TU WIEN Informatics

Flexible Safety Systems

Use Cases, Requirements, System Design, and Software Architecture

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der Technischen Wissenschaften

by

Dipl.Ing.(FH) Dieter J. Etz, MBA
Registration Number 00327813

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dipl.-Ing. Dr.techn. Wolfgang Kastner

The dissertation has been reviewed by:

Prof. Dr.-Ing. habil. Leon Urbas	Prof. Dr.-Ing. Dirk Timmermann

Vienna, 10th March, 2024

Dieter J. Etz

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Dipl.Ing.(FH) Dieter J. Etz, MBA

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 10. März 2024

Dieter J. Etz

Acknowledgements

For the success of a dissertation, not only the professional environment and supervision are crucial, but it is also support and encouragement in the private sphere.

This challenging endeavor would not have been possible without the inspiration, support, and guidance of my advisor Wolfgang Kastner. I want to thank the entire Automation Systems Group, particularly Thomas Frühwirth and Patrick Denzler, for providing excellent conditions for research, talks, and discussions. Special thanks go to Ruth Fochtner, whose administrative support was very much appreciated. My sincere thanks also go to Heinz Deinhart for the indispensable help in the laboratory and valuable advice on programming, shell scripting, and IT infrastructure.

In addition to the scientific work at TU Wien, several activities have been carried out as part of the Common Research Programme within the Center for Digital Production. Therefore, I express my deep appreciation to Christoph Pollak, who made it possible to expand my research to several industry partners.

Lastly, I could not have undertaken this journey without the indulgence, patience, encouragement, and linguistic aid of my wife Dina.

Abstract

The transition from industrial automation towards customized mass production as a goal of Industry 4.0 presents challenges and opportunities for European manufacturing companies. To stay competitive in global markets, these manufacturing companies have to increase the productivity of their production facilities by taking advantage of flexible, modular, and reconfigurable production lines to compensate for the high labor costs in Europe. Such a reconfigurable manufacturing system is designed for rapid changes in its structure and configuration to adjust its functionality and production capacity in response to sudden market changes.

The objective of rapid reconfiguration of production facilities is a fundamental prerequisite for the realization of customized mass production, also called "lot-size one" production. Besides the reconfiguration of machinery and its programs, rapid reconfiguration also affects the safety systems that are mandatorily for protecting man, machines, and the environment and are, therefore, required by law. Currently, functional safety systems are designed and certified in a static way, which means that the safety configuration of a machine is defined during its design phase. Once the system is commissioned, certified, and put into operation, the safety configuration is not changed anymore. This approach impedes the flexibility that smart manufacturing production facilities require nowadays. Beyond that, the current way of safety configuration is characterized by high manual configuration efforts, which lead to high engineering costs and time-intensive and expensive machinery downtimes, and therefore, requires a new approach.

This research proposes a structure and services for a functional safety system that can be quickly and easily reconfigured once a machine is commissioned and even during operation. The design utilizes existing base technologies for safe and reliable communication and provides services for device discovery, configuration generation, and automatic deployment. It also provides a basis for further features such as automatic risk assessment, automatic safety verification, safety validation, and safety evaluation. Additionally, assistive features such as legal regulations checks and AI-supported configuration generation could be envisioned. The concept of a Flexible Safety System (FSS) aims to offer the possibility of safety configuration changes in operating production facilities and to assist the operator in applying configuration changes simply, quickly, and according to regulations.

Contents

CHAPTER 1

Introduction

> NO, SIR, WE WILL NOT GO INTO THE HOSPITAL BUSINESS. WE WILL ELIMINATE ACCIDENTS INSTEAD. IF WE OWE IT TO OUR MEN TO CARE FOR THEM WHEN THEY ARE HURT, WE CERTAINLY OWE IT TO THEM TO DO EVERYTHING IN OUR POWER TO KEEP THEM FROM GETTING HURT; AND IF IT WOULD BE A GOOD INVESTMENT FOR US TO BUILD A LARGER INDUSTRIAL HOSPITAL, IT CERTAINLY WILL BE A BETTER INVESTMENT FOR US TO GET RID OF ACCIDENTS. THAT'S YOUR JOB FROM NOW ON. PREVENT ACCIDENTS, EVEN IF YOU HAVE TO REDESIGN OUR MACHINES OR METHODS TO DO SO. [1]
>
> – Henry Ford, 1914

One of the fundamental human needs is safety. This fact was not only determined by Maslow in his 1943 paper "A theory of human motivation" [2] and affirmed by Andersen et al. [3], it was already recognized by ancient Babylonians around 2000 BCE. Their ruler, king Hammurabi, developed his "Code of Hammurabi" which encompassed all the laws of the land at that time. The significance of the code from the perspective of safety is that it contained clauses dealing with injuries, allowable fees for physicians, and monetary damages assessed against those who injured others. There is also evidence for safety concerns from later Egyptian and Roman civilizations. In the 16th century, Philippus Aureolus and Georgius Agricola published treatises regarding diseases of miners and emphasized the need for ventilation. Agricola also proposed devices that could be used to introduce fresh air into mines. Bernardino Ramazzini, also known as the father of occupational medicine, wrote his most important contribution "De Morbis Artificum Diatriba" ("Diseases of Workers") in the 18th century, outlining various health hazards encountered by workers in more than fifty occupations [4].

Accidents have always been part of human history, but their nature and location have shifted over time. Industrialization brought dramatic changes to societies in the nineteenth century, reshaping the nature of accidents. The introduction of mechanized factories and production processes, exposed everyone in society, to entirely new dangers.

