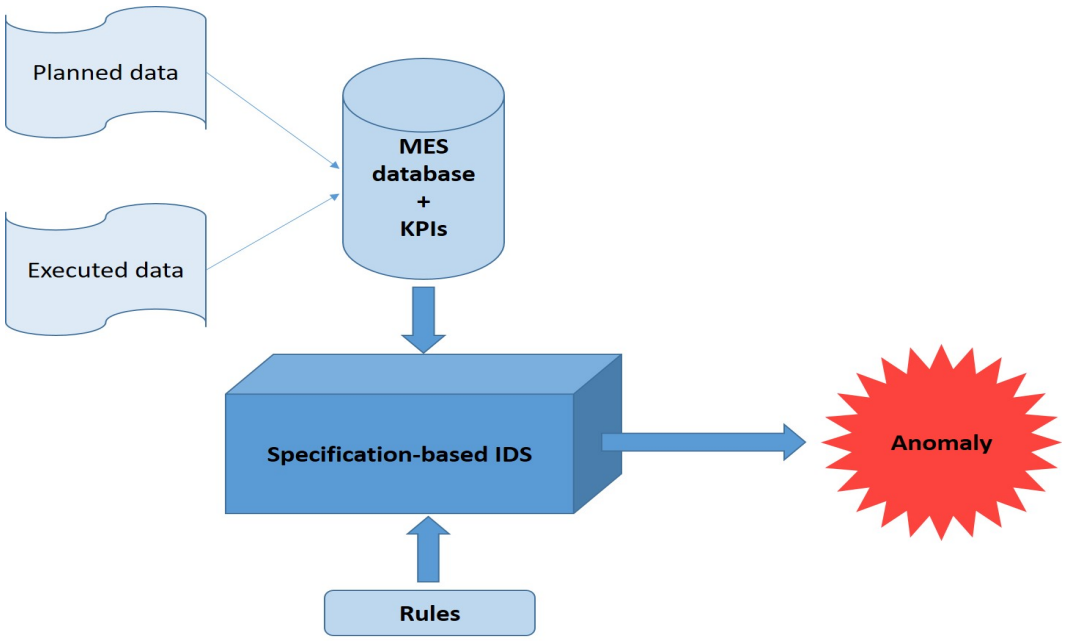


Figure 4.10 – Specification-based IDS formalism

4.7 IDS tool: Technical specifications

Specifications-based IDS is a GUI built in Windows Forms (WinForms) as rich client applications. It uses Framwework: .NET 4.5.3. It is developed in C# 6.0 using Visual Studio 2017. To handle Excel documents and databases, this GUI uses OpenXML library. It uses the MES database compliant with the ISA-95 standard. The database uses SQL Server 12 and is composed of 881 tables. It is unique for all of the modules to have a unique and a specific data warehouse. The purpose of this technique is to have the possibility to cross all its tables for a deep analysis. This database is designed so that the tables are not redundant.
In order to check whether there is a difference between the planned production order (PO) and the one that is actually executed. A traceability table named T_tracing was created from the different tables of our database (see figure below) using SQL joins between five tables which are described below:

1. A PO table (T_OF): contains some descriptive elements about the PO (planned start time, planned end time, real start time, real end time, etc).
2. A users or personnel table (T_OF_Users_Interm): contains some information about operators (Users ID, qualifications, etc).

3. A table for equipment (T_Enum_Ligne_Prod): contains some information about equipment (Equipment ID, produced quantity, etc).
4. A table for the production line (T_Ligne_Interm): contains some information about the production line (ID, name, capacity, state, stock zone ID...).
5. A table for the PO segments (T_Sequence_phases): contains some information about different PO segments (ID, name, state, planned start time, planned end time..).

The T_tracing table groups 35 fields together (see figure below) allowing the detection of the 13 previously determined anomalies.

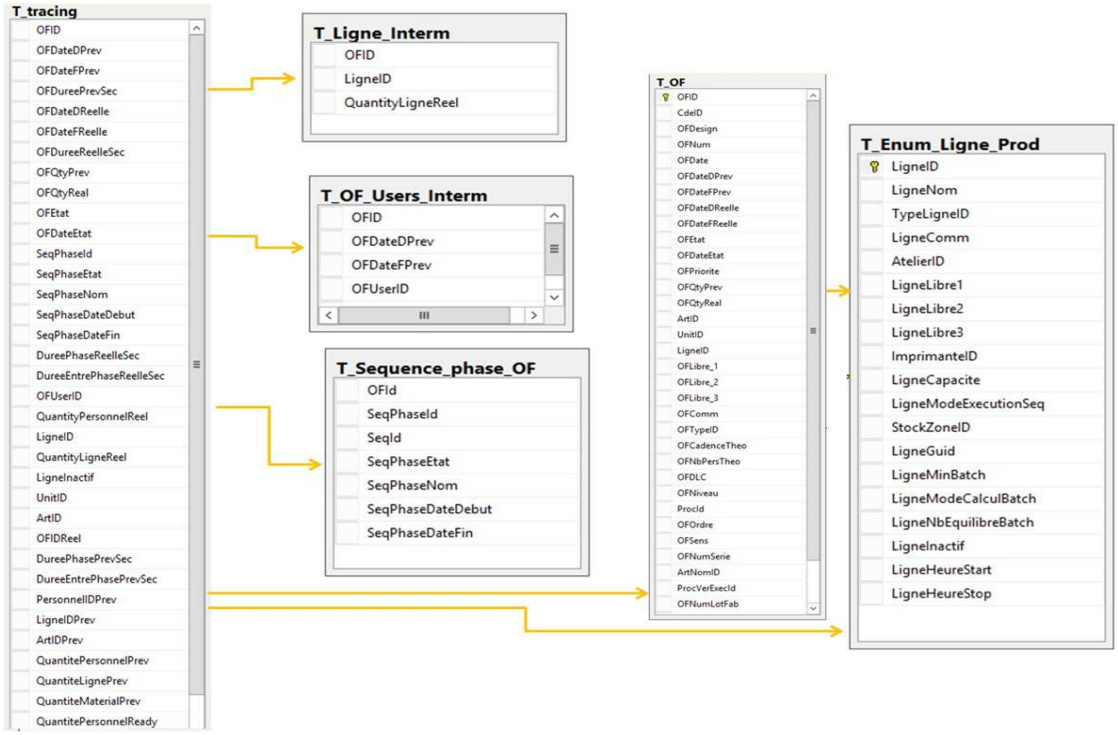


Figure 4.11 – Specification-based IDS tables basis

The IDS is designed to give more flexibility to the operator thanks to some functionalities which are listed below:

- It allows two checking modes: static mode using Excel files and dynamic mode via database connection.
- It allows for one or more anomaly checks.

- It allows one or more PO checks.
- It allows to reset logs for a daily checks.

After performing all the necessary checks, a log text file is produced for more investigation containing the nature of the anomaly, the related PO and some information about the detected anomaly. The same log file is used in the third module of the presented hybrid IDS which is a decision-making system.

4.8 Results

To test our specification-based IDS, a production cycle was launched without any process disruption or attack. The IDS detection results are shown in Figure 4.12 where all anomaly checks are performed without any violation.

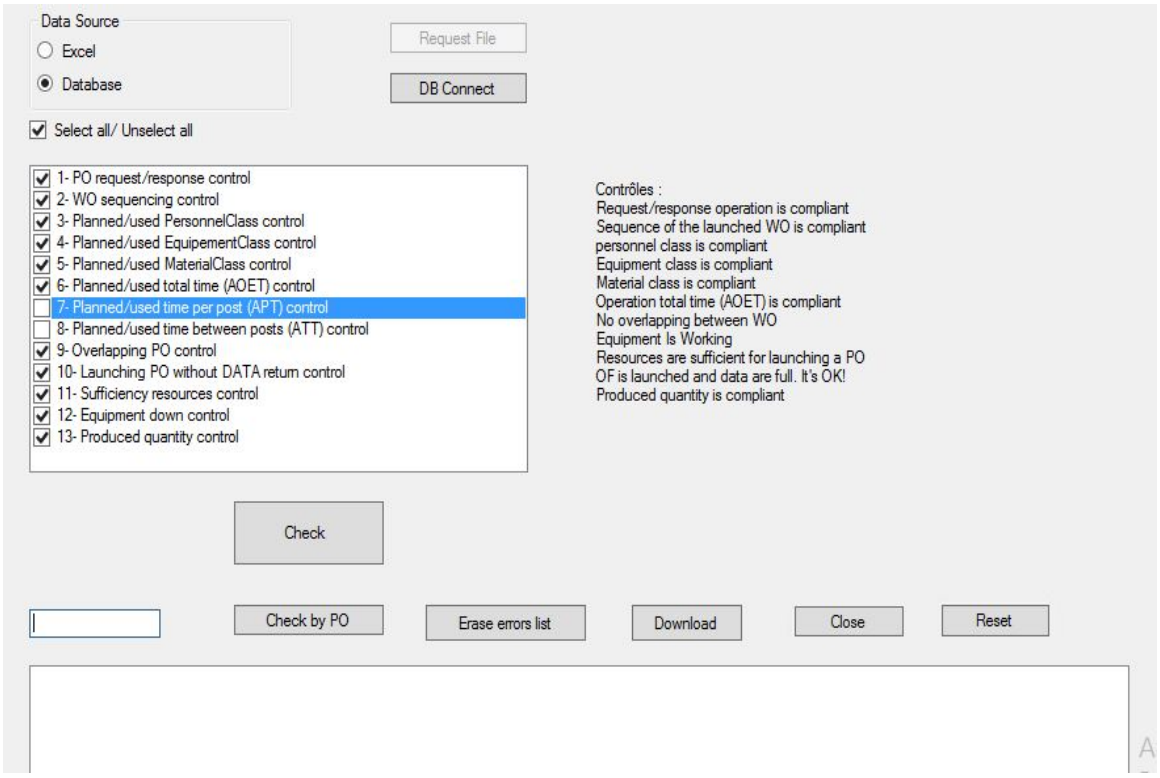


Figure 4.12 – Specifications-based IDS checking results: No anomalies

To test some temporal anomalies, an emergency stop has voluntarily been carried out causing a temporal delay. The latter was detected thanks to the anomalies 7 and 8. APT

and ATT metrics have exceeded the tolerated value. Consequently, an alert was raised (See Figure 4.13).

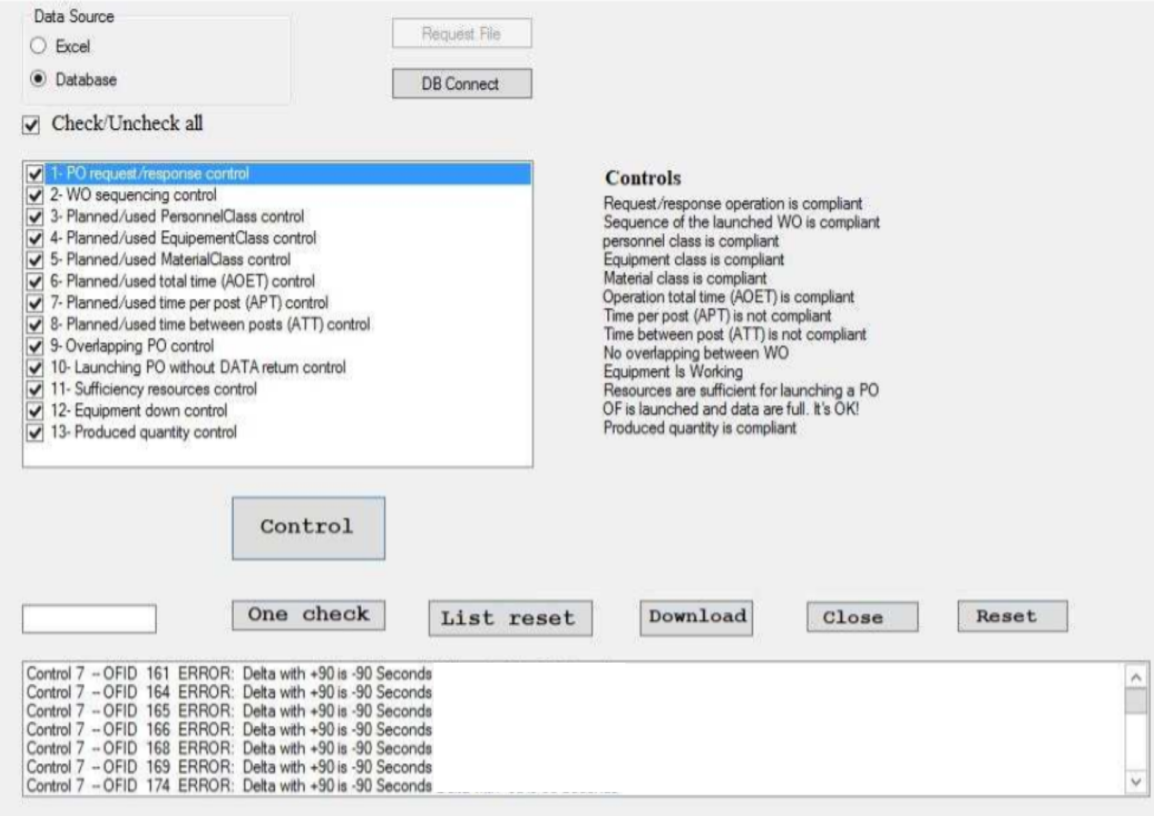


Figure 4.13 – Specifications-based IDS checking results: some detected anomalies

After detecting the previous anomalies, a log report was generated for more investigation. This report is verbose and provides the control which was violated, PO number causing a problem and the anomaly nature.

```
Control 10  -- OFID  161  ERROR:  LigneInactif = 1    OFDateFreelle = NULL
Control 10  -- OFID  181  ERROR:  LigneInactif = 1   DureePhaseReelleSec = NULL
Control 10  -- OFID  182  ERROR:  LigneInactif = 1    OFDateFreelle = NULL
Control 12  -- OFID  161  ERROR:       OFDateFreelle = NULL     OFQtyReal = 0,0000
Control 12  -- OFID  164  ERROR:      OFDureeReelleSec = 15/01/2020 13:45:05       OFQtyReal = 0,0000
Control 12  -- OFID  165  ERROR:      OFDureeReelleSec = 15/01/2020 16:11:49       OFQtyReal = 5,0000
Control 12  -- OFID  166  ERROR:      OFDureeReelleSec = 15/01/2020 16:12:39       OFQtyReal = 0,0000
Control 12  -- OFID  168  ERROR:      OFDureeReelleSec = 15/01/2020 16:12:25       OFQtyReal = 0,0000
Control 12  -- OFID  169  ERROR:      OFDureeReelleSec = 15/01/2020 16:15:37       OFQtyReal = 2,0000
Control 12  -- OFID  174  ERROR:      OFDureeReelleSec = 18/03/2020 15:17:53       OFQtyReal = 0,0000
Control 12  -- OFID  177  ERROR:      OFDureeReelleSec = 18/03/2020 15:18:04       OFQtyReal = 0,0000
Control 12  -- OFID  178  ERROR:      OFDureeReelleSec = 23/03/2020 20:40:56       OFQtyReal = 0,0000
Control 12  -- OFID  181  ERROR:      OFDureeReelleSec = 25/03/2020 14:37:19       OFQtyReal = 0,0000
Control 12  -- OFID  182  ERROR:       OFDateFreelle = NULL     OFQtyReal = 0,0000
```

Figure 4.14 – Specifications-based IDS logs report

After testing all the anomalies and checking all the controls, specification-based IDS has detected all the simulated scenarios with a high accuracy as result of the rules that were set. Detection time is about a few milliseconds which is an advantage for industrial systems. This approach is applicable to other complex automated systems. However, it is inconvenient for systems where the operator is often called upon. The operator interventions can generate false positives.

4.9 Conclusions and discussion

In this chapter, a novelty has been introduced in the industrial intrusion detection field. This novelty consists of using the MESA model as a reference, which traces the cycle life of a PO. From this model, a list of thirteen frequent anomalies has been identified. Other anomalies and rules could be added according to the MESA model in addition to the KPIs (Key Performance Indicator) that could be retrieved from a platform. In this thesis, only AOET, APT and ATT had be extracted but the ISO 22400 standard proposes 34 industrial KPIs that could be implemented.

The main purpose of the IDS is to detect attacks through their impact on an industrial system in order to help the operator execute an efficient reaction. Impacts of the attacks are classified into three categories which are : temporal, sequential and content. To reduce the false positive rate, an error margin has been added to the used rules. This margin of error is further advantageous in its adaptability of the IDS to other industrial systems by modifying the error margin according to their requirements.

Using the IDS alone only allows detection of industrial faults and not intrusion. Consequently, it has to be used in conjunction with the behavioral-based IDS presented in the previous chapter. This is the only weakness that the IDS has.

Chapter 5

BI-ANOmaly-based IDS: BIANO-IDS

5.1 Introduction

Today, cybersecurity community confirms that anomaly-based IDS are more efficient and more robust to detect sophisticated and advanced attacks. Unlike signature-based IDS, they can detect new threats because all events that differ from the normal behavior are considered as an attack. For this reason and in an attempt to fill the gaps observed in this field, this thesis proposes a new approach whose principle, components and more details are reported in this chapter.
In the previous chapters, two anomaly-based IDS were with their respective advantages and limitations, as summarized below:

	Advantages	Disadvantages
Behavioral-based IDS	• the discrimination between industrial faults and real attacks	• Discrimination with with a high false positives rate
Specification-based IDS	• Accurate since it retrieves data from the main central information system (MES) • Allows measurement of the attack impact (classification into many categories)	• It does not allow for the discrimination between industrial faults and real attacks

To take advantage of these two IDS and overcome their limitations, both IDS have been combined through a third component of the presented approach named the Decision Making System (DMS). This item is detailed in Section 5.4. Fortunately, as a result of this combination, the false positive rate has been reduced.
BI-ANOmaly-based IDS: BIANO-IDS is the global proposed approach. It is composed

of two anomaly-based IDS: behavioral-based IDS which was presented in Chapter 3 and specification-based IDS detailed in Chapter 4. In addition to these previous IDS, another component is proposed in this chapter. It consists of a decision making system (DMS). The purpose of the latter is to give the final decision regarding the detected anomaly by comparing the log files of the previously presented IDS. More details related to its function is given in the following sections.
This chapter begins with a global view and principle of BIANO-IDS followed by an outline of its components. Finally, the chapter is ended with an exploration of the DMS, as this is the system responsible for providing the nature of the detected anomaly. Notably, the DMS principle, its rules of function and its graphical interface and results will be discussed.

5.2 Approach: Global view and principle

BIANO-IDS is a BI-ANOmaly-based IDS. It aims to detect intrusions and anomalies of industrial production lines. For more efficiency, the hybridization notion is used in this approach by proposing two types of anomaly-based IDS as shown in Figure 5.1.
The main working principle of the BIANO-IDS is based on the detection results of its components which are:

- Behavioral-based IDS
- Specification-based IDS

The last item is the Decision Making System (DMS) which provides the decision regarding the nature of the detected anomaly.

BIANO-IDS takes as inputs the detection logs resulting from the two IDS presented in the previous chapters. These logs are processed and analysed by the DMS to make a decision about the nature of these anomalies as shown in the following figure:

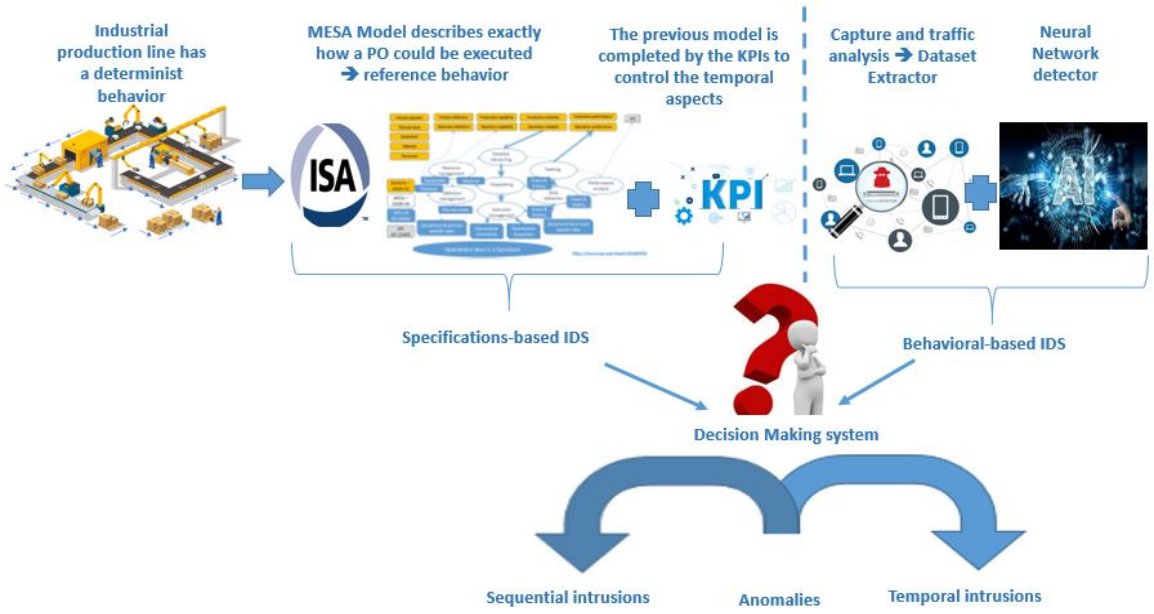


Figure 5.1 – BIANO-IDS global overview

5.3 Approach: BIANO-IDS components

The figure below gives an overview of the proposed hybrid IDS named BIANO-IDS and its components: Specification-based IDS, Behavioral-based IDS and DMS

1. Behavioral-based IDS: Assuming that most attacks pass through the network layer of the OSI model, this IDS analyses network traffic captured during the operation of the production line. It is based on the neural network which learns the nominal functioning of the industrial line, detects any suspicious activity and allocates it into an attack classification that it learned during its training phase.

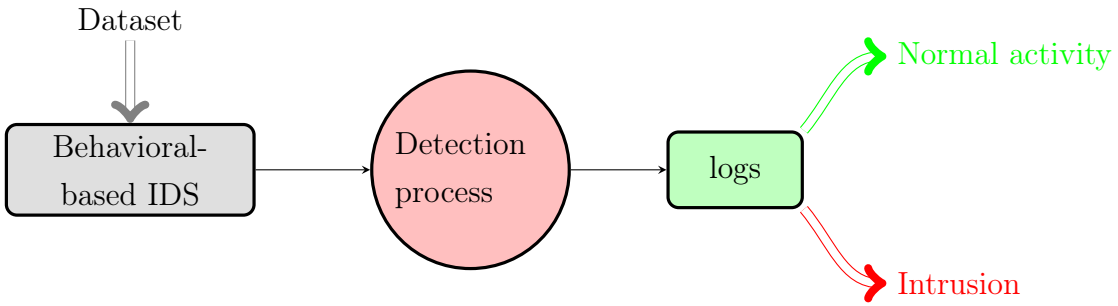


Figure 5.2 – Behavioral-based IDS principle

2. Specification-based IDS: Assuming that the behavior of an industrial line is deterministic and cyclical, the specification-based IDS analyses the production line operation with a focus on the planned and actual executed production order (PO). Its role is to check that there is no difference between the planned and executed PO according to the 13 identified rules from the MESA model as discussed in the previous chapter.

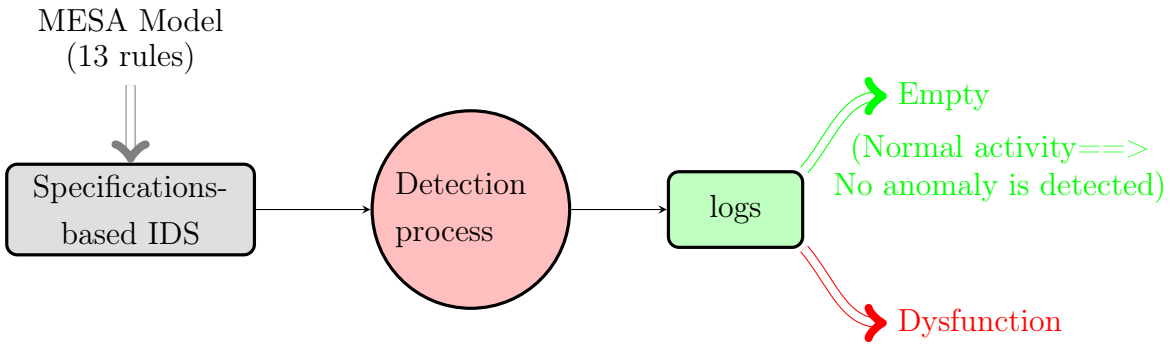


Figure 5.3 – Specification-based IDS principle

3. DMS: This is a decision-making system that aims to take advantage of the previous two components and overcome their limitations. It analyses the results logs of the two previous components to determine the nature of the anomaly as shown in Figure 5.9. More details regarding this third component are given in Section 5.4.

Consequently, BIANO-IDS works as a collaborative system and requires the results of all of the previously presented components in order to discriminate between an industrial fault and a real intrusion. The discrimination decision is given by the DMS as described in Figure 5.4.

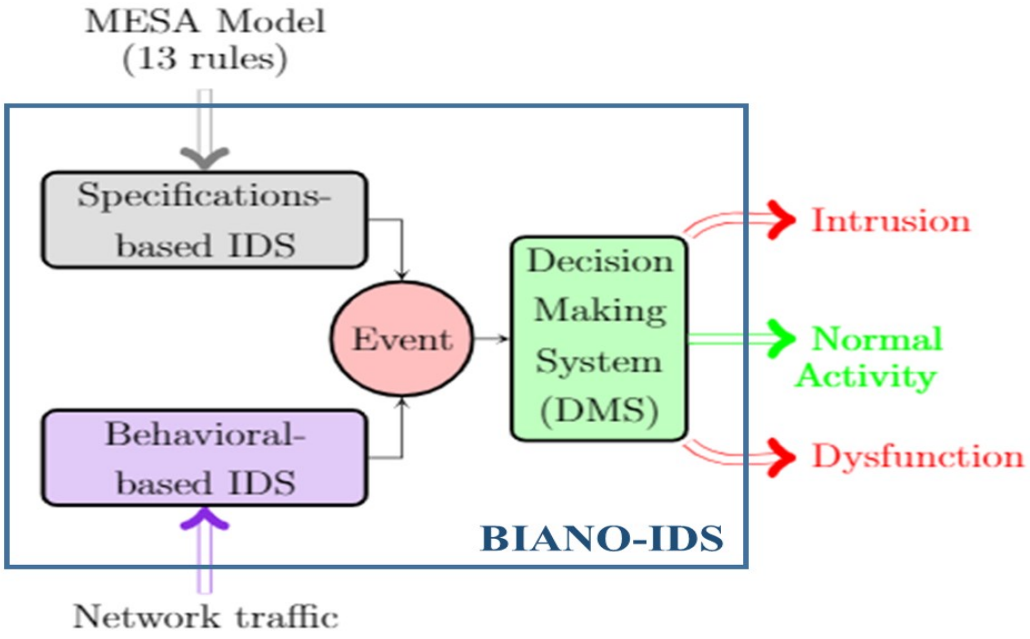


Figure 5.4 – BIANO-IDS principle

5.4 Decision Making System: DMS

5.4.1 Decision Making System: theory

5.4.1.1 Decision Making System (DMS): Definition

The Decision Making System (DMS) refers to the set of techniques used in choosing the best possible decision-making. Decision support is mainly used in areas such as finance, banking, IT, politics and crisis management.
It is useful to benefit from "simple" tools that allow us to quickly verify and analyse information in order to be able to make the most appropriate decision at any given time, without necessarily having an extensive knowledge of mathematics or computer science. The decision support tools aim to better choose among several solutions, according to established criteria, in a more transparent and more robust way.

5.4.1.2 Decision Making System (DMS): theoretical steps and markers

According to [Simon 1960], and depicted in Figure 5.5, a manager goes through four steps in order to make a decision:

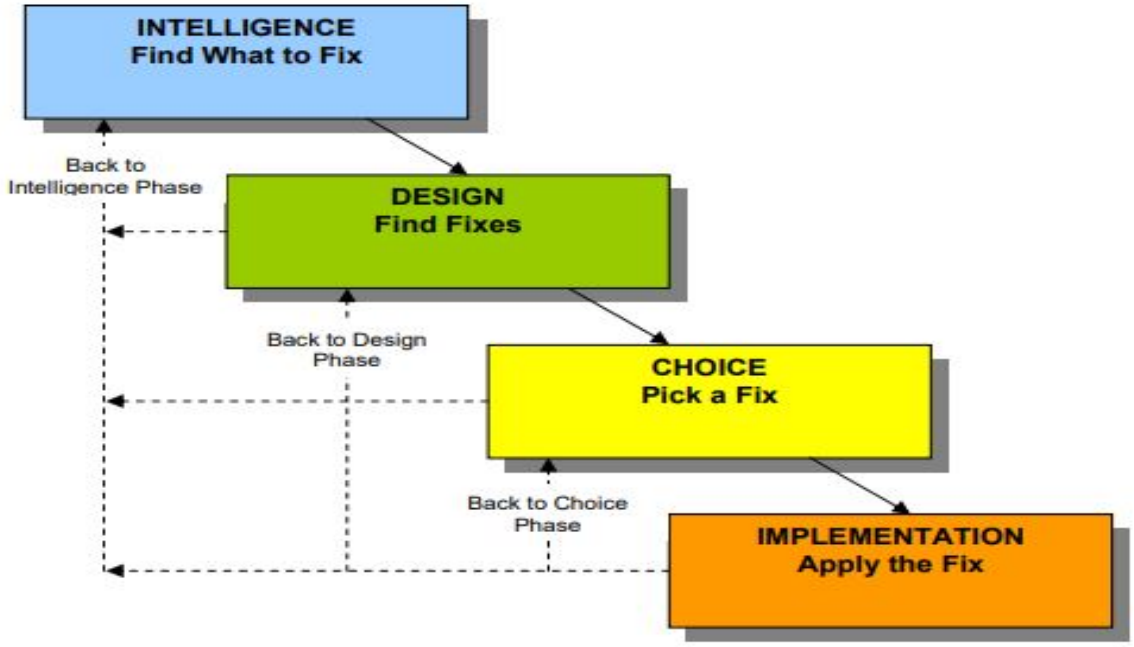


Figure 5.5 – Decision making system steps

Within the decision-making process, several types of decision markers appear. These markers have been classified by [Marakas 2003] as can be seen in Figure 5.6.
Two kinds of decision markers exist in this field, as summarised in Figure 5.6:

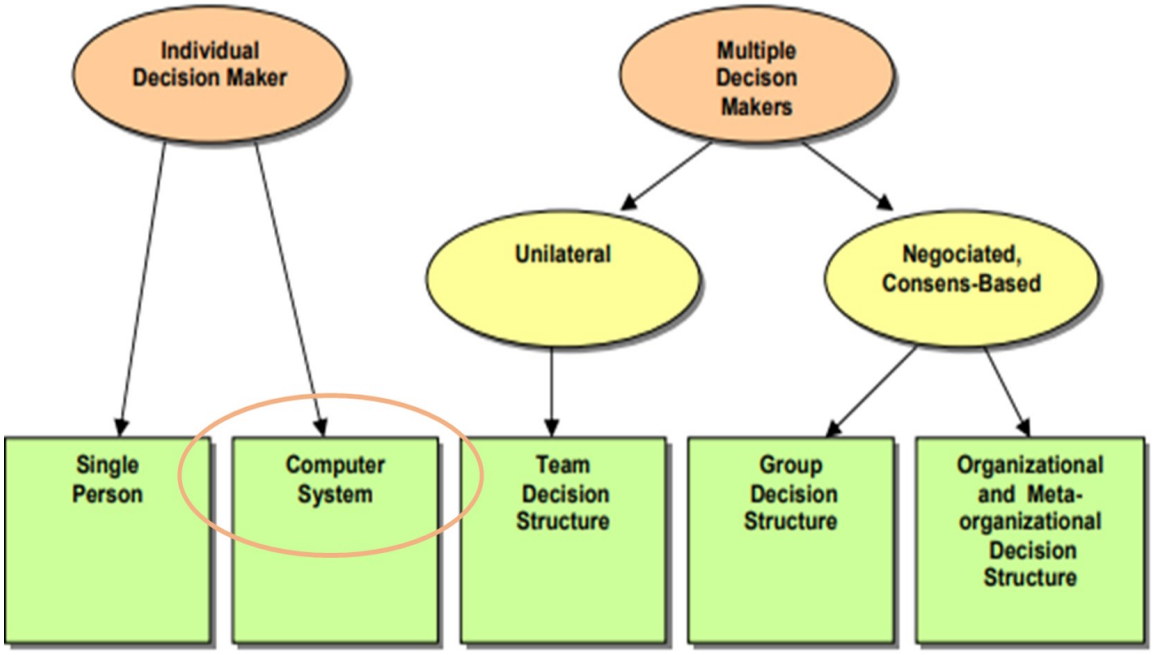


Figure 5.6 – Decision making system markers

- Individual Decision Markers
  - Single Person: A decision is made according to an expert knowledge.
  - Computer System: A decision is made by a computer applications which help to perform a diagnostic test and make a decision to solve the problem.
- Multiple Decision Markers
  - Team Decision structure: A decision is made by putting a group of individuals tasked to search for an optimal solution. The more participants involved in the decision, the more likely it is to be accepted and applied.
  - Group Decision Structure: A decision is made by a set of teams. These teams may have similar or different specialties.
  - Organizational and Meta-organizational Decision Structure: A decision is made by a set of groups that may belong to the same or different organizations. This is the top level in he decision-making process.

The decision-making system proposed in this approach is part of to Computer System category since it uses the logs result of the aforementioned IDS: Behavioral-based IDS and Specification-based IDS .

5.4.2 DMS: Model and global view

DMS aims to analyse the results logs of the behavioral-based and the specification-based IDS in order to determine the nature of the anomaly. The DMS prompts these logs to be processed and analysed in order to make a decision about the nature of these anomalies as shown in Figure 5.7.

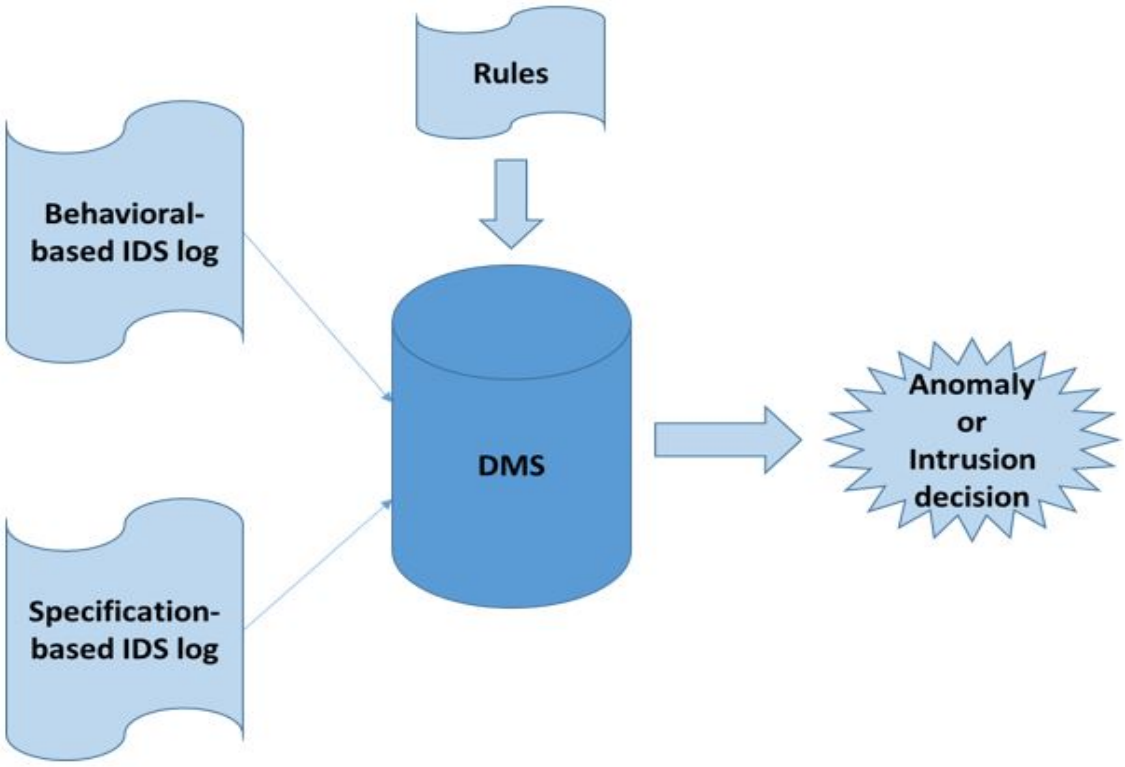


Figure 5.7 – Decision Making System (DMS) model

Above all, it is important to understand and distinguish the meaning of industrial fault or dysfunction from intrusion. In the detection intrusion field, both of these notions cause confusion. Therefore, it is important to explain their meanings as follows:

- Industrial dysfunction or fault: The state of an installation makes it unfit to perform a required function. This malfunction could be partial or total.

- Intrusion: An action which consists of accessing, without authorization, to the data of a computer system or of a network, by bypassing or defusing the implemented security devices.

More details regarding this distinction and how the DMS discriminates faults from intrusions and classifies them into other specific categories will be explored in subsequent sections of this chapter.

5.4.3 DMS: Principle and rules

There exist two types of DMS in the DMS field: Decisions are either made collaboratively, or independently. Since BIANO-IDS is composed of two kinds of IDS which work in parallel, it is logical that the DMS should work in a cooperative manner.
This module works according to many rules (explained in Figure 5.8), which allow us to determine the nature of the detected anomaly (fault or real intrusion).
As mentioned in the chapter related to behavioral-based IDS, the assumption that the majority of the attacks in the past were launched from the network. Therefore, according to the behavioral-based IDS (which analyses network traces) detection results, DMS checks the results logs, analyses them and determines if we are facing a simple fault or a real intrusion.