The changes in methods of producing goods since the beginning of the first industrial revolution, in the period from about 1760 to 1840, required a greater focusing of attention on the safety and health of workers. During the first industrial revolution it was the steam power which increased markedly the potential for life-threatening injuries [4]. At that time, there were no laws or regulations regarding the operation of factories as there had been no need for them before. Therefore, dangerous machinery was used that could cause serious injuries to workers. Additionally, child labor was an essential part of industrial manufacturing and people were required to work incredibly long hours. As a result of these working conditions, fatigue or inattentiveness at a machine could lead to accidents culminating in the loss of limbs or even death. Due to the rapidly rising number of factories, the UK Government passed the "Health and Morals of Apprentices Act" in 1802, the very first piece of factory legislation [5]. This act was followed later, among others, by the "1833 Factory Act" in the UK and the American common law of workplace accidents in the "1842 case of Farwell v. Boston & Worcester Rail Road" [6].

Over time the challenges of safety and occupational health in industrial environments changed. During the first industrial revolution the challenge comprised the excessive labor of children and young persons for 60 hours a week or more, under nutrition and malnutrition, grossly insanitary conditions, acute infectious diseases and physical strain, and illiteracy. In the course of the second industrial revolution, which is usually dated between 1870 and 1914, safety hazards were mostly related to machines which did not have any safety covers or fences and were operated by inexperienced staff or children as young as 5 years old [7]. The quote at the beginning of the chapter, illustrates the situation in the Ford factories at that time [1]. By the begin of the third industrial revolution all these matters had been substantially remedied or at least improved. By contrast the new challenges included the ageing population and the elderly workman, mental stress, maintenance and promotion of health, rehabilitation and resettlement of persons handicapped from any cause, and technical education [5].

The third industrial revolution, or digital revolution, lasted from the 1969 until 2010 and was characterized by the spread of industrial automation made possible by two major inventions: Programmable Logic Controllers (PLCs) and Robots. It was also the time of the rise of electronics, the spread of telecommunications, the widespread availability of computers, the discovery of nuclear energy, and the invention of the Internet. The Internet emerged out of the US Department of Defense's Advanced Research Projects Agency Network (ARPANET), which first host-to-host message was sent in 1969 and thereby marked the beginning of the third industrial revolution. In this era, most of today's safety standardization was enacted. Starting with the creation of the Occupational Safety and Health Administration (OSHA) and the Occupational Safety and Health Act in 1970 by the US Congress [8]. The Council of the European

Communities published the "Machinery Directive 89/392/EEC" in 1989 with the objective of ensuring a high level of safety and protection for users of machinery and other people exposed to it [9]. Many generic and sector-specific standards followed such as IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems [10], IEC 62061: Safety of machinery - Functional safety of safety-related electrical, electronic and programmable electronic control systems [11], and ISO 13849-1: Safety of machinery – Safety-related parts of control systems [12]. To this day, IEC 61508 is the main standard for the functional safety of safety-related systems. The major objectives of this standard are to define the requirements of safety systems and to facilitate the development of product and application sector safety standards.

The fourth industrial revolution was heralded with the broad introduction of the Internet of Things (IoT) in 2011 as part of Gartner's "Hype Cycle for Emerging Technologies"[1]. In that very same year, two major countries presented their strategies for the fourth industrial revolution. Germany started the "Platform Industry 4.0" and the United States launched the "Advanced Manufacturing Partnership". Other countries like China and Japan followed a few years later [13]. Industry 4.0 (I4.0) focuses on the intelligent, horizontal, and vertical networking of people, machines, objects, and Information and Communication Technology (ICT) systems in real-time for the dynamic management of complex production systems to make manufacturing more efficient and flexible[14].

One key paradigm of I4.0 is the "Smart Machine". It describes the process of machines becoming Cyber-Physical Production Systems (CPPS) by using automation components that can communicate to other field devices, production modules, and products through open and standardized networks and semantic descriptions. Thus, the easy integration or replacement of manufacturing unities in the case of reconfiguration becomes possible. In this way, production lines become so flexible and modular that even the smallest lot size can be produced under the condition of highly flexible mass production [15].

Modular factory systems already exist today. However, these systems are based on vendor-specific communication and data exchange. They are thus not in accordance with the I4.0 vision, which stipulates interoperability and congruence of multi-vendor solutions. In the context of I4.0, factories are comprised of field devices, machines, production modules, and products that autonomously exchange information, trigger actions, and control each other. Such factory automation requires smart factory networks that enable dynamic re-engineering processes and provide the ability to respond flexibly to disruptions and failures during the production process [15].

In order to allow unhindered exchange of information across devices from various manufacturers, standardization plays a key role for the adoption of I4.0. It is the foundation of a service-oriented and interconnected automation model which requires a common language. This allows the exchange of information through the same vocabulary, syntax, semantics, formats, physical interfaces, communication protocols, interoperability, and management platforms. The information exchange includes horizontal as well as vertical

[1] https://www.gartner.com/en/documents/1754719/hype-cycle-for-emerging-technologies-2011

communication. In the I4.0 domain, this means the convergence of Operational Technology (OT) and Information Technology (IT) resulting in the Industrial Internet of Things (IIoT). OT is characterized by efficiency, consistency, continuity, and safety, whereas IT focuses on agility, cost reduction, security, speed, and commercial vision [16].