According to Figure 5.8, several decisions could be made. Before detailing them, it is important to understand the notations of the diagram 5.8. Letter "B" and "S" correspond respectively to the behavioral-based IDS log file and the specification-based IDS. In addition, numbers "0" and "1" show respectively the absence or the presence of the alerts in the log files. For instance, S1 means that the behavioral-based IDS log file contains alerts and B0 shows that the specification-based IDS log file contains no alerts. The potential decisions that a DMS could make are detailed below:

- If there are no alerts in neither the behavioral-based IDS (B0) nor the specification-based IDS (S0) log files ==> we are facing a normal activity.
- If there is an alert in behavioral-based IDS log file (B1) despite the fact that we have no alert in specification-based IDS (S0) log file ==> we are facing a real intrusion since most played attacks in industry went inevitably through network.
- If there is no alert in the behavioral-based IDS log file (B0) but an alert exists in the specification-based IDS (S1) ==> we are facing an industrial fault for the same reason as before.

- If there are alerts in both the behavioral-based IDS log file (B1) and in the specification-based IDS (S1) log files ==> we are facing a real intrusion.

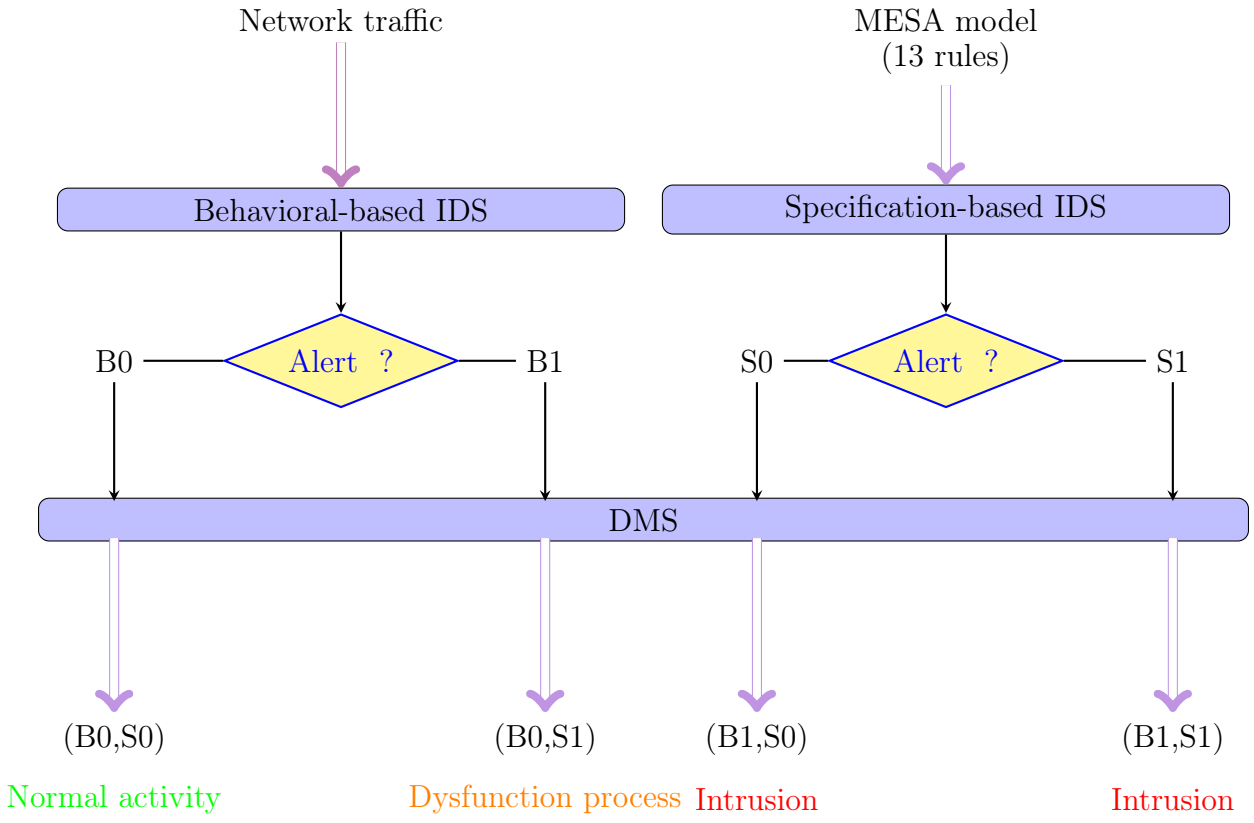


Figure 5.8 – DMS Rules

Legend:

B0: No alert in the behavioral-based IDS log
B1: Alert in the behavioral-based IDS log
S0: No alert in the specification-based IDS log
S1: Alert in the specification-based IDS log

5.4.4 DMS: Programming

The following code details the rules that the DMS must adhere in order to make a decision regarding the nature of the detected anomaly. It firstly defines two files: log1 and log2, which represent the behavioral-based IDS and the specification-based IDS log files respectively. Then, it reads their contents, which the code relies on in order to check the previously outlined rules. The term of "controls" corresponds to the anomalies that the specification-based IDS checks (the 13 anomalies presented in Chapter 4). Consequently, controls two and 12 are sequential. Controls six, seven and eight are temporal and the others controls denote the PO content.
More information about these controls are referenced in Table 4.2 in Chapter 4.

Algorithm 1: DMS code programming

```
Result:
log1 ← Behavioral − based IDS log;
log2 ← specification − based IDS log;
Read log1;
Read log2;
if "Normal" in log1 and log2 ←None then
  | print("Normal activity");
end
if "Normal" in log1 and controls [6, 7, 8] in log2 then
  | print("Temporal anomalies");
end
if "Normal" in log1 and controls [2,12] in log2 then
  | print("Sequential anomalies");
end
if "Normal" in log1 and controls [1, 3, 4, 5, 9, 10, 11, 13] in log2 then
  | print("Content anomalies");
end
if "Normal" in log1 and controls [1, 2, 3, 4, 5, 9, 10, 11, 12, 13] in log2 then
  | print("Sequential and content anomalies");
end
if "Normal" in log1 and controls [2, 6, 7, 8, 12] in log2 then
  | print("Temporal and sequential anomalies");
end
```

```
if "Attack" in log1 and controls [6, 7, 8] in log2 then
    print("Temporal attacks");
end
if "Attack" in log1 and controls [2, 12] in log2 then
    print("Sequential attacks");
end
if "Attack" in log1 and controls [1, 3, 4, 5, 9, 10, 11, 13] in log2 then
    print("Content attacks");
end
if "Attack" in log1 and controls [2, 12, 1, 3, 4, 5, 9, 10, 11, 13] in log2 then
    print("Sequential and content attacks");
end
if "Attack" in log1 and controls [6, 7, 8, 2, 12] in log2 then
    print("Temporal and sequential attacks");
end
if "Attack" in log1 and controls [1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13] in log2 then
    print("Temporal and content attacks");
end
```

5.4.5 DMS: alerts classification

According to the logs content, thirteen results detection (categories) are possible as mentioned in Figure 5.9. The previously presented decisions could be classified furthermore into either temporal or sequential or content alerts. By adopting the assumption that most attacks go through Tthe network, the DMS could make one decision among 13 possibilities, as summarized below:

- Normal activity: if both IDS logs are empty and contain no alert.

- Industrial dysfunction: if the behavioral-based IDS did not detect any intrusion (log file is empty) and the specification-based IDS has detected some alerts, the DMS decides that the detected anomaly is a dysfunction since no anomaly is detected through the network.

  - Temporal dysfunction: Faults due to a delay during the execution of the PO. For instance, the PO execution time is abnormally long
  - Sequential dysfunction: Faults due to the disrespect of the PO order. For instance, the risk of overlapping
  - Content dysfunction: Faults due to the content of the PO. For instance, the resource used to perform a PO
  - Temporal and sequential dysfunctions: Faults due to both temporal and sequential reasons.
  - Temporal and content dysfunctions: Faults due to both temporal and content PO reasons
  - Sequential and content dysfunctions: Faults due to both content PO and sequential reasons

- Real intrusion: if behavioral-based IDS detects some intrusions (log file contains alerts)

  - Temporal intrusion: Attack due to a delay during the execution of the PO. For instance, the PO execution time is abnormally long due to a DDoS or MITM attacks.
  - Sequential intrusion: Attack due to the disrespect of the PO order. For instance, the risk of overlapping due to a MITM attack
  - Content intrusion: Attack due to the content of the PO. For instance, the use of the wrong resources to perform a PO due to SQL injection attack
  - Temporal and sequential intrusions: Attacks due to both temporal and sequential reasons due to MITM and DDoS attacks
  - Temporal and content intrusions: Attacks due to both temporal and content PO reasons due to SQL injections and DDoS attacks
  - Sequential and content intrusions: Attacks due to both content PO and sequential reasons

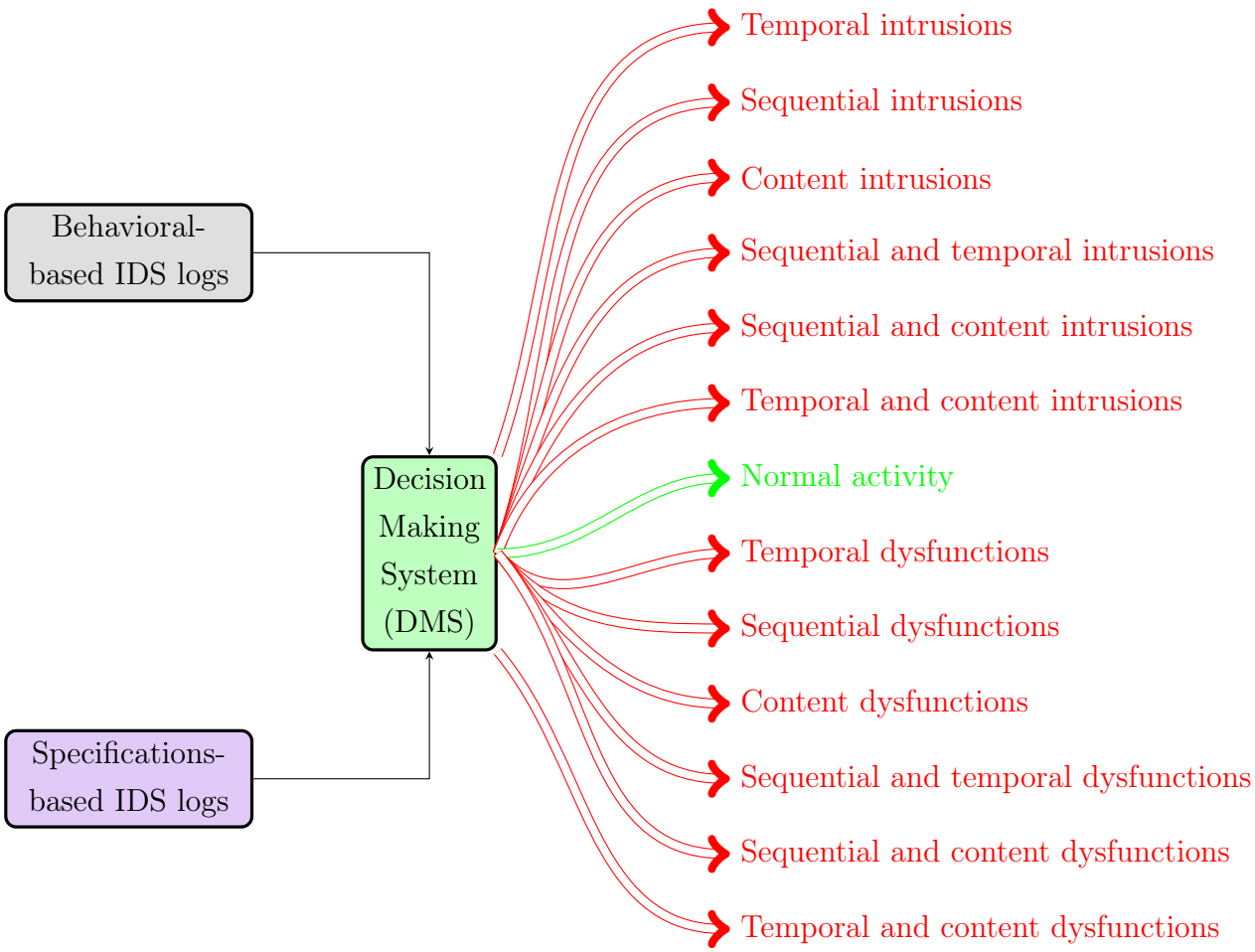


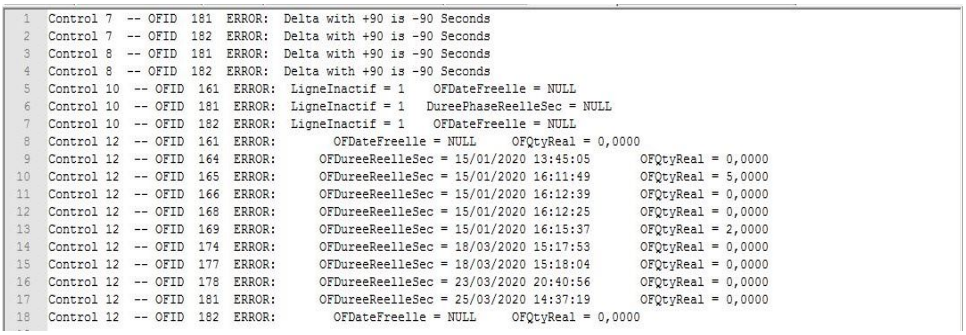
Figure 5.9 – Decision Making System (DMS) principle

5.5 Results

The DMS is developed in Python 3.8. It uses natural language prepossessing and data mining techniques. Its GUI is developed by using several libraries such as Tkinter (Figure 5.12). This GUI allows users to browse both log files of the behavioral-based and specification-based IDS. According to the established rules, the DMS provides either normal activity, intrusion or dysfunction as result. In addition, in case of intrusion or anomaly, its nature is also provided (temporal, sequential or content).

Figures 5.10 and 5.11 represent respectively examples of the behavioral-based and specification-based IDS log files.

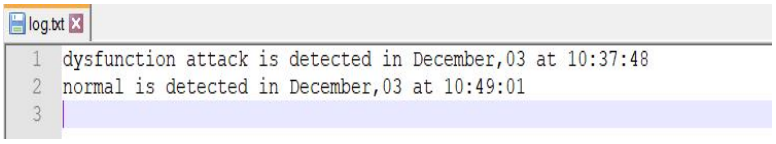
The specification-based IDS log file contains the number of the checked anomaly, the number of the abnormal PO with more details related to the nature of the anomaly.



```
Control 7  -- OFID  181  ERROR:  Delta with +90 is -90 Seconds
Control 7  -- OFID  182  ERROR:  Delta with +90 is -90 Seconds
Control 8  -- OFID  181  ERROR:  Delta with +90 is -90 Seconds
Control 8  -- OFID  182  ERROR:  Delta with +90 is -90 Seconds
Control 10  -- OFID  161  ERROR:  LigneInactif = 1    OFDateFreelle = NULL
Control 10  -- OFID  181  ERROR:  LigneInactif = 1    DureePhaseReelleSec = NULL
Control 10  -- OFID  182  ERROR:  LigneInactif = 1    OFDateFreelle = NULL
Control 12  -- OFID  161  ERROR:        OFDateFreelle = NULL     OFQtyReal = 0,0000
Control 12  -- OFID  164  ERROR:      OFDureeReelleSec = 15/01/2020 13:45:05       OFQtyReal = 0,0000
Control 12  -- OFID  165  ERROR:      OFDureeReelleSec = 15/01/2020 16:11:49       OFQtyReal = 5,0000
Control 12  -- OFID  166  ERROR:      OFDureeReelleSec = 15/01/2020 16:12:39       OFQtyReal = 0,0000
Control 12  -- OFID  168  ERROR:      OFDureeReelleSec = 15/01/2020 16:12:25       OFQtyReal = 0,0000
Control 12  -- OFID  169  ERROR:      OFDureeReelleSec = 15/01/2020 16:15:37       OFQtyReal = 2,0000
Control 12  -- OFID  174  ERROR:      OFDureeReelleSec = 18/03/2020 15:17:53       OFQtyReal = 0,0000
Control 12  -- OFID  177  ERROR:      OFDureeReelleSec = 18/03/2020 15:18:04       OFQtyReal = 0,0000
Control 12  -- OFID  178  ERROR:      OFDureeReelleSec = 23/03/2020 20:40:56       OFQtyReal = 0,0000
Control 12  -- OFID  181  ERROR:      OFDureeReelleSec = 25/03/2020 14:37:19       OFQtyReal = 0,0000
Control 12  -- OFID  182  ERROR:        OFDateFreelle = NULL     OFQtyReal = 0,0000
```

Figure 5.10 – An example of the specification-based IDS log file

The behavioral-based IDS log file contains the nature of the detected attack with some information related to the time when the attack was detected.



log.txt

```
1  dysfunction attack is detected in December,03 at 10:37:48
2  normal is detected in December,03 at 10:49:01
3
```

Figure 5.11 – An example of the behavioral-based IDS log file

In Figure 5.12, a log file containing the detected alerts by the behavioral-based IDS and another one with the detected temporal and content anomalies (See chapter 4) by the specification-based IDS are uploaded. Therefore, as shown in this figure, the detection is accurate and temporal and content intrusions are detected.

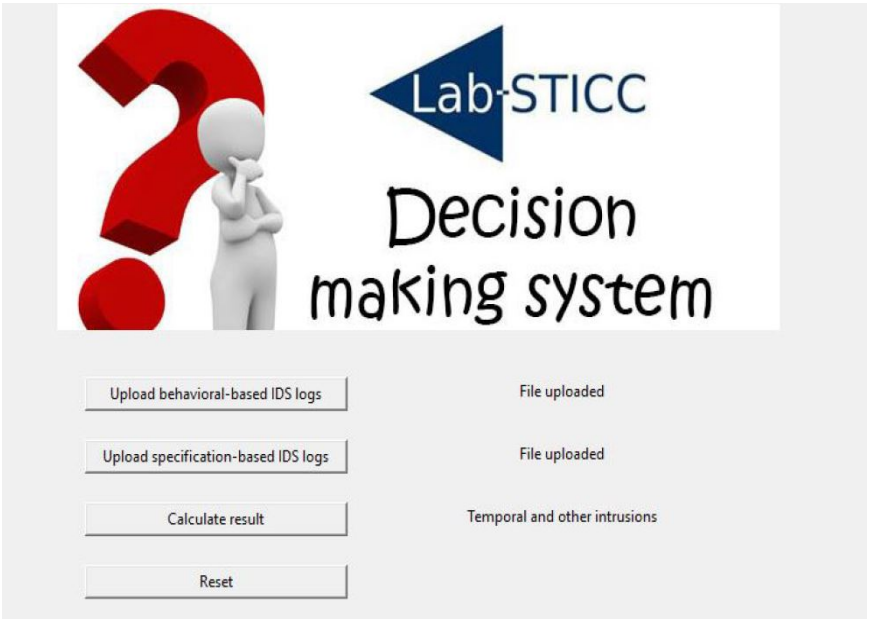


Figure 5.12 – Decision Making System (DMS): Temporal and other intrusions

To further test the detection accuracy of our DMS, a log file containing the detected alerts with behavioral-based IDS and a log file with the detected sequential and content anomalies are used. Consequently, the results are also accurate as depicted in Figure 5.13. This analysis is made according to the same temporal basis. Detection time is about a few milliseconds which is advantageous for a better reaction by the operator.