Converged networks in industrial settings imply using the same infrastructure for real-time and safety-critical manufacturing data, as well as for best effort IT traffic. There are several driving forces behind the transformation towards converged networks. First, new manufacturing architectures where rigid lines of production units being replaced by flexible modular production units. Second, comprehensive system integration which entails a progressive use of IT systems in OT, e.g. the use of virtualization technologies in OT. As a consequence, the requirements of OT, such as real-time capability, must also be fulfilled by IT systems. Third, manufacturer-independent use of devices which is made possible by standardized, uniform interfaces [17].

The high degree of automation technology in smart factories, together with high demands for flexibility and interoperability, entails also a huge configuration effort of the indispensable safety system. Therefore, the scope of this thesis is on functional safety systems that consist of sensors, logic, and actuators as well as the communication between these nodes, as depicted in Figure 1.1. It focuses on the challenges of re-configurable safety systems due to the required flexibility and interoperability in smart manufacturing. Configuration of many devices and the simultaneous deployment on several layers, such as safety devices, safety communication, and real-time data transfer, is required to enable flexibility and rapid reconfiguration. Therefore, considerations on how to facilitate the operation of such a complex system with worker assistance systems are made.

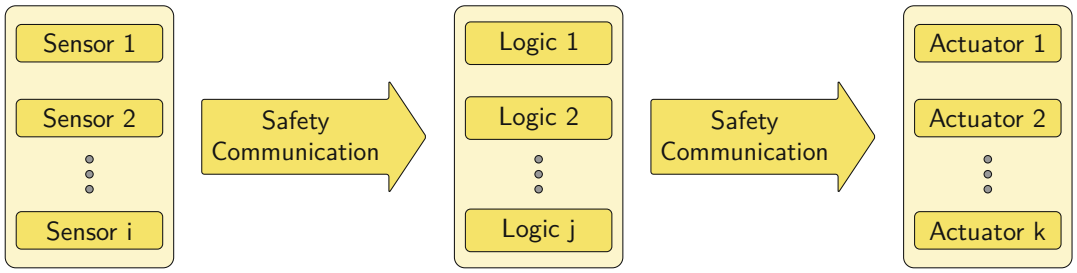


Figure 1.1: Functional safety system components

In this context, a concept of a flexible safety system is proposed which allows quick functional safety (re-)configuration changes even during operation. The concept takes into account not only the requirements flexibility and interoperability, but also reduces engineering efforts on configuration changes and consequently reducing machine downtime.

1.1 Motivation

The transition from industrial automation (Industry 3.0) to Cyber-Physical Production System (CPPS) (Industry 4.0) implies a huge demand for connectivity along the whole value chain. This chain encompasses vertical and horizontal communication from the sensor/actuator level up to enterprise resource planning (ERP) systems and further to cloud applications. Factories, in this context, comprised of a heterogeneous array of machines from a multitude of vendors and manufacturers, have their own specific demands on flexibility and interoperability which impose also higher requirements on functional-safety-related applications [18].

Flexibility and reconfiguration of manufacturing systems have been subjects of research for almost half a century. With the emergence of customized mass production as a goal of I4.0, Reconfigurable Manufacturing Systems (RMSs) have experienced a renaissance. Already in 1995, operational principles and design of RMSs were defined and formulated by Koren et al. [19]. The six core characteristics of ideal RMSs, shown in Table 1.1, are defined with the goal to reduce the time and cost of reconfiguration and thereby enhancing system responsiveness. Besides the configuration of machinery and their programs, this also affects the reconfiguration of the safety system [20].

Characteristic	Interpretation
Scalability (design for capacity changes)	The capability of modifying production capacity by adding or removing resources and/or changing system components
Convertibility (design for functionality changes)	The capability of transforming the functionality of existing systems and machines to fit new production requirements
Diagnosability (design for easy diagnostics)	The capability of real-time monitoring the product quality, and rapidly diagnosing the root-causes of product defects
Customization (flexibility limited to part family)	System or machine flexibility around a part family, obtaining thereby customized flexibility within the part family
Modularity (modular components)	The compartmentalization of operational functions into units that can be manipulated between alternative production schemes
Integrability (interfaces for rapid integration)	The capability of integrating modules rapidly and precisely by hardware and software interfaces

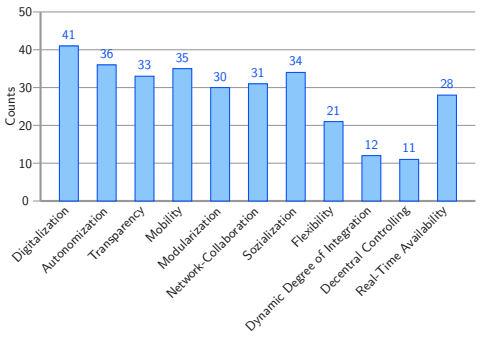
Table 1.1: Core characteristics of RMS [19]

Although the term I4.0 is not uniformly specified, there are several definitions using different characterizing features. In recent years, several studies have been carried out, based on the structured literature review, in order to find out characteristic features or design principles of I4.0. The results of the studies are not identical, but most of the identified characteristics are congruent. Table 1.2 shows the identified major features of four selected studies [21][22][23][24]. Three characteristic features or design principles, which are directly related to the significant properties and requirements of functional safety systems as part of modern machinery, are (1) network collaboration or interoperability, (2) flexibility, and (3) real-time capability.

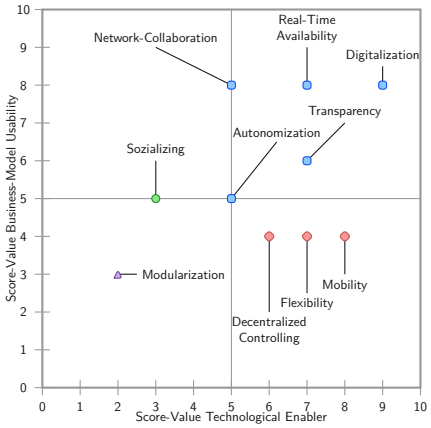
Pfohl et al. [21] 2015	Gilchrist et al. [22] 2016	Vogel-Heuser et al. [23] 2017	Perales et al. [24] 2018
Digitalization	**Interoperability**	Service orientation	Virtualization
Autonomization	Virtualization	Decentralization	**Interoperability**
Transparency	Decentralization	**Interoperability**	
Mobility	**Real-time capability**	Information aggregation and representation	Autonomization
Modularization	Service orientation	Virtualization	**Real-time availability**
Network-collaboration	**Modularity Flexibility**	**Real-time capability**	**Flexibility**
Socializing		**Flexible adaptation**	Service Orientation
Flexibility		Cross-disciplinary modularity	Energy efficiency
Dynamic degree of integration		Optimization of processes	
Decentral controlling		Data integration	
Real-time availability		Secure communication and data access	

Table 1.2: Characteristic features of Industry 4.0

Based on the results of an analysis of 49 technologies and concepts [21], shown in Figure 1.2a, Pfohl et al. developed an importance portfolio of characteristic features asking the questions (1) if features are enabling the other feature from a technological perspective and (2) if features of specific I4.0-related technologies and methods can or cannot be utilized in specific parts of the Canvas business model [25]. The results are illustrated in Figure 1.2b.



(a) Characteristic features [21]



(b) Importance portfolio [25]

Figure 1.2: Industry 4.0 characteristics

From an economic point of view, it can be recognized that network collaboration or interoperability has a high impact on the business model, whereas flexibility is a technological enabler. The basis for interoperability, which allows subsequent flexibility, is the standardization of architectures, data exchange formats, semantics, vocabularies, taxonomies, ontologies, and interfaces. Standardization is key in facilitating interoperability between devices from various vendors, especially when different communication technologies are involved in a complex, extremely heterogeneous field like I4.0 [26].

Manufacturing companies in Europe face two key challenges: strong global competition with pressure to move manufacturing to locations with cheap labour, and a necessity to address environmental challenges. In order to remain competitive in global markets, these manufacturing companies have to apply measures to increase the productivity of their production facilities, such as optimization of plant utilization and minimization of unplanned downtime. These measures target the "Six Big Losses" of the Overall Equipment Effectiveness (OEE): equipment failure, setup & adjustments, idling & minor stops, reduced speed, process defects, and reduced yield, as illustrated in Figure 1.3 [27].

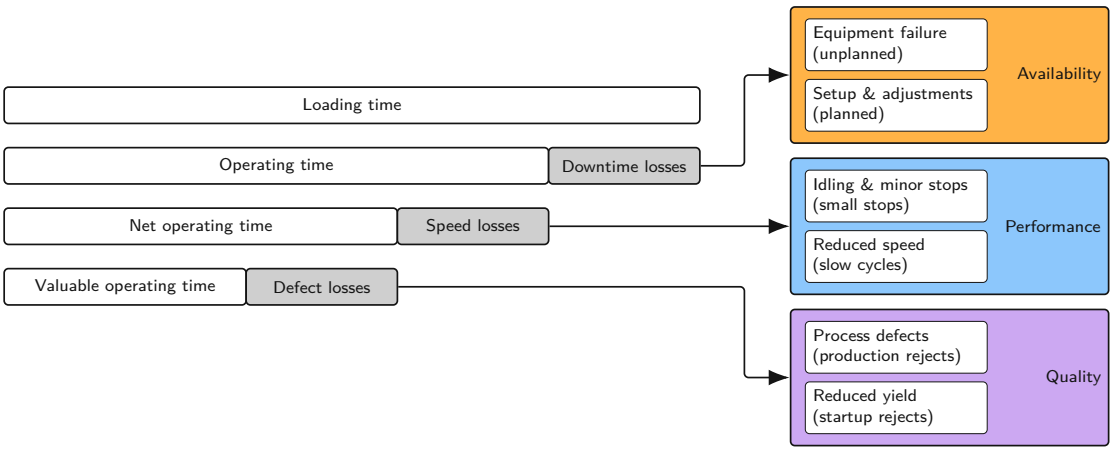


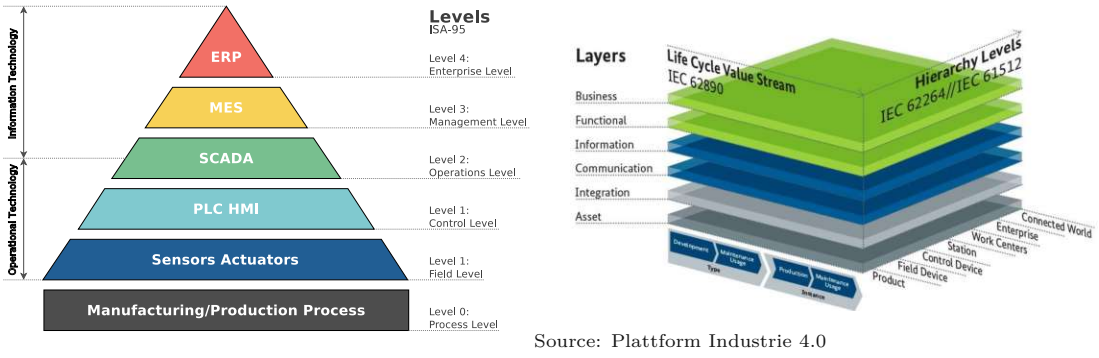
Figure 1.3: The six big losses of OEE

Planned and unplanned downtime can occur in highly automated scenarios. Reducing or avoiding those downtimes and thus increasing availability is imperative as there are adverse financial effects. Rapid reconfiguration can help reduce planned downtimes, whereas a worker assistance system can help to minimize unplanned downtime. Therefore, a safety system that provides worker assistance and supports rapid reconfiguration contributes to the improvement of the OEE of the production facility.