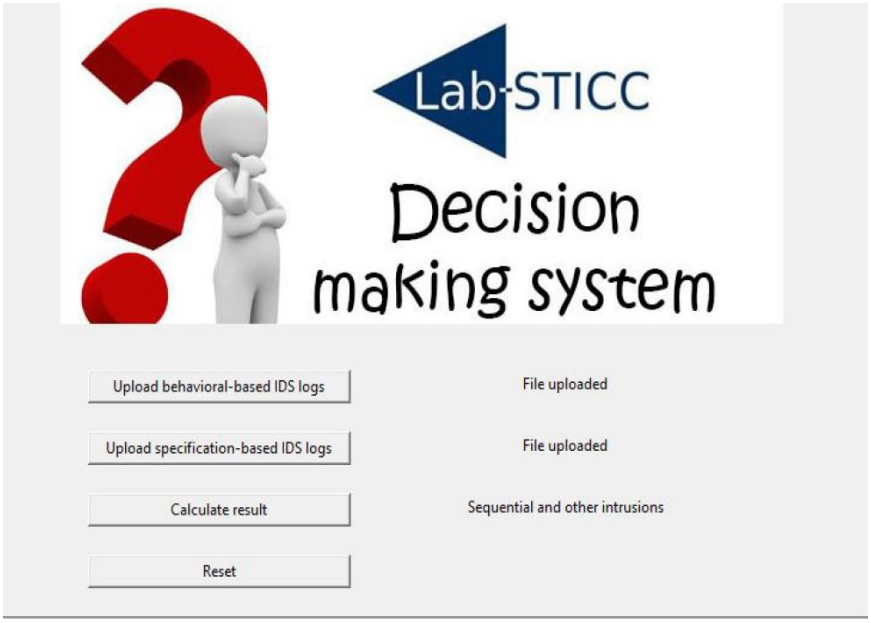


Figure 5.13 – Decision Making System (DMS): Sequential and other intrusions

5.6 Conclusions and discussion

BIANO-IDS is a new intrusion detection approach. In this chapter, the global approach has been presented with all its components and results. These results are provided by a third module called the DMS. The DMS analyses the intrusion detection log files of behavioral-based and specification-based IDS to distinguish industrial faults from real intrusion.

Thanks to this approach and the hybridization of two types of anomaly-based IDS, the false positive rate has been reduced.

All BIANO-IDS components have to operate in a cooperative way to obtain a better detection efficiency and a good classification. This point can be considered simultaneously as an advantage and a limitation. In addition to this limitation,it is important to keep in mind the limitations presented previously, from which each components of BIANA-IDS suffers if they are used separately. To overcome these limitations, all BIANO-IDS components should work together.

In future works, the false positive and false negative rates could be improved by exploring the syslog of a PLC to correlate all events. In perspectives, we intend to develop and investigate the possibility to improve this approach and obtain more information on all events which occur inside a PLC. Another potential research avenue could be to apply some IA techniques on these syslog traces in order to analyse the PLC behavior.

Chapter 6

Conclusions and perspectives

6.1 Summary

Today's industry is now, more than ever, in the focus of hackers. For this reason, it is time to protect factories with a robust technique that can withstand and combat the increasingly sophisticated attacks. Among the defence mechanisms, anomaly-based IDS remains the best way to ensure a defense in depth.
In this thesis, a bi-anomaly-based IDS called BIANO-IDS is proposed. This approach is composed of a behavioral-based and a specification-based IDS. The first one (see Chapter 3) is based on a neural network from which it learns normal behavioral of our platform and considers any sort of diversion from it as intrusions.
The evaluation metrics are good and all attacks have been classified correctly according to the confusion matrix. The results have been compared with other models and the results have been concluded regarding the suitability of a neural network for this study.
To test the proposed behavioral-based IDS, a dataset has been captured and built. The latter is considered as one of the novelties of this work. It is a labelled dataset with 9 labels, captured in a real industrial platform, containing 133 features, some of them are related to the transport layers and others are related to the Modbus/TCP protocol. This dataset has filled a real gap in the datasets field.
For both of the IDS, every detected intrusions are logged in a log files. These logs files are used by a final module called Decision-Making System (DMS), and serves as an input to decide the nature of the detected anomaly.

6.2 Contribution

Despite researcher's attempts to fill them, many gaps remain in the industrial intrusion detection field. This thesis brings new contributions to the industrial cybersecurity field,

as summarised below:

- Proposal of a new intrusion detection approach using two sub-categories of anomaly-based IDS. Thanks to this new approach, the discrimination of industrial dysfunction from real intrusion is possible using the neural network technique and MESA model as a reference model.

- Enrichment of the MESA model with a cybersecurity layer. The MESA model is a state machine describing how a PO could be scheduled and planned, how it could be executed and tracked later for a decision making by industrial actors. However in this model, the aspects of cybersecurity is missing, and its exchanged data is not secured. This thesis proposes a new method of introducing aspects of cybersecurity into these models. This is possible by proposing a specification-based IDS which allows the detection of a list of anomalies. This list is not exhaustive and could be enriched with other anomalies specific to industrial platforms. This IDS could be used in any industrial platform compliant with ISA-95 thanks to the added margin error which is an adaptive parameter.

- In the intrusion detection field, many researchers have proposed detection approaches using machine learning algorithms which requires a reliable dataset in order to be tested. During our state of the art study and analysis, a real lack of reliable industrial datasets has been noted. Therefore, in this thesis a behavioral-based IDS is proposed using a neural network algorithm which was trained and tested by our own dataset: an industrial dataset, containing 133 features that is labelled and captured in a real industrial platform. With this dataset, we have contributed to filling a real gap in the datasets field. This dataset is related to Modbus/TCP protocol but the approach is extendable to other industrial protocols such as OPC-UA and MQTT as it is explained in Chapter 3.

6.3 Limitations

The proposed approach brings many contributions to the field of industrial intrusion detection. However, it suffers from some limitations which are summarised below:

- Features classification: Behavioral-based IDS has correctly classified every attack. This IDS has been trained thanks to a dataset which is composed of 133 features.

However, the proposed approach could not shed light on the features which are responsible for this classification.

- System retraining: The proposed behavioral IDS uses a neural network as a model. This type of algorithm requires a training phase. For any addition, deletion or modification of data, behavioral-based IDS has to be retrained accordingly.
- IDS complementarity: For a high level of efficiency, the behavioral-based and specification-based IDS have to be used in parallel. Using behavioral-based IDS alone allows only the detection of the intrusion and does not allow to measure the impact of an attack on a system. Furthermore, using specification-based IDS alone only allows detection of industrial fault since most attacks are played through the network. Therefore, to detect attacks with accuracy and reliability, the two proposed IDS have to be used in a complementary way.

6.4 Perspectives and future works

During the research and execution of this thesis, I was inspired by several ideas and I was curious to explore other directions in the intrusion detection field. Unfortunately, due to lack of time, I could not test and explore all these directions in this instance.
Consequently, in the future works, several points will be developed and explored regarding the proposed IDS approach and how to improve the proposed dataset. These future works are summarised below:

- IDS approach:
  - The MESA model: In depth analysis of the MESA model is planned in order to identify more anomalies in addition to those presented in this thesis.
  - IDS investigation: For further efficiency, the syslog of the Programmable Logic Controller (PLC) is intended to be extracted and analysed for the purpose of security auditing. Syslog is a good way to centralise event logs, making it possible to identify the failures of a system more quickly and efficiently. The final goal is to use the results of analysis of these syslogs to consolidate those of the proposed IDS.
  - Extending this approach to other industrial protocols related to industry 4.0 such as MQTT and OPC-UA.
- Dataset:
  - Propose an approach to perform the labelling task automatically.
  - Apply a reduction or a selection techniques to determine the most suitable features for our study. This would result in reduced training time and improved performance metrics.
  - Focusing on these features to determine which among them allow the classification of our attacks.
  - The proposed dataset is based on network traffic, another one which will be based on process parameters is planned to be proposed for further investigations.

Bibliography

Simon, Herbert A (1960), "The new science of management decision.", in:

Hochberg, Judith et al. (1993), "NADIR: An automated system for detecting network intrusion and misuse", in: Computers & Security 12.3, pp. 235–248.

Anderson, Debra, Thane Frivold, and Alfonso Valdes (1995), "Next-generation intrusion detection expert system (NIDES): A summary", in:

Paxson, Vern (Jan. 1998), "Bro: A System for Detecting Network Intruders in Real-Time", in: 7th USENIX Security Symposium (USENIX Security 98), San Antonio, TX: USENIX Association, url: https://www.usenix.org/conference/7th-usenix-security-symposium/bro-system-detecting-network-intruders-real-time.

Hettich, Seth and SD Bay (1999), "The UCI KDD Archive [http://kdd. ics. uci. edu]. Irvine, CA: University of California", in: Department of Information and Computer Science 152.

Agatonovic-Kustrin, S and R Beresford (2000), "Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research", in: Journal of pharmaceutical and biomedical analysis 22.5, pp. 717–727.

McHugh, John (2000), "Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory", in: ACM Transactions on Information and System Security (TISSEC) 3.4, pp. 262–294.

Williams, Paul (2001), "Information security governance", in: Information security technical report 6.3, pp. 60–70.

Choi, Byoung K and Byung H Kim (2002), "MES (manufacturing execution system) architecture for FMS compatible to ERP (enterprise planning system)", in: International Journal of Computer Integrated Manufacturing 15.3, pp. 274–284.

Yin, Jian et al. (2002), "Byzantine fault-tolerant confidentiality", in: Proceedings of the International Workshop on Future Directions in Distributed Computing, Citeseer, pp. 12–15.

Marakas, George M (2003), Decision support systems in the 21st century, vol. 134, Prentice Hall Upper Saddle River, NJ.

Mellia, Marco, Andrea Carpani, and Renato Lo Cigno (2003), "Tstat: TCP statistic and analysis tool", in: International Workshop on Quality of Service in Multiservice IP Networks, Springer, pp. 145–157.

Johnsson, Charlotta (2004), "ISA 95-how and where can it be applied", in: ISA Expo, pp. 1–10.

Kazienko, Przemyslaw and Piotr Dorosz (2004), "Intrusion detection systems (IDS) Part 2-Classification; methods; techniques", in: WindowsSecurity. com.

Ostermann, Shawn (2005), Tcptrace.

Tsang, Chi-Ho and Sam Kwong (2005), "Multi-agent intrusion detection system in industrial network using ant colony clustering approach and unsupervised feature extraction", in: 2005 IEEE international conference on industrial technology, IEEE, pp. 51–56.

Igure, Vinay M, Sean A Laughter, and Ronald D Williams (2006), "Security issues in SCADA networks", in: computers & security 25.7, pp. 498–506.

Yang, Dayu, Alexander Usynin, and J Wesley Hines (2006), "Anomaly-based intrusion detection for SCADA systems", in: 5th intl. topical meeting on nuclear plant instrumentation, control and human machine interface technologies (npic&hmit 05), pp. 12–16.

Cheung, Steven et al. (2007), "Using model-based intrusion detection for SCADA networks", in: Proceedings of the SCADA security scientific symposium, vol. 46, Citeseer, pp. 1–12.

Zhengbing, Hu, Li Zhitang, and Wu Junqi (2008), "A novel network intrusion detection system (nids) based on signatures search of data mining", in: First International Workshop on Knowledge Discovery and Data Mining (WKDD 2008), IEEE, pp. 10–16.

Barika, F, K Hadjar, and N El-Kadhi (2009), "Artificial neural network for mobile IDS solution", in: Security and Management, pp. 271–277.

Brown, C. et al. (2009), "Analysis of the 1999 DARPA/Lincoln Laboratory IDS evaluation data with NetADHICT", in: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, pp. 1–7.

Lee, W and S Jang (2009), "A study on information security management system model for small and medium enterprises", in: Recent advances in e-activities, information security and privacy, pp. 84–87.

Linda, Ondrej, Todd Vollmer, and Milos Manic (2009), "Neural network based intrusion detection system for critical infrastructures", in: 2009 international joint conference on neural networks, IEEE, pp. 1827–1834.

Tavallaee, Mahbod et al. (2009), "A detailed analysis of the KDD CUP 99 data set", in: 2009 IEEE symposium on computational intelligence for security and defense applications, IEEE, pp. 1–6.

Fovino, I. N. et al. (2010), "Modbus/DNP3 State-Based Intrusion Detection System", in: 2010 24th IEEE International Conference on Advanced Information Networking and Applications, pp. 729–736, doi: 10.1109/AINA.2010.86.

Carcano, Andrea et al. (2011), "A multidimensional critical state analysis for detecting intrusions in SCADA systems", in: IEEE Transactions on Industrial Informatics 7.2, pp. 179–186.

Prusty, Swagatika, Brian Neil Levine, and Marc Liberatore (2011), "Forensic investigation of the OneSwarm anonymous filesharing system", in: Proceedings of the 18th ACM conference on Computer and communications security, pp. 201–214.

Song, Jungsuk et al. (2011), "Statistical analysis of honeypot data and building of Kyoto 2006+ dataset for NIDS evaluation", in: Proceedings of the first workshop on building analysis datasets and gathering experience returns for security, pp. 29–36.

Shiravi, Ali et al. (2012), "Toward developing a systematic approach to generate benchmark datasets for intrusion detection", in: computers & security 31.3, pp. 357–374.

Lin, Hui et al. (2013), "Adapting bro into scada: building a specification-based intrusion detection system for the dnp3 protocol", in: Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop, pp. 1–4.

Raza, Shahid, Linus Wallgren, and Thiemo Voigt (2013), "SVELTE: Real-time intrusion detection in the Internet of Things", in: Ad hoc networks 11.8, pp. 2661–2674.

Sedjelmaci, Hichem, Sidi Mohammed Senouci, and Mohammed Feham (2013), "An efficient intrusion detection framework in cluster-based wireless sensor networks", in: Security and Communication Networks 6.10, pp. 1211–1224.

Almalawi, Abdulmohsen, Xinghuo Yu, et al. (2014), "An unsupervised anomaly-based detection approach for integrity attacks on SCADA systems", in: Computers & Security 46, pp. 94–110.

Aparicio-Navarro, Francisco J, Konstantinos G Kyriakopoulos, and David J Parish (2014), "Automatic dataset labelling and feature selection for intrusion detection systems", in: 2014 IEEE Military Communications Conference, IEEE, pp. 46–51.

Bhattacharya, Sangeeta and S Selvakumar (2014), "Ssenet-2014 dataset: A dataset for detection of multiconnection attacks", in: 2014 3rd International Conference on Eco-friendly Computing and Communication Systems, IEEE, pp. 121–126.

Diallo, David and Mathieu Feuillet (2014), "Détection d'intrusion dans les systèmes industriels: Suricata et le cas de Modbus", in: C&ESAR2014.(cf. p 44).

Garcia, Sebastian et al. (2014), "An empirical comparison of botnet detection methods", in: computers & security 45, pp. 100–123.

Maglaras, Leandros A and Jianmin Jiang (2014), "Intrusion detection in SCADA systems using machine learning techniques", in: 2014 Science and Information Conference, IEEE, pp. 626–631.

Morris, Thomas and Wei Gao (2014), "Industrial control system traffic data sets for intrusion detection research", in: International Conference on Critical Infrastructure Protection, Springer, pp. 65–78.

Parvania, Masood et al. (2014), "Hybrid control network intrusion detection systems for automated power distribution systems", in: 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, IEEE, pp. 774–779.

Wheelus, Charles et al. (2014), "A Session Based Approach for Aggregating Network Traffic Data–The SANTA Dataset", in: 2014 IEEE International Conference on Bioinformatics and Bioengineering, IEEE, pp. 369–378.

Almalawi, Abdulmohsen, Adil Fahad, et al. (2015), "An efficient data-driven clustering technique to detect attacks in SCADA systems", in: IEEE Transactions on Information Forensics and Security 11.5, pp. 893–906.

Kabir-Querrec, Maëlle et al. (2015), "Architecture des systèmes d'automatisation des postes résiliente aux attaques des trames GOOSE", in:

Kumar, Malay et al. (2015), "Data outsourcing: A threat to confidentiality, integrity, and availability", in: 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), IEEE, pp. 1496–1501.

Pereira, Lucas and Nuno J Nunes (2015), "Semi-automatic labeling for public non-intrusive load monitoring datasets", in: 2015 Sustainable Internet and ICT for Sustainability (SustainIT), IEEE, pp. 1–4.

Pongle, Pavan and Gurunath Chavan (2015), "Real time intrusion and wormhole attack detection in internet of things", in: International Journal of Computer Applications 121.9.

Shang, Wenli et al. (2015), "Industrial communication intrusion detection algorithm based on improved one-class SVM", in: 2015 World Congress on Industrial Control Systems Security (WCICSS), IEEE, pp. 21–25.

Stouffer, KA et al. (2015), NIST SP 800–82 rev2. Guide to Industrial Control Systems (ICS) Security: SCADA Systems, DCS, and Other Control System Configurations Such As Programmable Logic Controllers (PLC), tech. rep., Technical report, USA.

Zhou, K., Taigang Liu, and Lifeng Zhou (2015), "Industry 4.0: Towards future industrial opportunities and challenges", in: 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), pp. 2147–2152.

Zuech, Richard et al. (2015), "A new intrusion detection benchmarking system", in: The Twenty-Eighth International Flairs Conference.

Barbosa, Rafael Ramos Regis, Ramin Sadre, and Aiko Pras (2016), "Exploiting traffic periodicity in industrial control networks", in: International journal of critical infrastructure protection 13, pp. 52–62.

Caselli, Marco et al. (2016), "Specification mining for intrusion detection in networked control systems", in: 25th {USENIX} Security Symposium ({USENIX} Security 16), pp. 791–806.

Garcia, Luis et al. (2016), "Detecting PLC control corruption via on-device runtime verification", in: 2016 Resilience Week (RWS), IEEE, pp. 67–72.

Gharib, Amirhossein et al. (2016), "An evaluation framework for intrusion detection dataset", in: 2016 International Conference on Information Science and Security (ICISS), IEEE, pp. 1–6.

Hodo, Elike et al. (2016), "Threat analysis of IoT networks using artificial neural network intrusion detection system", in: 2016 International Symposium on Networks, Computers and Communications (ISNCC), IEEE, pp. 1–6.

Kang, Ningxuan et al. (2016), "A Hierarchical structure of key performance indicators for operation management and continuous improvement in production systems", in: International Journal of Production Research 54.21, pp. 6333–6350.

Lemay, Antoine and José M Fernandez (2016), "Providing {SCADA} network data sets for intrusion detection research", in: 9th Workshop on Cyber Security Experimentation and Test ({CSET} 16).

Saeed, Ahmed et al. (2016), "Intelligent intrusion detection in low-power IoTs", in: ACM Transactions on Internet Technology (TOIT) 16.4, pp. 1–25.

Thanigaivelan, Nanda Kumar et al. (2016), "Distributed internal anomaly detection system for Internet-of-Things", in: 2016 13th IEEE annual consumer communications & networking conference (CCNC), IEEE, pp. 319–320.

Yang, Yi et al. (2016), "Multidimensional intrusion detection system for IEC 61850-based SCADA networks", in: IEEE Transactions on Power Delivery 32.2, pp. 1068–1078.

Becker, Johan and My Vester (2017), "Intrusion detection system framework for Internet of Things", PhD thesis, MS thesis, Dept. Comput. Sci. Eng., Chalmers Univ. Technol., Gothenburg, Sweden.

BOUROUH, Mouloud and Zakaria KANOUN (2017), "Détection d'intrusions à base des réseaux de neurones et algorithmes génétiques.", PhD thesis, 14-01-2018.

Huh, Jun-Ho (2017), "PLC-based design of monitoring system for ICT-integrated vertical fish farm", in: Human-centric Computing and Information Sciences 7.1, pp. 1–19.

Lashkari, Arash Habibi et al. (2017), "Characterization of Tor Traffic using Time based Features.", in: ICISSP, pp. 253–262.

McDermott, Christopher D and Andrei Petrovski (2017), "Investigation of computational intelligence techniques for intrusion detection in wireless sensor networks.", in: International journal of computer networks and communications 9.4.

Meany, Tom (2017), "Functional safety and Industrie 4.0", in: 2017 28th Irish Signals and Systems Conference (ISSC), IEEE, pp. 1–7.

Medjek, Faiza et al. (2017), "A trust-based intrusion detection system for mobile rpl based networks", in: 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), IEEE, pp. 735–742.

Ring, Markus et al. (2017), "Flow-based benchmark data sets for intrusion detection", in: Proceedings of the 16th European Conference on Cyber Warfare and Security (ECCWS), ACPI, pp. 361–369.

Schuman, Catherine D et al. (2017), "A survey of neuromorphic computing and neural networks in hardware", in: arXiv preprint arXiv:1705.06963.

Van Aubel, Pol et al. (2017), "Side-channel based intrusion detection for industrial control systems", in: International Conference on Critical Information Infrastructures Security, Springer, pp. 207–224.

Hijazi, Ahmad, Abed El Safadi, and Jean-Marie Flaus (2018), "A Deep Learning Approach for Intrusion Detection System in Industry Network.", in: BDCSIntell, pp. 55–62.

Koucham, Oualid et al. (2018), “Efficient Mining of Temporal Safety Properties for Intrusion Detection in Industrial Control Systems”, in: IFAC-PapersOnLine 51.24, pp. 1043–1050.

Maciá-Fernández, Gabriel et al. (2018), “UGR ‘16: A new dataset for the evaluation of cyclostationarity-based network IDSs”, in: Computers & Security 73, pp. 411–424.

Sharafaldin, Iman et al. (2018), “Towards a reliable intrusion detection benchmark dataset”, in: Software Networking 2018.1, pp. 177–200.

Sharma, Rohini, RK Singla, and Ajay Guleria (2018), “A New Labeled Flow-based DNS Dataset for Anomaly Detection: PUF Dataset”, in: Procedia computer science 132, pp. 1458–1466.

Sicard, Franck, Eric Zamai, and Jean-Marie Flaus (2018), “Filters based approach with temporal and combinational constraints for cybersecurity of industrial control systems”, in: IFAC-PapersOnLine 51.24, pp. 96–103.

Alem, S. et al. (2019), “A Hybrid Intrusion Detection System in Industry 4.0 Based on ISA95 Standard”, in: 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), pp. 1–8.

Alem, Salwa et al. (2019), “A hybrid intrusion detection system in industry 4.0 based on ISA95 standard”, in: 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), IEEE, pp. 1–8.

Gao, Jun et al. (2019), “Omni SCADA Intrusion Detection Using Deep Learning Algorithms”, in: arXiv preprint arXiv:1908.01974.

Khraisat, Ansam et al. (2019), “Survey of intrusion detection systems: techniques, datasets and challenges”, in: Cybersecurity 2.1, p. 20.

Monzer, Mohamad Houssein, Kamal Beydoun, and Jean-Marie Flaus (2019), “Model based rules generation for Intrusion Detection System for industrial systems”, in: 2019 International Conference on Control, Automation and Diagnosis (ICCAD), IEEE, pp. 1–6.

Arshad, Junaid et al. (2020), “A Review of Performance, Energy and Privacy of Intrusion Detection Systems for IoT”, in: Electronics 9.4, p. 629.

Dalal, Kushal Rashmikant (2020), “Analysing the Role of Supervised and Unsupervised Machine Learning in IoT”, in: 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), IEEE, pp. 75–79.

Liu, Jinping et al. (2020), "Toward security monitoring of industrial cyber-physical systems via hierarchically distributed intrusion detection", in: Expert Systems With Applications, p. 113578.

Qian, Junlei et al. (2020), "Cyber-physical integrated intrusion detection scheme in SCADA system of process manufacturing industry", in: IEEE Access 8, pp. 147471–147481.

Ross, Ronald S (2020), "Security and Privacy Controls for Information Systems and Organizations", in:

Basler, (2020), Real-Time Capability, https://www.baslerweb.com/en/vision-campus/camera-technology/real-time-capability/, (visited on 2020).

Brandl, Dennis (2016), New integration architectures for federated systems, https://www.isa.org/intech-home/2016/march-april/features/new-integration-architectures-for-federated-system, (visited on 2016).

us-cert (2016), Recommended Practice: Improving Industrial Control System Cybersecurity with Defense-in-Depth Strategies, https://us-cert.cisa.gov/sites/default/files/recommended_practices/NCCIC_ICS-CERT_Defense_in_Depth_2016_S508C.pdf, (visited on 2016).

choisirmonerp (2020), Définition d'un ERP ou PGI (Progiciel de Gestion Intégré), https://www.choisirmonerp.com/erp/definition-d-un-erp, (visited on 2020).

Dennis Brandl BR&L Consulting, Inc (2016), ISO 22400-2:2014, https://www.iso.org/fr/standard/54497.html (2016), (visited on 2016).

Fukuda, Yoshiro (2014), ISO22400Standardization of Key Performance Standardization of Key Performance Indicator for Manufacturing Execution St System, http://www.mstc.or.jp/iaf/event/2014w/01fukuda.pdf, (visited on 2014).

i-scoop (2017), Industry 4.0: the fourth industrial revolution – guide to Industrie 4.0, https://www.i-scoop.eu/industry-4-0/, (visited on 2017).

MESA (2014), MESA MOM Capability Maturity Model Version 1.0, https://services.mesa.org/ResourceLibrary/ShowResource/a4fcb3cc-bc28-4f87-84cb-3da7432cc3b2 (2014), (visited on 2014).

Mickael, Michelle (2019), Whitepaper Industrial Security based on IEC 62443, https://vdocuments.mx/whitepaper-industrial-security-based-on-iec-62443-tuvitde-contents-list-of-abbreviations.html, (visited on 2019).

Publications list

Articles

Alem, Salwa, David Espes, Eric Martin, Laurent Nana, and Florent De Lamotte (2019), "A hybrid intrusion detection system in industry 4.0 based on ISA95 standard", in: pp. 1–8.

Alem, Salwa, David Espes, Eric Martin, Laurent Nana, and Florent de Lamotte (2020), "New Dataset for Industry 4.0 to Address the Change in Threat Landscape", in: Lecture Notes in Computer Science 12528, ed. by Joaquin Garcia-Alfaro et al., pp. 273–288, doi: 10.1007/978-3-030-68887-5_16, url: https://doi.org/10.1007/978-3-030-68887-5%5C_16.

Seminaries

Alem, Salwa (2019), Cybersécurité dans l'industrie du futur IDS (Intrusion Detection System) basé sur ISA95, Presented at ISA France, Intelligence artificielle et industrie du futur, Grenoble - November 5, 6, 2019.

Journal

Alem, Salwa, David Espes, Eric Martin, Laurent Nana, and Florent de Lamotte (2021), A novel hybrid intrusion detection system approach for industry 4.0, Computers in Industry, 2021 (in submission).

Appendix

Appendix A

Specification-based IDS

A.1 Structuring concepts

A.1.1 Activity models

Figure A.1 illustrates the activity models of this standard in relationship to ISA-95.01 and ISA-95.02. The activities in this standard exchange information with activities defined as Level 4 and Level 2 activities. The grey circles indicate the activities detailed in this standard. The information flows between the activities of this standard, indicated as heavy dashed lines, are described in general in this standard. In addition, the information flows between the activities of this standard and dependent Level 2 activities are identified.

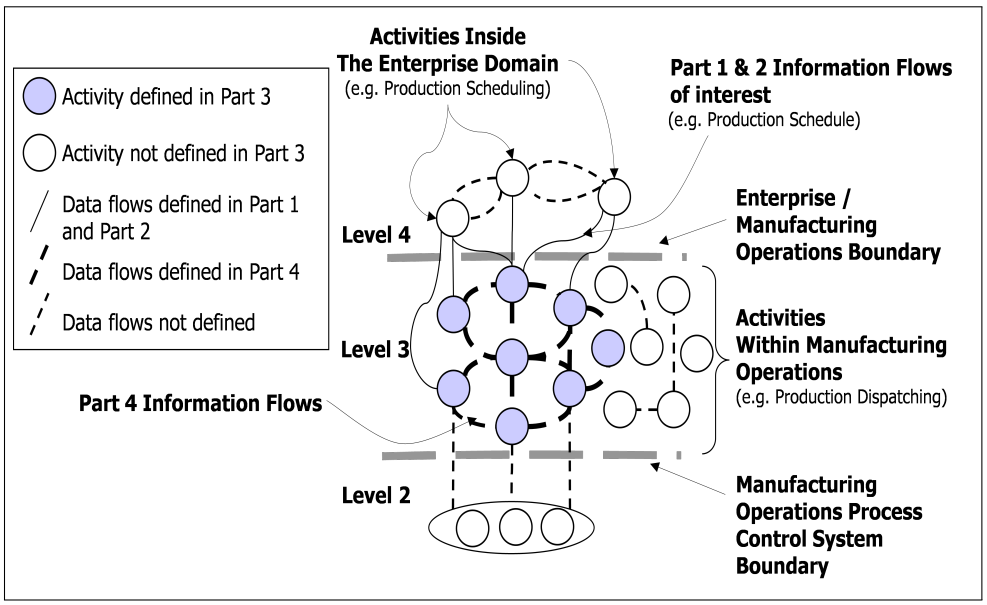


Figure A.1 – Activity relationships

The shaded areas in Figure A.1 represent the manufacturing operations management activities modelled in this standard. Manufacturing operations management is the collec-

tion of production operations management, maintenance operations management, quality operations management, inventory operations management and other activities of a manufacturing facility.
This standard defines four formal models: production operations management, maintenance operations management, quality operations management and inventory operations management. These are detailed in Clauses 6, 7, 8 and 9 and are listed below.

1. The production operations management model, which shall include the activities of production control (3.0) that operate as Level 3 functions and the subset of the production scheduling (2.0) that operate as Level 3 functions and as shown in Figure A.1.

2. The maintenance operations management model, which shall include the activities of maintenance management (10.0) that operate as Level 3 functions.

3. The quality operations management model, which shall include the activities of quality assurance (6.0) that operate as Level 3 functions.

4. The inventory operations management model, which shall include the activities of management of inventory and material including product inventory control (7.0) and material and energy control activities (4.0) defined as operating as Level 3 functions.

A.2 Structuring models

A.2.1 Generic template for categories of manufacturing operations management

A.2.1.1 Template for management of operations

A generic model for management of operations shall be used as a template to define the production operations management, maintenance operations management, quality operations management and inventory operations management models. This model is shown in Figure A.2. This generic model is extended for each specific area in later clauses.
NOTE: The fine details of the generic model are different for each of the manufacturing operations management areas.

A.2.1.2 Use of the generic model

The generic model is instantiated for the four categories listed in 5.1.1. However, this same template may be instantiated for other possible manufacturing operations categories or for other operations areas within the enterprise. EXAMPLE 1 A company could apply the model to receiving operations management and shipping operations management where these are separately managed.
EXAMPLE 2 A company could apply the model to cleaning and sterilization operations management, where these are separately managed.
EXAMPLE 3 A company could apply the model to independent logistics operations management categories for inbound logistics, outbound logistics, internal transfer and inventory control.
NOTE: This clause is normative so that companies that apply the generic model to areas other than the four detailed in this standard can determine and document their degree of conformance to the model.
When the generic model is instantiated for a new category, the activities within a category shall include the definitions of resource management, definition management, dispatching, tracking, data collection, analysis, detailed scheduling, and execution management.

A.2.1.3 Generic activity model

There shall be a hierarchy used in this standard that starts at a category of operations management. Each category is composed of a collection of activities and each activity is composed of a set of tasks. The generic model applies to the sets of activities.
The generic activity model defines a general request-response cycle that starts with requests or schedules, converts them into a work schedule, dispatches work according to the schedule, manages the execution of work, collects data and converts the collected data back into responses.
This request-response cycle is supported with:

- analysis of the work performed for improvements or corrections;
- management of the resources used in execution of the performed work;
- management of the definitions of the performed work.

The generic activity model and the detailed models are not intended to represent an actual implementation of a manufacturing information system. However, they do provide a consistent framework for such systems. Actual systems may use different structures supporting other task arrangements. The purpose of these models is to identify possible data flows within manufacturing operations. The generic model is illustrated in Figure A.2. The ovals in the model indicate collections of tasks, identified as the main activities. Lines with arrowheads indicate some of the information flows between the activities.

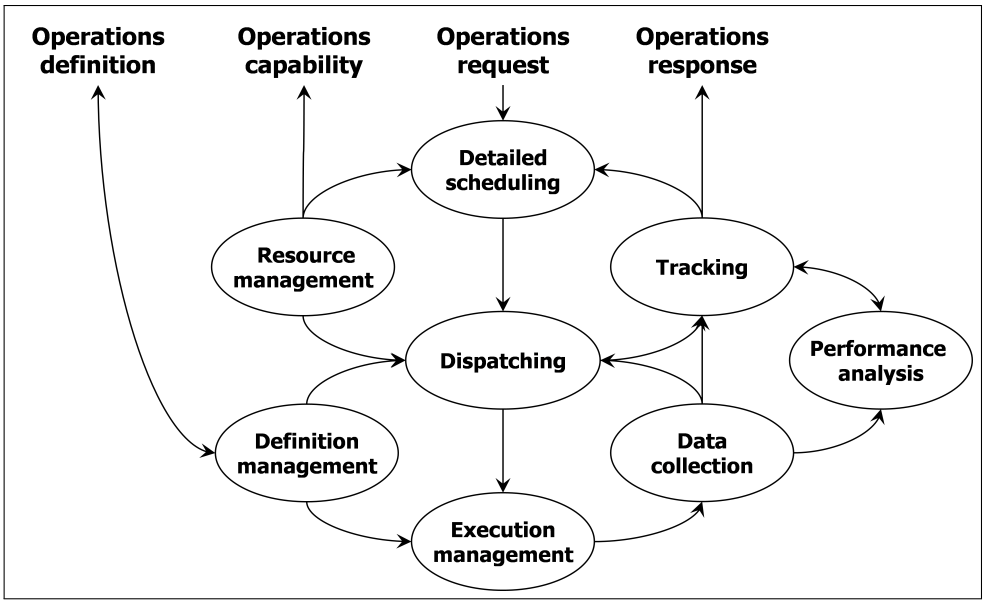


Figure A.2 – Generic activity model of manufacturing operations management

A.2.2 Interaction among generic activity models

A.2.2.1 Information flows between generic activity models

In addition to the information flows within the activities of specific operations categories, there are also information flows between the different categories. Some of this information is defined in the following clauses, but not all information flows are explicitly defined in this standard. NOTE:Specific implementations of activity models may give prominence to one specific activity model over others.

EXAMPLE 1 In pharmaceutical industries, quality operations may provide the direction for other operations.

EXAMPLE 2 In distribution centres, inventory operations may provide the direction for other operations.

EXAMPLE 3 In consumer packaged goods, production operations may provide the direction for other operations.
EXAMPLE 4 In refining, inventory operations may provide the direction for production operations.
5.2.2 Handling resources within the generic activity models Information about resources (materials, personnel and equipment) may be handled within any one of the four activity models of manufacturing operations (production, quality, maintenance and inventory) presented in this standard.
Although data for different resources may be found in different models, there are primary reporting paths through which information should be obtained.

1. Personnel information specific to each activity model may be obtained from the specific activity model.

2. Equipment information specific to each activity model may be obtained from the specific activity model.

3. Material information specific to each activity model may be obtained from the specific activity model. However, material inventory information, including finished goods and raw materials, may be obtained from the inventory activity model. Material movement operations may be managed by activities in the production, quality, maintenance, or inventory activity models. A specific material movement instance only exists within one activity model at any given point in time.

A.2.3 Information exchange in production operations management

A.2.4 Equipment and process specific production rules

Equipment and process specific production rules shall be defined as the specific instructions sent to Level 2 based on the specific assigned tasks.
EXAMPLE Programs for CNC machines for a specific product type, PLC programs that change on the basis of the process under control, or unit recipes where these are executed in Level 2 or Level 1 equipment.

A.2.4.1 Operational commands

Operational commands shall be defined as the request information sent to Level 2. These are typically commands to start or complete elements of a work order. This information may also be SOPs displayed or given to operators, such as procedures for setting up machines or cleaning of machines.
NOTE: This information exchange corresponds to the recipe-equipment interface defined in IEC 61512-1 (see Clause 2).

A.2.4.2 Operational responses

Operational responses shall be defined as information received from Level 2 in response to commands. These typically correspond to the completion or status of elements of work orders.
NOTE:This information exchange corresponds to the recipe-equipment interface defined in IEC 61512-1 (see Clause 2).

A.2.4.3 Equipment and process specific data

Equipment and process specific data shall be defined as information received as a result of monitoring Level 2. This is typically information about the process being performed and the resources involved.

A.2.5 Product definition management

A.2.5.1 Activity definition

Product definition management shall be defined as the collection of activities that manage all of the Level 3 information about the product required for manufacturing, including the product production rules.
Product definition information is shared between product production rules, bill of material and bill of resources. The product production rules contain the information used to instruct a manufacturing operation how to produce a product. This may be called a work master (Part 4 of this standard), general, site, or master recipe (ANSI/ISA-88.00.01 and IEC 61512-1 definitions), standard operating procedure (SOP), standard operating conditions (SOC), routing, or assembly steps based on the production strategy used. The product definition information is made available to other Level 3 functions and to Level

2 functions as required.
Product definition management includes management of the distribution of product production rules. Some of the product production rules may exist in Level 2 and Level 1 equipment. When that is the case, downloads of this information shall be coordinated with other manufacturing operations management functions to avoid affecting production. This information may be included as part of operational commands when the download is part of a production execution management activity.