Two major driving forces for Industry 4.0 were identified by Lasi et al. in [28], application-pull and technology-push. The first, application-pull, induces a remarkable need for changes due to changing operative framework conditions. Triggers for these are short development and innovation periods, individualization on demand, higher flexibility in product development and especially in production, faster decision-making procedures

and therfore reduced organizational hierarchies. The second, technology-push, identifies innovative technologies such as further increasing mechanization and automation, digitalization of all manufacturing and manufacturing-supporting tools, increased networking of technical components, increase of the digitalization of produced goods and services, and a trend towards miniaturization.

Automation systems, as a central part of machinery in smart manufacturing, consist of interlinked devices and systems which can be logically grouped according to their degree of abstraction. This classification is well known as the "Automation Pyramid"[29], it is illustrated in Figure 1.4a. In the era of smart manufacturing, the concept of the Automation Pyramid is no longer sufficient enough, as it describes only the communication across the four levels. A more comprehensive way of modeling all aspects of smart manufacturing has been introduced by the Reference Architecture Model Industrie 4.0 (RAMI 4.0) [30]. This reference model considers three dimensions of I4.0: the product and plant life cycle, questions about the business idea, and the factory hierarchy. Figure 1.4b illustrates the three-dimensional structure of RAMI 4.0.



Source: Plattform Industrie 4.0

(a) Automation pyramid (b) RAMI 4.0 [30]

Figure 1.4: Reference models in automation

An essential part of smart manufacturing are production facilities that dynamically adapt to changing production needs. A prerequisite for this is a modular design of machines as well as robots and conveyors since the composition can change as needed during operation. This brings completely new challenges to functional safety systems, which are mandatory for the protection of man and machine. The components of a functional safety system in Industry 3.0 are located in Level 1 (Field/Control Level) of the Automation Pyramid, which includes safety sensors, safety actuators, and a safety PLC. As long as a machine produces things according to the same procedures, this model is sufficient, but when a machine produces different things in a different order with different forms or characteristics, a new approach is necessary. This new modular and dynamic approach has to address the need for flexibility and interoperability in the domain of functional safety. It includes the safety life cycle, risk assessment, certification, and safety communication, among others.

Besides safety sensors, safety actuators, and safety PLCs, a functional safety system in smart manufacturing must also take into account the product, production flow, safety communication, and safety configuration. RAMI 4.0 offers the possibility to create a model for each individual product, including the necessary safety configuration. Therefore, a safety configuration is not only bound to the machine but also to the product and production flow. These additional dependencies entail a very complex and time-consuming safety configuration. As a consequence, a self-organizing safety configuration generator is a fundamental prerequisite for an efficient, flexible safety system, as it allows to change the configuration of safety measures in a simple, quick, and, most importantly, safe way. The term self-organizing systems refers to a class of systems that are able to change their internal structure and their function in response to external circumstances [31]. In the context of flexible safety systems, as addressed in this work, self-organization specifies the ability to create (semi-)automatically a safety configuration triggered by changes within the system. As a consequence, not only the field and control device level in the RAMI 4.0 model are addressed, but also station, enterprise, and connected world. This approach influences the communication relationships among participating devices. The strict communication limitations within the automation pyramid, which allow only data exchange between neighboring levels, do not work any longer. The components of a flexible safety system require communication, including safety-relevant communication, among all involved levels and therefore flattening the system structures. Figure 1.5 illustrates the transition of communication patterns from Industry 3.0 to Industry 4.0.

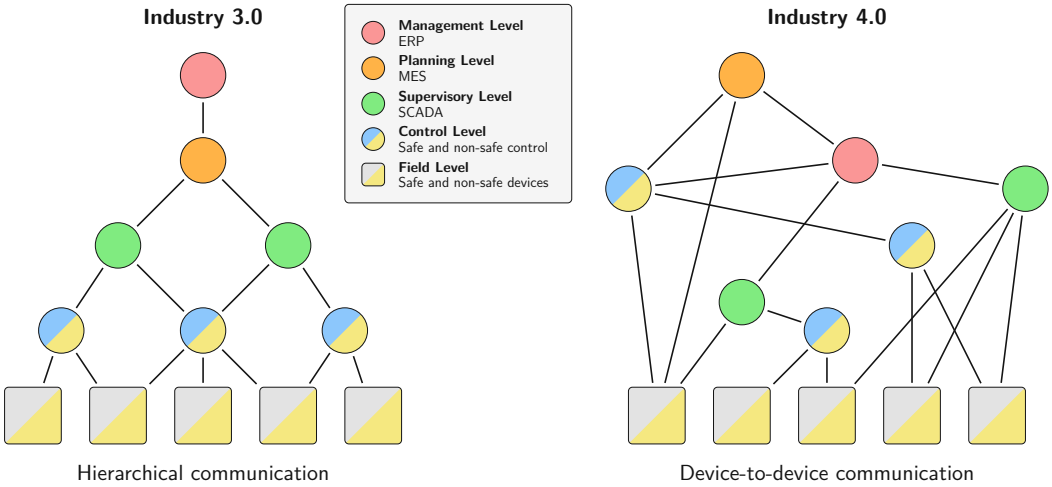


Figure 1.5: Transition of communication patterns (adapted from [32])

Current safety systems therefore need additional features and configuration assistance in order that the functional safety system does not become a hindrance for flexibility and reconfiguration of manufacturing systems.