A.2.6 Activity model

Figure A.3 illustrates some of the interfaces to product definition management.

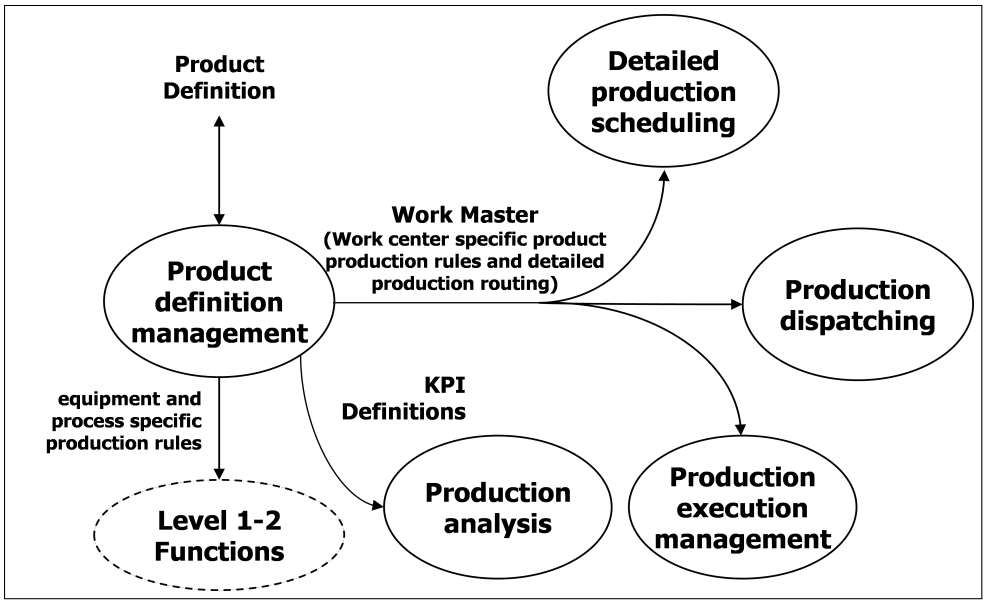


Figure A.3 – Product definition management activity model interfaces

A.2.6.1 Tasks in product definition management

Product definition management tasks may include:

- managing documents such as work masters, manufacturing instructions, recipes, product structure diagrams, manufacturing bills and product variant definitions;
- managing new product definitions;
- managing changes to product definitions; NOTE 1 This may include the ability to route work masters, designs and manufacturing bill changes through an appropriate

approval process, management of versions, tracking of modifications and security control of the information.

- providing product production rules to personnel or other activities; EXAMPLE These may take the form of work masters, manufacturing steps, master recipes, machine set-up rules and process flow sheets;
- maintaining the feasible detailed production routings for products;
- providing the route to manufacturing operations in the level of detail required by manufacturing operations;
- managing the exchange of product definition information with Level 4 functions at the level of detail required by the business operations;
- optimizing product production rules based on process analysis and production performance analysis;
- generating and maintaining local production rule sets indirectly related to products, such as for cleaning, start-up and shutdown;
- managing the key performance indicator (KPI) definitions associated with products and production.
  NOTE 2: There are a number of tools to assist in the product definition management activity, including mechanical and electronic computer-aided design (CAD), computer-aided engineering (CAE) and computer-aided software.
  NOTE 3: Engineering (CASE), recipe management systems, computer-aided process engineering (CAPE) and electronic work instructions (EWIs).

A.2.6.2 Product definition management information

Product definition is the information exchanged with engineering, R&D and others to develop the site-specific product production rules or work masters. This information may include R&D manufacturing definitions that are translated and extended by product definition management into site-specific definitions using local material, equipment and personnel. This may also involve translation of product definition information to elements of a work master.

EXAMPLE Translation to work masters, master recipes, machine set-up rules and process flow diagrams.
Product definition management may also include managing other product information in conjunction with manufacturing information. This may include:

- customer requirements, product design and test specifications;
- process design and simulation;
- technical publications and service materials;
- regulatory filings requirement information.

The product definition management activity interacts with production scheduling, production dispatching and production execution management to get the work done and interacts with research development and engineering to obtain the product production rules for executing the work.
EXAMPLE Production dispatching activities may need to refer to production dependencies to identify when a specific resource will be required.
The product production rule can contain information regarding personnel, equipment, material and product parameters. To perform these functions, product definition management may need to exchange information with resource management.

A.2.6.3 Detailed production routing

The product definition information may contain a finer granularity of definition than is visible to business systems, but is required for detailed routing of work between workcenters (process cells, production lines and production units). Detailed work order element routing is organized by the physical production process.
NOTE: A detailed production routing is sometimes called a production route, master business system route, master route, or business route.

A.2.7 Production resource management

A.2.7.1 Activity definition

Production resource management shall be defined as the collection of activities that manage the information about resources required by production operations, and relationships

between resources. The resources include machines, tools, labour (with specific skill sets), materials and energy, as defined in the object models given in ISA-95.01. Direct control of these resources in order to meet production requirements is performed in other activities, such as production dispatching and production execution management. Management of information about segments of production is also an activity in resource management.
Management of the resource information may be handled by computer systems but it may be partly or entirely handled by manual processes.
Management of the resources may include local resource reservation systems to manage information about future availability. There may be separate reservation systems for each managed critical resource. There may be separate activities for each type of resource, or combined activities for sets of resources.
Information about resources and relationships between resources needed for a segment of production must be maintained and provided on the available, committed and unattainable capacity for specific periods of time of specified resources as defined in ISA-95.01.

A.2.7.2 Activity model

Figure A.4 illustrates some of the interfaces to production resource management.

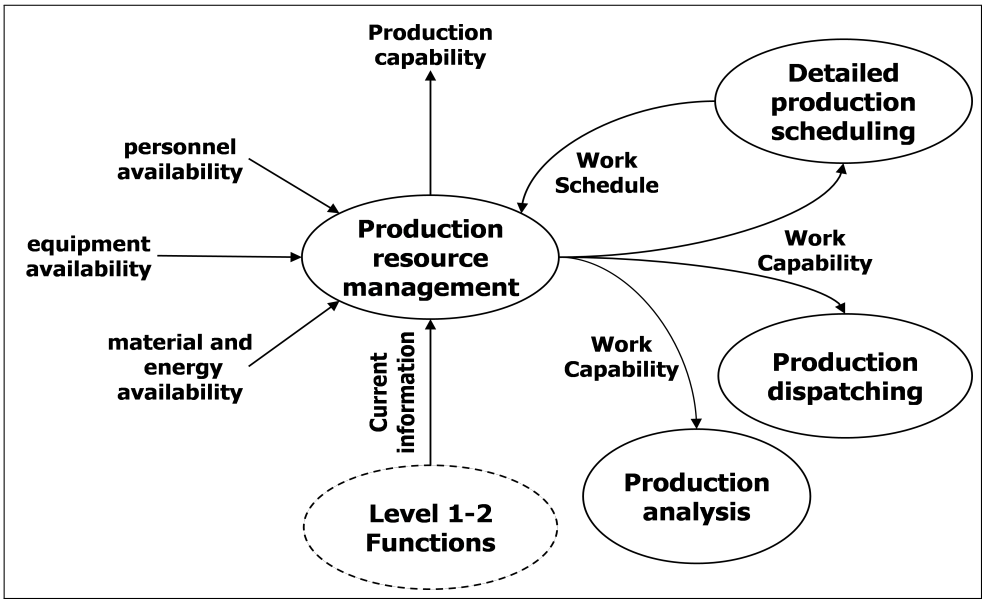


Figure A.4 – Production resource management activity model interfaces

6.5.3 Tasks in production resource management Production resource management tasks may include: a) providing personnel, material and equipment resource definitions.

The information may be provided on demand or on a defined schedule and may be provided to people, to applications, or to other activities; b) providing information on resource (material, equipment, or personnel) capability (committed, available, or unattainable). The information is based on the current statuses, future reservations and future needs (as identified in the production plan and work schedule) and is specific for resources, for defined time spans and process segments. It may include information on current balance and losses to product cost accounting and may be provided on demand or on a defined schedule and may be provided to people, to applications, or to other activities; c) ensuring that requests for acquisition of resources to meet future operational capabilities are initiated; d) ensuring that equipment is available for the assigned tasks and that job titles are correct and training is current for personnel assigned to tasks; EXAMPLE 1 Checking that an equipment sterilization status is correct ("clean") before it is assigned to a production operation. e) providing information on the location of resources and assignment of resources to areas of production; EXAMPLE 2 Providing a location for a mobile inspection machine that can be used in multiple locations. f) coordinating the management of resources with maintenance resource management and quality resource management; g) collecting information on the current state of personnel, equipment and material resources and on the capacity and capability of the resources. Information may be collected on the basis of events, on demand and/or on a defined schedule and may be collected from equipment, people and/or applications; h) collecting future needs such as from the production plan, current production, maintenance schedules, or vacation schedules; i) maintaining personnel qualification test result information; j) maintaining equipment capability test result information; k) managing reservations for future use of resources. 6.5.4 Resource availability Resource availability provides time-specific definitions needed for scheduling and reporting on a resource. The resource availability must take into account elements such as working hours, labour regulations, holiday calendar, breaks, plant shutdowns and shift schedules. EXAMPLE The available time can be a fixed time or a flexible time. For example, in personnel resource management, the time for lunch may be flexible between 11:00 a.m. and 2:00 p.m., or a machine may be unavailable for 8 h within a 16-h period. Personnel availability may define working days and days off; Monday to Friday are available for work, Saturday and Sunday are unavailable for work, or available for 2 days early shift, 2 days late shift, 2 days night shift and 3 days off. Figure 9 illustrates the types of information about the capacity of a single resource that may be provided by resource management.

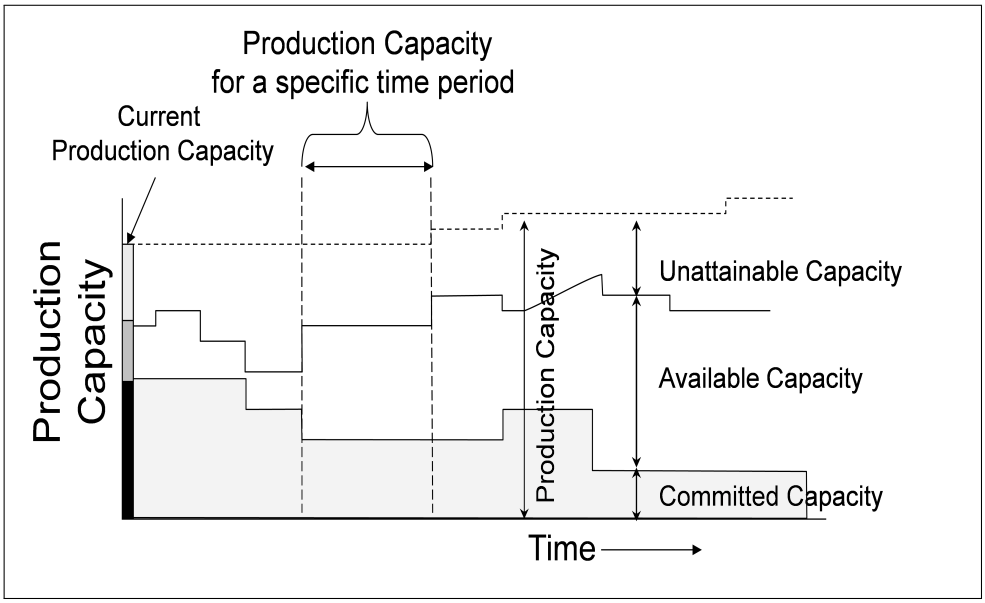


Figure A.5 – Resource management capacity reporting

A.2.7.3 Collecting future committed resource information

Production resource management manages committed resource availability based on the work schedule and product requirements. An assigned resource is identified as committed for the period of time defined by the production plan, or until the completion of the scheduled task.

NOTE:Once the schedule window requiring the resource is completed, the resource is typically taken back to the available state, unless it was already dispatched for a new assignment. In the most basic systems, the end of the planned schedule window triggers this ending of committed time window; but, for more sophisticated systems, it may be triggered by production tracking that relays the actual time the work is completed to production resource management.

A.2.7.4 Collecting resource definition changes

The production resource management activity includes collecting information about new, modified, or deleted resource definitions, classes and instances. This includes information on resource property definitions.

A.2.7.5 Personnel resource information management

Management of information about personnel resources and future personnel availability is part of resource management.

EXAMPLE If an individual has vacation planned or is known to be sick for a certain period of time, then a business-level human resource (HR) function may report this situation to production resource management. This prevents production from assigning the resource within this period of time. As an extension, the whole working schedule of the personnel should be known by production in order to make the right allocation decisions.
This may include information such as levels of certification, tracking of time spent for specific tasks and managing availability of personnel resources. In some cases this information is maintained and managed in corporate HR systems, but shall be available to manufacturing. Often the level of detail required for manufacturing, such as certification expiration dates and union line of seniority, is not maintained in the HR systems. In these cases, labour management can be considered as part of the manufacturing operations activities. The production resource management activity also has to address skill levels. Each member of the personnel may have recognized skills through qualification tests results. This defines a skill profile utilized by production resource management to allow the dispatch of the qualified personnel to each specific production activity.

A.2.7.6 Equipment resource information management

Management of information about equipment resources and future equipment availability is part of resource management.
Maintenance operations often have a major impact on resource utilization. Periods of future unavailability, based on yet unscheduled maintenance requirements, also affect utilization. EXAMPLE When a piece of equipment is reported defective, a maintenance task request could request the equipment to be classified as unavailable. The equipment would be also classified as unavailable if preventive maintenance is scheduled for this equipment. When the equipment is repaired or the preventive maintenance activity is over, the maintenance task would request that the equipment is to be taken back to its available status.
Selected equipment may be submitted to an equipment capability test as defined in ANSI/ISA 95.01 and IEC 62264-1. This test result determines if specific equipment may be assigned for a specific task in a specific process segment.

A.2.7.7 Material resource information management

Management of information about material and energy resources and future material and energy availability is part of resource management. Production resource management is informed as material is received or energy is made available. Future availability is also maintained to provide information for production scheduling.
Production resource management includes managing information about changes in material conditions, such as when material lot/sub-lot or energy source is found to have changed its specifi-

cation. Changes are often indicated from QA test results.
EXAMPLE A material lot may change from “dry” to “wet”, a pH may change from 7.0 to 7.1, or available electrical power may change from 300 kW to 280 kW.

A.2.8 Detailed production scheduling

A.2.8.1 Activity definition

Detailed production scheduling shall be defined as the collection of activities that take the production schedule and determine the optimal use of local resources to meet the production schedule requirements. This may include ordering the requests for minimal equipment set-up or cleaning, merging requests for optimal use of equipment and splitting requests when required because of batch sizes or limited production rates. Detailed production scheduling takes into account local situations and resource availability. NOTE: Enterprise-level planning systems often do not have the detailed information required to schedule specific work centers, work units, or personnel.

A.2.8.2 Activity model

Figure A.6 illustrates some of the interfaces to detailed production scheduling.

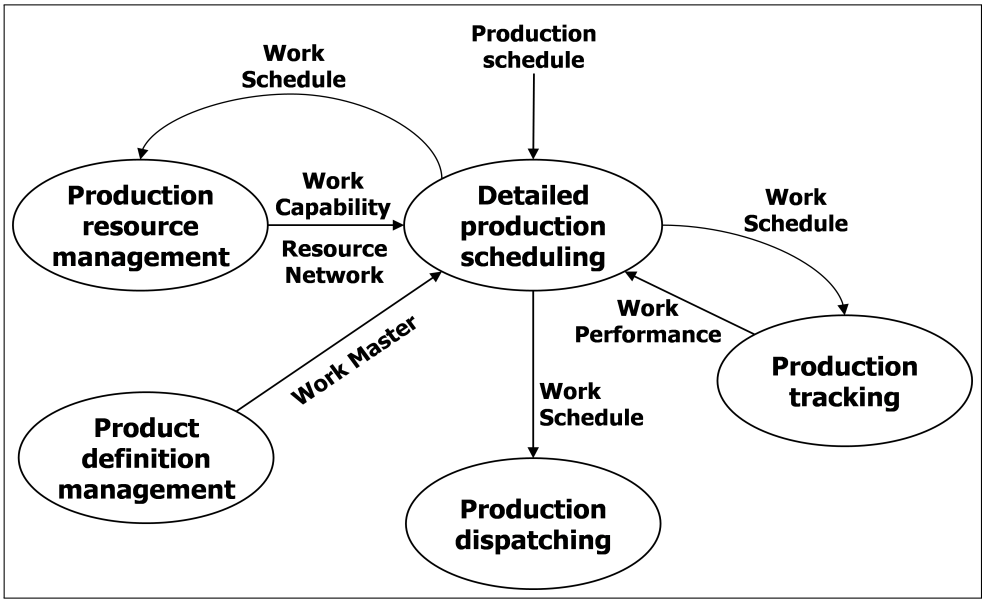


Figure A.6 – Detailed production scheduling activity model interfaces

A.2.8.3 Tasks in detailed production scheduling

Detailed production scheduling tasks may include:

1. creating and maintaining a work schedule;
2. comparing actual production to planned production;
3. determining the committed capacity of each resource for use by the production resource management function;
4. obtaining information from maintenance operations management, quality operations management and inventory operations management;
5. executing what-if simulations. This task may include activities such as calculating production lead time or final completion time for each production request provided by Level 4 functions; determining bottleneck resources for each period; and ensuring the time of future production availability for particular production.

EXAMPLE 1 Ability to promise inquiry from a Level 4 system. A work schedule is created from a Level 4 production schedule. A work schedule is based upon the requirements defined in the Level 4 schedule, the product definition and the resource capability. It accounts for constraints and availability and uses information from production tracking activities to account for actual work in progress. It may be provided either on demand or on a defined schedule. It may be recalculated on the basis of unanticipated events such as equipment outages, manpower changes and/or raw material availability changes. It may be provided to people, to applications, or to other activities.

EXAMPLE 2 Detailed production scheduling may enforce a scheduling strategy such as forward (push) or backward (pull) selection, priority assignment for each job order, application of specific constraints for the plant, time buffer allocation on bottleneck resource and so forth.

A.2.8.4 Finite capacity scheduling

Detailed production scheduling may take the form of finite capacity scheduling. Finite capacity scheduling is a scheduling methodology where work is scheduled for production resources, in such a way that no production requirement exceeds the capacity available to the production resource. Finite capacity scheduling is typically accomplished locally, at the site or area, because of the amount of detailed local information required to generate a valid work schedule. Information on current and future resource capability and capacity, as defined in ISA-95.01, is required for detailed production scheduling and is provided by production resource management activities.

A.2.9 Splitting and merging production schedules

Figure A.7 illustrates how production schedules can be split or merged prior to being sent to dispatching. The left side of Figure 11 illustrates how a single schedule is split into multiple work

schedules and the right side illustrates how multiple production schedules from multiple sources can be merged into a work schedule.
EXAMPLE 1 Multiple work schedules may be generated from a weekly production schedule, one schedule for each day of production.
EXAMPLE 2 A single work schedule may be created that combines multiple production schedule elements in order to reduce set-up time and optimize production.

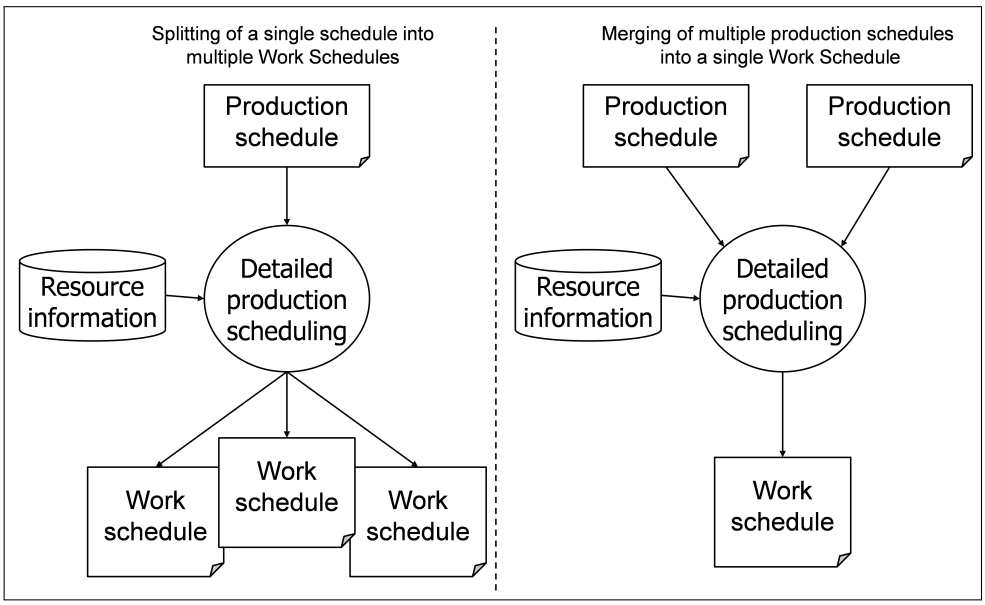


Figure A.7 – Splitting and merging production schedules to work schedules

One common function of detailed production scheduling involves merging production requests into single elements of work for purposes of reducing start-up and switchover times. This is common in scheduling for dispensing operations, where the same material is dispensed for multiple production requests at the same time in order to minimize set-up and cleaning time. This may also involve the definition of a work schedule so that related products may be produced in series, reducing or eliminating product changeover delays. Another optimization may be the optimization of batch sizes by the merging of multiple requests for the same product.
NOTE: A benefit of optimizing a work schedule for selected objective functions may be the solving of conflicts or reducing penalty of constraint violations by better sequencing and assignment of jobs.

A.2.9.1 Work schedule for production

A work schedule for production shall be defined as a collection of job orders for production and their sequencing involved in the production of one or more products, at the level of detail required

for manufacturing. Detailed production scheduling may define the generation of intermediate materials that are not included as part of higher level scheduling definitions. A work schedule ties physical and/or chemical processing to specific production equipment or classes of production equipment, with specific starting times or starting events. This is typically accomplished through job orders. A work schedule may reference specific personnel, or classes of personnel.
A work schedule defines the assignment of resources to production tasks in greater detail than the "business-oriented" process segments. A product or process segment, defined in ISA-95.01, may be realized through the execution of one or more job order elements. For example, the work schedule may define the various sub-levels of "operations-oriented" job order elements that may be required.
The work schedule also contains the information required by the production tracking activity to correlate actual production with the requested production.
EXAMPLE Work schedule for production, Figure A.8 illustrates an example work schedule for equipment represented in a Gantt chart format. The hashed rectangles in the figure represent job orders and each different hash pattern represents a different work request.

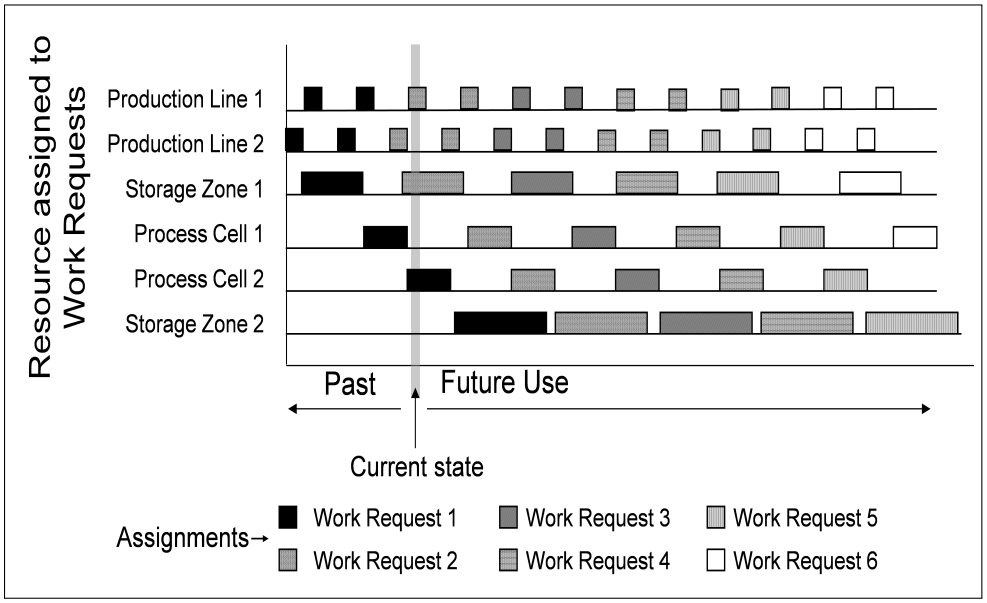


Figure A.8 – Work schedule

A.2.10 Production dispatching

A.2.10.1 Activity definition

Production dispatching shall be defined as the collection of activities that manage the flow of production by dispatching production to equipment and personnel. This may include:

1. scheduling batches to start in a batch control system;
2. scheduling production runs to start in production lines;
3. specifying standard operating condition targets in production units;
4. sending job orders to work centers;
5. issuing job orders for manual operations;

EXAMPLE Dispatched job orders may be machine set-up, grade change switchovers, equipment cleaning, run rate set-up, or production flow set-up.

A.2.10.2 Activity model

Figure A.9 illustrates some of the interfaces to production dispatching.

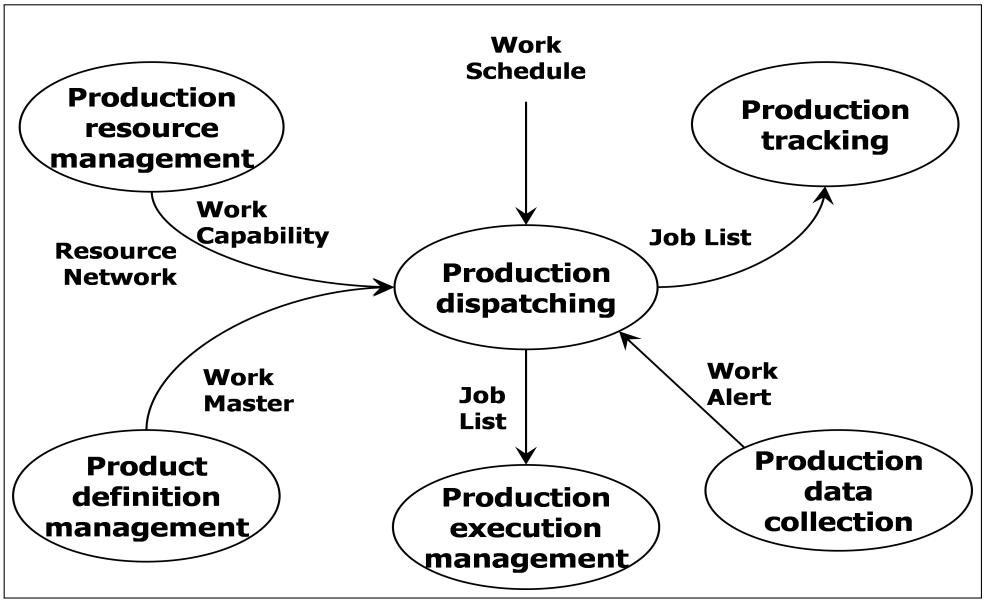


Figure A.9 – Production dispatching activity model interfaces

A.2.10.3 Tasks in production dispatching

Production dispatching tasks may include:

1. issuing job orders as identified by the schedule;
2. assigning local resources to production, where these are not identified as part of the work schedule;

3. releasing local resources to start job orders;

4. handling conditions not anticipated in the work schedule. This may involve judgment in managing workflow and buffers. This information may have to be communicated to maintenance operations management, quality operations management, inventory operations management and/or production resource management operations;

5. maintaining status of job orders; EXAMPLE Approved, fixed, in process, or cancelled.

6. ensuring that process constraints and ordering below the level of detail of the detailed schedule are met in production. This takes place after the schedule is created but before its elements are executed;

7. informing detailed production scheduling when unanticipated events result in the inability to meet the schedule requirements;

8. receiving information from quality operations management that indicates unanticipated conditions that may relate to scheduled events;

9. receiving information from production resource management about unanticipated future resource availability that may relate to scheduled events;

10. sending, or making available, the job list specifying the production activities to be performed.

A.2.10.4 Job list for production

A job list for production shall be defined as the set of job orders ready to be executed. Job orders define the specific job elements to be performed at work centers and work units. Each item in the production job list shall include the time or event to start the activity as specified in the work schedule. A production job list may take multiple forms, including batch lists (see IEC 61512-1, Clause 2), operating directives, line schedules, set-up times, or process flow specifications. The production job list correlates equipment to detailed production elements and makes this information available to production data collection and production tracking activities.

A.2.10.5 Sample production job list and jobs

Figure A.10 illustrates an example of a production work schedule and production job list represented in a GANTT chart format. Each of the hatched rectangles in the figure represents a job and each different hash pattern represents a different work request. A job list is represented as a set of job orders for a specific period of time. A job order may be defined by lower-level

elements. The collection of job orders for a specific resource is represented as a detailed resource schedule.

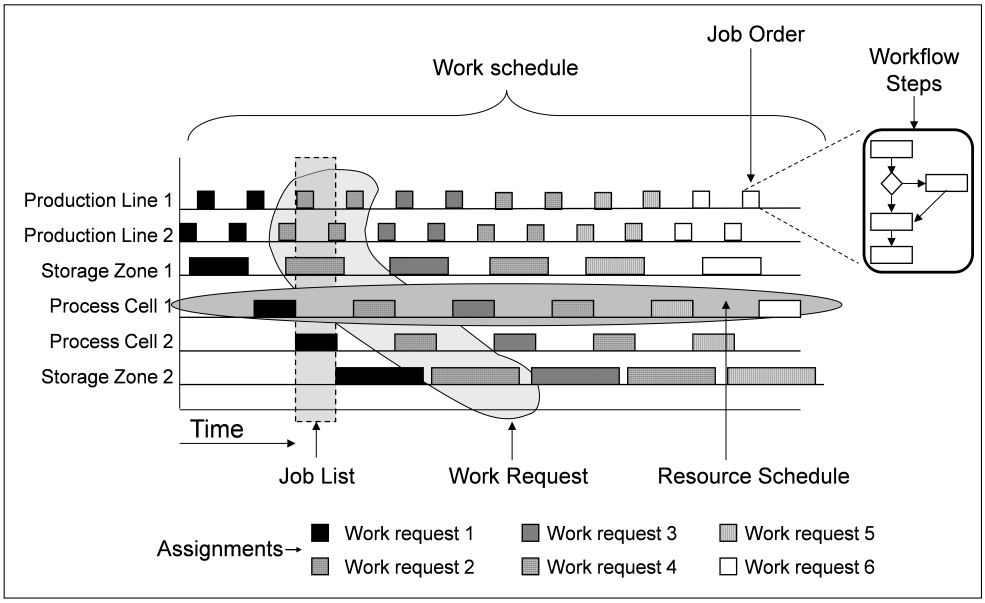


Figure A.10 – Sample job list

A.2.10.6 Assigning work

Production dispatching may include:

1. assigning material to be used in a job order;
2. assigning equipment to be used in a job order;
3. assigning personnel to execute a job order;
4. assigning storage and other resources to be used in a job order.

This activity includes the ability to control the amount of work in process through buffer management and management of rework and salvage processes, using feedback from production execution management. The activity includes the ability to cancel or reduce assigned work.
Figure A.11 illustrates an example of how the work dispatching activity may set up work in a mixed facility, with continuous, batch and discrete production segments. In this example, job lists would specify set-up for a continuous premix operation, including any initial charging. The job list would then define the sequence of batches for primary production and would also define the set-up of the back-end discrete packaging system.

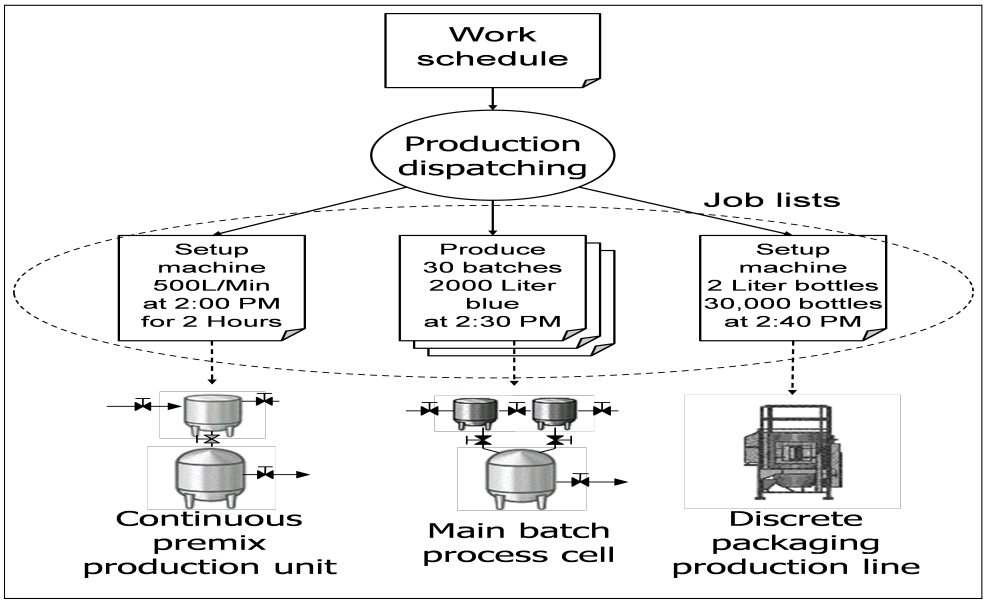


Figure A.11 – Work dispatching for mixed process facility

A.2.11 Production execution management

A.2.11.1 Activity definition

Production execution management shall be defined as the collection of activities that direct the performance of work, as specified by the contents of the job list elements. The production execution management activity includes selecting, starting and moving those units of work (for example, lots, sublots, or batches) through the appropriate sequence of operations to physically produce the product. The actual work (manual or automatic) is part of the Level 2 functions.
NOTE The definition of a sequence may take the form of a detailed production route specific for a particular produced item. Production execution transacts the individual units of work from one operation or step to the next, collecting and accounting for such things as actual materials consumed, labour hours used, yields and scrap at each step or operation. This provides visibility into the status and location of each lot or unit of work or production order at any moment in the plant and offers a way to provide external customers with visibility into the status of an order in the plant. Production execution management may use information from previous production runs, captured in production tracking, in order to perform local optimizations and increase efficiencies.

A.2.11.2 Activity model

Figure A.12 illustrates some of the interfaces to production execution management.

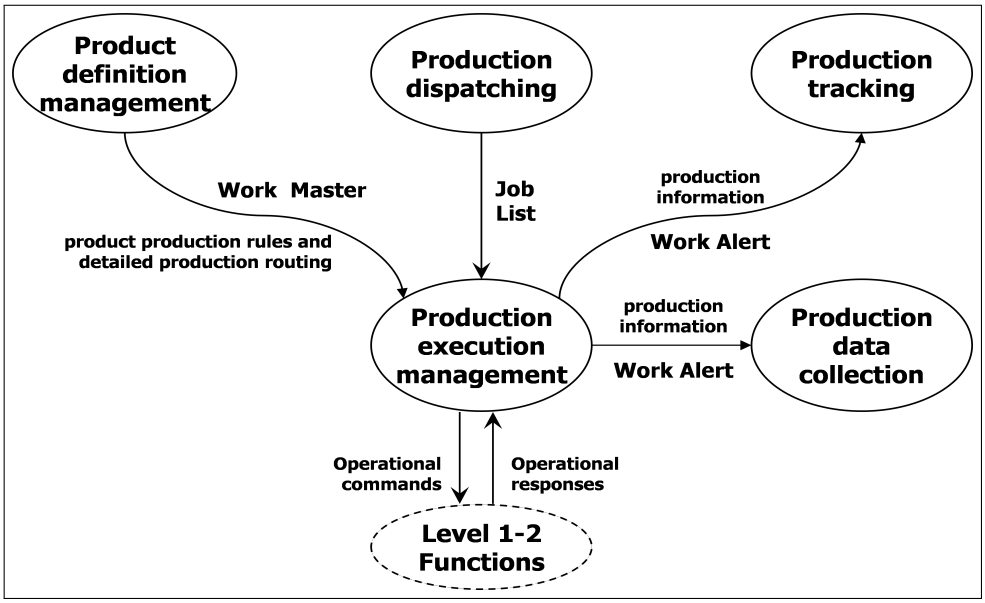


Figure A.12 – Production execution management activity model interfaces

A.2.11.3 Tasks in production execution management

The production execution management activities include the coordination of the manual and automated processes in a site, area, or work center. This often requires well-defined communication channels to automated control equipment.

Production execution management tasks may include:

1. directing the performance of work and initiating Level 2 activities;

2. ensuring that the correct resources (equipment, materials and personnel) are used in production;

3. confirming that the work is performed according to the accepted quality standards. This may involve receiving information from quality activities;

4. ensuring that resources are valid for the assigned tasks; EXAMPLE 1 This may be ensuring that equipment sterilization status is correct for the assigned operation (for example, a vessel is “clean” before use in production).
   EXAMPLE 2 Equipment certifications are current, personnel qualifications are up to date and materials are released for use.

5. assigning resources under local run time control; EXAMPLE 3 The assignment of units to a batch, if the work schedule does not define unit allocation.

6. informing other activities when unanticipated events result in the inability to meet the work requirements;

7. receiving information from production resource management about unanticipated future resource availability;

8. providing production information and events on production execution management, such as timing, yields, labour and material used, start of runs and completion of runs.

A.2.12 Production data collection

A.2.12.1 Activity definition

Production data collection shall be defined as the collection of activities that gather, compile and manage production data for specific work processes or specific production requests. Manufacturing control systems generally deal with process information such as quantities (weight, units, etc.) and associated properties (rates, temperatures, etc.) and with equipment information such as controller, sensor and actuator statuses. The managed data may include sensor readings, equipment states, event data, operator-entered data, transaction data, operator actions, messages, calculation results from models and other data of importance in the making of a product. The data collection is inherently time- or event-based, with time or event data added to give context to the collected information.