1.2 Related Work

Although the topic of I4.0 is no longer new, there has been surprisingly little research on the topic of reconfigurable functional safety systems applied in I4.0 scenarios so far. The following literature study shall give an overview of related work in the fields of functional safety and worker assistance systems. The study was conducted with the databases: IEEE Xplore, Google Scholar, and Scopus between 2005 and 2023 using the keywords: functional safety, modular safety, interoperability, flexibility, real-time communication, smart manufacturing, and worker assistance.

The challenges of safety in I4.0 were already recognized by Liggesmeyer et al. in 2015. The authors analyzed uncertainties of production systems originating in the central element of I4.0 defined as a "network of autonomous, situational self-controlling, self-configuring, knowledge-based, sensor-supported, and spatially distributed production resources". These uncertainties are in conflict with the safety certification process, which expects a deterministic and predictable system behavior. For this reason, safety could easily become a bottleneck in the transition to I4.0 because innovation must never be at the expense of safety despite its high economic potential. Therefore, a concept is proposed based on a modular safety verification procedure that supports the safe and flexible composition of machinery. Additionally, an approach for runtime verification is presented to allow dynamic reconfiguration of a system during operation. The article concludes that even on the basis of the current state of science, the vision of I4.0 is not fully supported. If the safety of the systems can not be proven due to the lack of methods and technologies, the vision will ultimately not be realized [33].

In his paper "Functional safety and Industrie 4.0", Tom Meany explores some of the implications of functional safety for Industrie 4.0. The publication explains the design principles of I4.0, such as interoperability, virtualization, decentralization, real-time capability, service orientation, and modularity followed by a short summary of the three important key aspects networking, security, and robots. After that, functional safety is put into context with these three key aspects as well as with software and integrated circuits. The paper addresses the implications and challenges of functional safety in Industrie 4.0 without providing further solutions [34].

The challenges posed by I4.0 on safety systems are also addressed by [35], but from a different perspective. It explores key safety challenges and identifies the characteristics that safety assurance of I4.0 should exhibit. Therefore, the key characteristics are classified into three groups: modular and cooperative, continuous, and on-demand. If a safety system cannot be built by a single stakeholder or organization, the individual components must be modular and have to cooperate. Safety cases should be proactively reviewed and maintained to continuously demonstrate the safety performance status. Sometimes, however, updating the safety cases is not feasible because of the nature of the changes. Reconstructing the safety cases on-demand might be necessary as a more cost-effective option than updating the existing cases. Furthermore, a modular assurance approach is proposed, potentially addressing some of the presented challenges.

The topic of safety in flexible, dynamic environments arises also in related industries such as automotive. In [36], an exploratory study is conducted on which parts of ISO 26262 represent the most critical gaps between safety engineering and Machine Learning (ML) development. Although the topic is not directly related to I4.0, there is some common ground. The functional safety standard ISO 26262 (Automotive) is derived from IEC 61508, just like the I4.0 relevant standard IEC 62061 (Machinery). Furthermore, ML will play an important role in smart manufacturing applications as well as in self-organizing systems.

The reliable operation of a functional safety system in smart manufacturing facilities essentially depends on the underlying communication technology. Standardization is key to a unrestricted Machine-to-Machine (M2M) communication. Therefore, the authors of [37] address the coexistence of Operation Technology (OT) and Information Technology (IT), including their corresponding standardization bodies, within the same infrastructure. The different requirements on OT, which has the goal to provide a useful function in a reliable and efficient way, and IT with its strong focus on agility and flexibility to build data processing systems, are identified. Several aspects of converging OT and IT such as security, data handling, management processes, real time, and Ethernet-based networking protocols are discussed. The authors identify the possibility to get a common wired infrastructure based on Ethernet and time-sensitive networking (TSN) technology. They conclude that convergence will not happen overnight but will be an ongoing process, since the operational cycle of 10 years or even longer in OT. Additionally, for convergence to be a success, engineers need to learn to converge IT and OT technologies in requirements, designs, opportunities, and constraints.

The modularization of process plants with the need for flexible production options in the process industry is addresses by Pelzer et al. in [38]. The modularization of process plants addresses the need for flexible production options in the process industry. This requires the implementation of safety concepts without counteracting the dynamic characteristics of modular plants. To address this research area, newly developed concepts and engineering strategies for functional safety in modular plants have been developed. In the context of the modular plant, a Module Type Package (MTP) is a standardized interface for the simple integration of Process Equipment Assemblies (PEA). Its goal is the description of the automation equipment and the control of a PEA. It enables the process industry to use new methods for making plants more flexible. Since not all safety-relevant scenarios can be covered at any time and in advance, a collection of requirements for functional safety in modular systems is presented [39] [40]. Moreover, the changeability of modular process plants implies the need for fundamentally new safety strategies: Since operators must implement safety measures, assistance systems are needed to enable the operator to execute safety engineering tasks. Therefore, based on a state-of-the-art analysis of conventional safety engineering tasks, requirements to assist operators by a technical system were derived. Subsequently, the authors designed an assistance system and a module self-description to enable operators to implement the safety-related interconnection of modules [41].