A.2.12.2 Activity model

Figure A.13 illustrates some of the interfaces to production data collection.

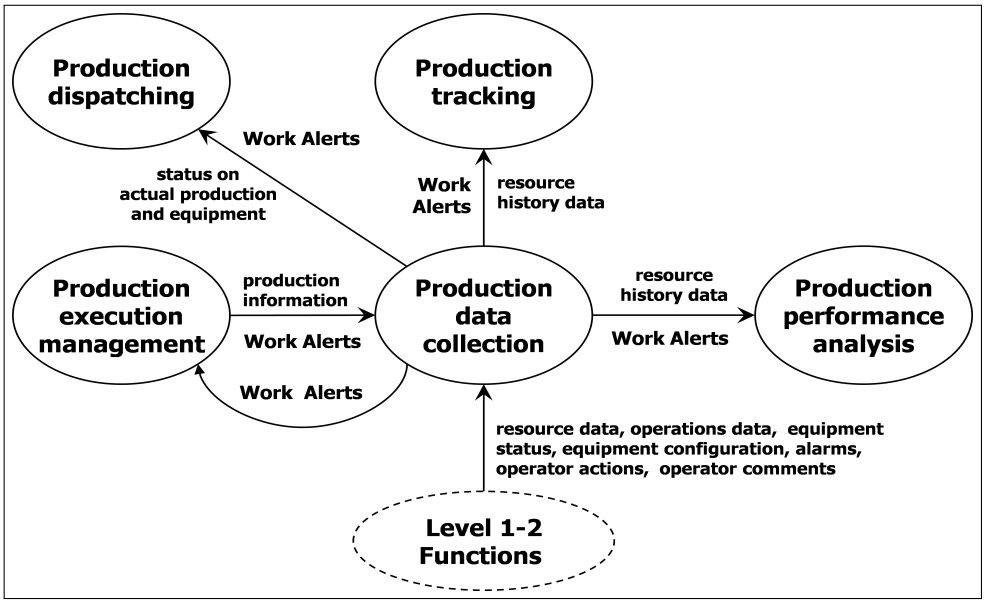


Figure A.13 – Production data collection activity model interfaces

A.2.12.3 Tasks in production data collection

Production data collection tasks may include:

1. collecting, retrieving and archiving information related to the execution of production requests, equipment usage, including information entered by production personnel; EXAMPLE: This could include the following: – process data; – equipment status data; – lot and sublot location and amount data collection; – operations logs (plant entries and comments).

2. providing interfaces to the basic process or manufacturing line control system, laboratory information management systems and production management systems for automatic collection of information;

3. providing reports on production data;

4. maintaining information for local process and production analysis and for reporting to higher-level logistics systems;

5. maintaining information for product tracking to enable tracking and tracing capability such as tracing products to specific material lots, equipment and/or operators;

6. providing compliance monitoring and alarm management functionality (event logging and sequence of events);

7. providing collected product quality information for comparison against specifications.

A.2.13 Production tracking

A.2.13.1 Activity definition

Production tracking shall be defined as the collection of activities that prepare the production response for Level 4. This includes summarizing and reporting information about personnel and equipment actually used to produce product, material consumed, material produced and other relevant production data such as costs and performance analysis results. Production tracking also provides information to detailed production scheduling and Level 4 scheduling activities so schedules can be updated on the basis of current conditions.

A.2.13.2 Activity model

Figure A.14 illustrates some of the interfaces to production tracking.

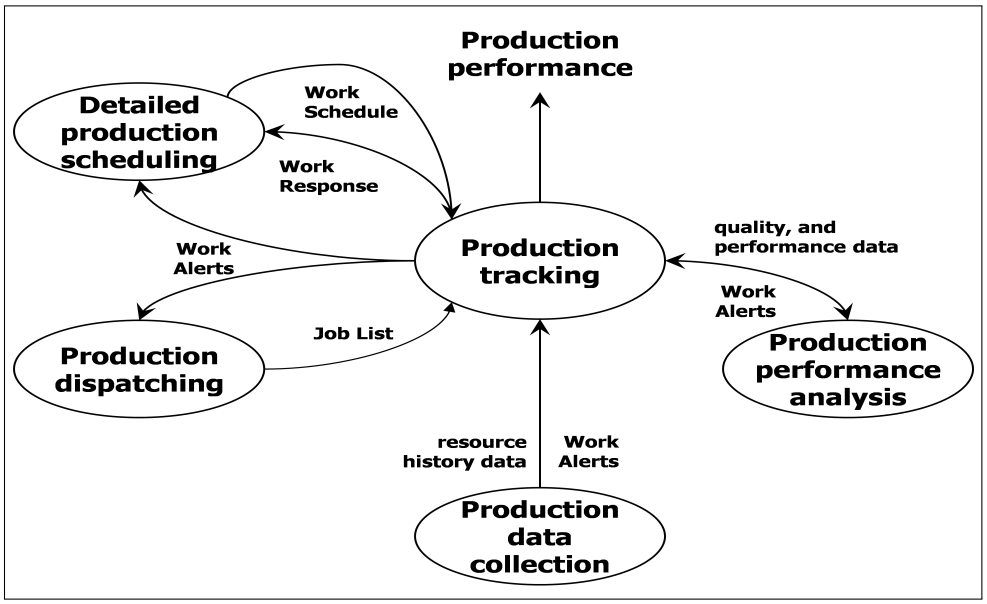


Figure A.14 – Production tracking activity model interfaces

A.2.13.3 Tasks in production tracking

Production tracking tasks may include:

1. following the movement of material through a plant by maintaining a description of what was in each vessel at specific times and tracing the path of all materials within the production domain;

2. recording the start and end of movements and collecting updates to lot and sublot quantities and locations as they occur;

3. receiving information from production data collection and production analysis; for example, information on materials consumed in the production of a lot (a part of the product tracking and tracing) and information on plant environmental conditions during the production of the lot;

4. translating process events, including production and movement events, into product information;

5. providing information for tracking (recording) and tracing (analysis);

6. generating production responses and production performance information. The information may be provided on demand or on a defined schedule and may be provided to people, to applications, or to other activities;

7. generating records related to the production process. This may include records required for regulatory or quality management purposes.

A.2.13.4 Merging and splitting production information

Production tracking may involve compiling production data into business information on actual production including in-work inventory, raw material usage, and energy usage. Production tracking may require combining resource history data from multiple batches or runs into a single production performance report. Alternatively, it may require splitting information about a single batch or run into multiple production performance reports. These are illustrated in Figure A.15.
EXAMPLE 1: Production history from multiple production lines used in completion of a single order may be combined to produce a single production response for the order.
EXAMPLE 2: Information from a single production run may be split into multiple production performance reports, one report for each shift used in the production.
EXAMPLE 3: A portion of a product run may be sent to an outside entity to perform a portion of the life cycle of completing the product. In this case, the product would share history until it leaves the internal manufacturing processes and upon return to the normal internal manufacturing processes, the same product would have a slightly different history than its peer product.

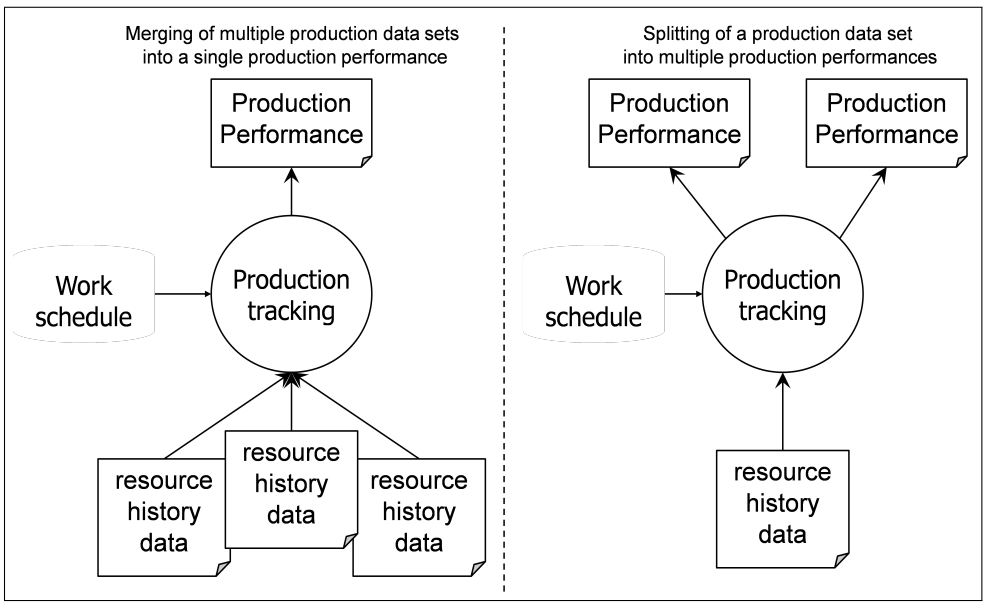


Figure A.15 – Merging and splitting production tracking information

A.2.14 Production performance analysis

A.2.14.1 Activity definition

Production performance analysis shall be defined as the collection of activities that analyse and report performance information to business systems. This would include analysis of information of production unit cycle times, resource utilization, equipment utilization, equipment performance, procedure efficiencies and production variability.

Relationships between these analyses and others may also be utilized to develop KPI reports. This information may be used to optimize production and the use of resources. Such information may be provided on a scheduled basis, it may be provided at the end of production runs or batches, or it may be provided on demand.

The process of production performance analysis is ongoing. Once an optimization has occurred and a constraint has been exploited, other system constraints may arise. Additionally, changing market conditions and product mixes may change the optimization criteria and system constraints. In a changing environment, production performance analysis activities regularly re-examine throughput and policies under current and expected conditions in order to maximize system throughput.

A.2.14.2 Activity model

Figure A.16 illustrates some of the interfaces to production performance analysis.

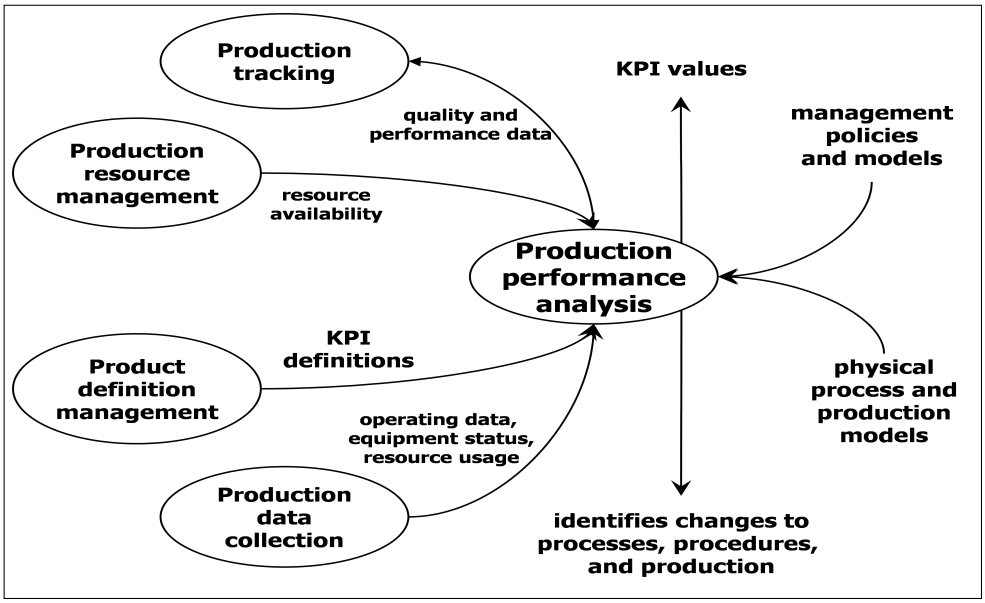


Figure A.16 – Production performance analysis activity model interfaces

A.2.14.3 Tasks in production performance analysis

Production performance analysis tasks may include:

1. producing reports of performance and cost;
2. evaluating constraints to capacity and quality,
3. performing performance tests where necessary to determine capacity;
4. comparing different production lines and creating average or target runs;
5. comparing and contrasting one run against another;
6. comparing production runs to identify “golden” runs; NOTE 1:“Golden” runs are runs that are the best run ever produced, where best may be the highest quality, or lowest cost, or any other criteria.
7. determining why the “golden” runs are exceptional;
8. comparing runs against defined “golden” runs;
9. providing changes to process and procedures based on the results of the analysis for continuing process improvements;
10. predicting the results of a production run, on the basis of current and past performance. This may include the generation of production indicators;

11. correlating the product segments with process conditions at the time of production.

EXAMPLE: The record of job order execution, product segments and process segments and their times, quantities and conditions of production could be searched and manipulated to answer the question of the form “what activity happened, how it happened (what setpoints were used, which procedure, etc.), where it happened, when it happened and who performed it?”. NOTE 2: In addition to this main question, questions related to resource tracking, such as "what was where, when and why?" for material tracking may be answered. This ability to track down product and minimize the impact from contamination can be the critical analysis tool needed to ensure future orders from customers.

A.2.14.4 Resource traceability analysis

Resource traceability analysis shall be defined as the collection of activities that trace the history of all resources (material, equipment and personnel) in terms of the process actions and events that dealt with the resources in production.
Resource traceability analysis may include analysis on

- materials produced, consumed, stored and moved;
- equipment used in production, testing and storage;
- personnel involved in the production and storage of material and operation of equipment.

NOTE 1: As a batch or lot moves through the production facility, on-the-spot decisions are made all along the way regarding raw materials locations to consume from, rework actions required based on analytical results and other similar decisions. When the unit of product moves into finished goods or out to end customers, it may be important to be able to retrace the parent supplier lots from which its raw materials were consumed, which specific personnel or equipment units were involved in the process, whether the unit of work was sent back for rework more than once, or any of a large number of similar questions.
NOTE 2: The record of a lot’s recent ancestry might be attached as part of the production response back to the enterprise system or could be of considerable value at the manufacturing operations level for implementing continuous improvement efforts.
NOTE 3: This clause deals with resource traceability from a production perspective and may need to be combined with equivalent information and functions in maintenance operations management, quality operations management and inventory operations management.
Resource traceability has two components, tracking and tracing.

1. Tracking is the process of following and recording the movements and change of resources and recording all inputs to the resource through all steps and agents.

2. Tracing is the process that determines a resource's history of use from any point, forward or backward, using tracking information.

A.2.14.5 Product analysis

Testing for product quality is one of the manufacturing operations activities. The testing may be in-line, at-line, or off-line. Product analysis also includes the off-line analysis typically performed in laboratories and the management of quality test procedures. The activities associated with product analysis are defined in 8.1.5.
Product analysis (quality assurance) activities include display of in-process information, such as statistical process control (SPC) or statistical quality control (SQC) data. Quality management handles the quality test procedures and often maintains quality test results.

A.2.14.6 Process analysis

Process analysis provides feedback about specific manufacturing processes across multiple production runs or batches. This information is used to optimise or modify specific production processes. The activity includes analysis of bad production runs to determine the root cause and analysis of exceptional quality production runs to determine optimal running conditions. Process analysis often includes SPC/SQC analysis and process modelling and uses information collected from the multiple activities that measure operating parameters.

A.2.14.7 Production performance simulation

Simulation is often used to model how a material flows through the plant and to evaluate how the process responds to changes. It may model changes in the process, changes in the production routing, or changes to the manufacturing procedures. It may also be used to predict the material properties based on the current operating process conditions. Simulation can be used during the life cycle of the plant to track performance, to track change effects and for operator training. NOTE: Simulation can show how to provide the following types of benefits to production: – adding additional capacity without significant addition of new equipment, machinery, or labour; – increasing the efficiency and effectiveness of an existing system; – eliminating bottlenecks, using existing assets better; – evaluating possibilities for quality and throughput improvements or cost reductions; – improving the ability to meet deadlines, customer commitment and changing customer requirements; – educating operators without putting personnel, the environment, physical systems, or production at risk.

A.2.14.8 KPIs

In addition to the formally defined production performance data model defined in ISA-95.01 and ISA-95.01, there is additional information about production that provides summaries of past performance, indications of future performance, or indicators of potential future problems. Collectively, this information is defined as KPIs (Key Performance Indicators). One of the activities within production performance analysis is the generation of KPIs. This information may be used internally within manufacturing operations for improvements and optimisation. If there is a receiving business process that requires the information, it may also be sent to higher-level business processes for further analysis and decisions. Manufacturing oriented KPIs are defined in the ISO 22400-2 standard, Automation systems and integration — Key performance indicators for manufacturing operations management — Part 2: Definitions and descriptions.

A.2.14.9 Performance management

Performance management shall be defined as the collection of activities that systematically capture, manage and present performance information in a consistent framework. This includes utilizing corrective actions to affect operational improvement. There is a business value to aligning lower-level manufacturing indicators with key business objectives. Some typical functions of performance management solutions are the following:

- monitoring to enable visibility of KPIs;
- ability to utilize KPI information in a model;
- root cause analysis;
- prediction of future KPI values;
- capability to enact control based on KPI values.

One of the main activities in performance management information is transforming the large volume of raw data into actionable information. A hierarchy model is typically used to analyse performance data in manufacturing and it may align with the equipment model.
EXAMPLE 1 This could be the ability to analyze all inventory by product families down to the individual product stock-keeping unit.
EXAMPLE 2 A simple model could be a summation of all subsidiary node values of an indicator. Performance indicators that are not visible significantly decrease the value of performance management. This can be compared with reports that have thousands of values on a single page. There can be an implied ranking to KPIs where those with greater impact to the enterprise have greater visibility. EXAMPLE 3 An example of a visibility metaphor is the use of a traffic

light indicating the status of an indicator. The green light indicates that the indicator is within specification. Yellow and red lights indicate an indicator has exceeded acceptable ranges. No light represents a lack of data or that the data is of poor quality. A single report may be made up of tens or hundreds of indicators allowing a quick survey if large amounts of information.
Root cause analysis is the determination of the key contributors to an indicator's value. Often an indicator's value may be caused by a hidden relationship to other information. The ultimate goal of root cause analysis is to expose the relationship so that corrective action can be taken on the underlying problem.
EXAMPLE 4 Performance management activities may be cross-functional and may look at the raw information used in the analysis. For example, this may include visibility into a lab system to see detailed results for recent lots. Another example could be visibility into production to see the current active constraints in the process control.
Prediction of future KPI values is an aspect of performance management. The traditional implementation of this prediction is in the plant plan/schedule. The plan/schedule contains information that shows future asset activity and this can be rolled up into KPIs. Another implementation of predictive indicators is to apply predictive statistics to current KPIs and estimate future values.
EXAMPLE 5 An example might be to take the historical mean time between failure values and develop a trend to predict the next failure for a piece of equipment.
Performance management includes the ability to identify and initiate an appropriate action based on an out-of-specification indicator.
EXAMPLE 6 A change of a control set point could be based on an online SPC high alarm for a key process or derived parameter.
Performance management has aspects that permeate throughout the activity model. Production, maintenance, quality and inventory operations management have critical metrics that are important not only to that function, but are used across other functions.