The design of safety systems for modular plants concerning exchangeability and compatibility is challenging. For that reason, Klose et al. propose in [42] as a solution the development of a safety interface for PEAs to create flexible and scalable safety systems. Furthermore, in analogy to the modular automation, a standardized interface for safety related information and functions is proposed, the Safety-MTP. However, to be fully independent from the implementation of the PEA manufacturer, a standardized interface for safety-related information is pursued by the authors. The authors developed a concept for safe and modular interaction of PEAs in a modular plant. The concept from the previous phase was examined afterwards in the context of a demonstrator.

Due to the digitalization, interconnection, and constantly increasing complexity of manufacturing systems in the context of Industrie 4.0, the manual effort necessary to achieve the required safety and security are becoming ever more complex and almost impossible to manage.Therefore, Ehrlich et al. propose in [43] a (partially) automated risk assessment of modular systems concerning safety and security.The authors describe that in order to enable safety and security in a modular system today, all possible variants and configurations have to be considered and evaluated manually.The authors conclude that the current style of risk assessments is not adaptive enough for the upcoming developments and requires a lot of manual efforts by domain experts, which contradicts the foreseen flexibility of Industrie 4.0 applications.

In order to find possible negative effects of deviations from standard operation on safety and performance in complex automated systems, a systematical assessment has to be carried out. HAZOP (HAZard and OPerability) studies can be used for this purpose. However, HAZOP studies require significant manual effort and tedious work of several costly experts. Therefore, Schreiber et al. propose in [44] a knowledge-based approach to support the HAZOP analysis and to reduce the required manual effort by incorporating knowledge about typical problems in automation systems in combination with their causes and their effects in a rule base. Subsequently, this rule base is applied by means of a rule engine on the description of the automated system under consideration.

The technological and computational developments of recent years, since the beginning of I4.0, have led to rapid progress in the development of technical worker assistance systems to support workers in their daily work. Mark et al. present in [45] a systematic literature review of worker assistance systems in manufacturing. The authors state that factories in the future will not be without humans, not even by introducing I4.0. People will work or operate with robots, machines, and other humans. In addition, both the products and the manufacturing environment are becoming increasingly complex. Therefore, it is necessary to support the employee with available assistive technologies in order to cope with the increasing diversity of work tasks and the complexity of industrial production. The authors conclude that the implementation and use of worker assistance systems can be an efficient strategy for companies to create advantages in manufacturing and to provide a better well-being for the employees. In addition, in times of shortage of qualified workers, worker assistance systems can also be used by companies to increase the attractiveness of jobs in manufacturing production.

1.3 Problem Statement and Research Questions

The increasing complexity of machinery in smart factories leads to very complex and time-consuming safety configurations. Production facilities in a smart manufacturing environment pose high requirements to flexibility because they have to dynamically adapt to changing production needs.

Currently, functional safety systems are designed and certified in a static way. The safety design and the configuration are derived from the risk assessment which is performed during the design of a machine. Once the system is put into operation, the safety configuration is not changed anymore. This approach constitutes an impediment to flexibility which smart manufacturing production facilities require. There is an enormous contradiction between current functional safety systems and desired flexibility of production facilities.

This research aims at designing a flexible safety system with the objective to assist an engineer who operates a smart manufacturing facility. Before developing such a system, some research questions must be clarified in advance.

RQ1: Which properties are essential for a safety communication platform that enables flexibility and interoperability of Industry 4.0 production facilities?

RQ2: What are the technical prerequisites for management and configuration that allow rapid reconfiguration of future functional safety systems?

RQ3: Which components and specific features are needed for an architecture that enables a functional safety system for future production facilities?

The answers to these research questions will build the solid foundation for designing a flexible safety system which fulfills the requirements of smart manufacturing on the one hand, and conforms to the relevant legislative standards on the other hand.

1.4 Structure

The document structure largely follows the path of the defined methodological approach in Chapter 3 and is organized as follows.

Chapter 1 gives an introduction to the topic and describes the motivation for this research. Related work is listed, and the problem statement as well as research questions are defined. After describing the structure, the scientific contributions are listed.

Chapter 2 presents the state of the art, starting with an overview of existing legal regulations and relevant standards. After explaining functional safety and its life cycle, including risk assessment, the specific properties of safety communication are discussed.

Furthermore, today's possibilities for functional changes regarding safety-related configurations during operation are presented. Requirements and classes of safety-related software tools conclude the chapter.

Chapter 3 defines the methodologies used in this research. An adapted system development life cycle is presented, which is derived from the System Development Life Cycle (SDLC). Moreover, the methodological approaches utilized in the individual phases of the system development life cycle are briefly discussed.

Chapter 4 investigates the role of safety in smart manufacturing and presents use cases that illustrate the need for flexibility, interoperability, and rapid reconfiguration in I4.0 production facilities based on determined requirements. As a result of the analysis, an abstract safety system model, configuration aspects, and the derived requirements will be the basis for technology integration and the design in the following steps.

Chapter 5 examines existing industrial communication technologies for suitability as a solid bedrock for flexible safety communication and proposes a communication platform for vendor-neutral and standardized safety-critical communication based on these technologies. Furthermore, the importance of assistance in a highly complex system and, therefore, the need for a comprehensive tool in the configuration process is emphasized.

Chapter 6 is dedicated to the design of a Flexible Safety System (FSS) and constitutes the core of this work. A concept for a system design based on the results of previous analysis and safety communication platform is presented. It includes safety devices, data transport, and the safety configuration procedure. It shows how existing base technologies can serve as a foundation for interoperable real-time communication and how the proposed system takes over the complex configuration of these technologies. Moreover, essential system services are presented as necessary for use in an industrial production environment.

Chapter 7 derives a software architecture model from the outcomes of the previous chapters. Architecture artifacts for context, containers, components, and code are created. Those artifacts are based on the previously proposed modular system design and the examined base technologies.

Chapter 8 covers the evaluation of the proposed software architecture. It is ensured that the architecture with its services, functions, and interfaces meets the identified needs of the stakeholders and whether or not the architecture with its containers and components complies with the previously analyzed requirements and other specifications.

Chapter 9 concludes the thesis by presenting the achieved outcomes and summarizing the answers to the research questions. Additionally, ongoing work in the project is presented, and a brief reflection offers some insights on the project so far. Finally, a list of possible system extensions outlines possible enhancements for the future.

1.5 Scientific Contributions

The following publications and scientific work emerged during the work on this thesis at the Automation Systems Group (ASG) at TU Wien and at the Center for Digital Production (CDP).

Self-Configuring Safety Networks

In the context of Industry 4.0, production lines as part of Cyber-Physical Systems (CPSs) have specific demands for interoperability and flexibility. Machinery, being part of these production lines, has additional requirements in terms of functional safety and real-time communications. The re-configuration of functional safety systems, which is characterized by high manual configuration efforts, leads to time-intensive and expensive downtimes. This paper presents the requirements on and the concept of self-configuring safety networks, which reduces the engineering efforts and allows the operator of production lines convenient re-configuration of safety functions and devices. The proposed concept is based on the vendor-neutral technologies Ethernet, Time-Sensitive Networking (TSN), and OPC Unified Architecture (OPC UA).

Conference paper at Kommunikation in der Automation (KommA) 2018 [46]

Simplifying functional safety communication in modular, heterogeneous production lines

Heterogeneous production lines as a keystone of smart factories, comprised of machines from various manufacturers, are placing a new range of demands on communication and interoperability. An important aspect of a production line, is functional safety and its technical implementation. Up to now, safety-relevant connections between devices of different manufacturers have been implemented using dedicated cables and line monitoring. The aim of this paper is to design an integrated safety architecture based upon existing technologies. The method proposed in this paper aims at achieving functional safety connectivity, along with non-safe data traffic, based on the vendor-neutral technologies Ethernet, Time-Sensitive Networking (TSN), and OPC Unified Architecture (OPC UA).

Conference paper at Int. Workshop on Factory Communication Systems (WFCS) 2018 [18]

Flexible Safety Systems for Smart Manufacturing

Smart manufacturing is realizing the idea and potential of Industry 4.0 in reality. An essential part of smart manufacturing are production facilities that dynamically adapt to changing production needs. This brings completely new challenges to functional safety systems, which are mandatory for the protection of man, machine, and environment. Currently, functional safety systems are designed and certified in a static way. This paper proposes the design of a self-organizing safety system with the objective to assist an engineer who operates a smart manufacturing facility by discovering all safety-related devices and generating automatically a suitable safety configuration. The proposed self-organizing safety system, simplifies the safety configuration in a dynamically changing environment. Consequently, it would reduce engineering efforts and decrease machine downtime which improves profitability.

Conference paper at Int. Conference on Emerging Technologies and Factory Automation (ETFA) 2020 [47]

Building Blocks for Flexible Functional Safety in Discrete Manufacturing and Process Industries

Discrete manufacturing and process industries move towards flexible production to cope with individualization. In this paper, a generalized methodology to draft an overall safety concept using different technologies is presented. Motivated by use-cases from both domains, requirements for future safety systems are described and summarized in a structured way. These requirements are then generalized and classified in building blocks to fulfill the needs of flexible functional safety. A methodology to structure safety systems and their general parts is presented. The used building blocks refer to aspects as interfaces, communication, and certification.

Conference paper at Int. Conference on Emerging Technologies and Factory Automation (ETFA) 2021 [48]

Smart Manufacturing Retrofit for Brownfield Systems

In recent years, the concept of Industry 4.0 and smart manufacturing have caused disruption and upheaval in many industries. These concepts promise high efficiency and flexibility but also require seamless communication and interoperability among machines. Brownfield systems, that do not satisfy these requirements, need a functional extension or enhancement in order to remain competitive. This paper proposes a solution for retrofitting a brownfield system on the example of an industrial robot.

Conference paper at Int. Conference on Industry 4.0 and Smart Manufacturing (ISM) 2019 [49]

Human Control of Self-Organizing Safety Systems

In the context of Industry 4.0, production lines as part of Cyber-Physical Production Systems (CPPS) have specific demands for interoperability and flexibility. Machinery, being part of such production lines, has additional requirements in terms of functional safety. The concept of self-organizing safety systems reduces the engineering efforts by assisting the system operator in the configuration of changes and consequently reducing machine downtime. The aim of this paper is to present a self-organizing safety system model and discuss the human control of self-organizing safety systems by making use of cybernetic approaches and examining the backgrounds for the need of human control over autonomous systems.

Conference paper at Int. Conference on Societal Automation (SA) 2020 [50]

Functional Safety Use Cases in the Context of Reconfigurable Manufacturing Systems

Flexibility and reconfiguration of manufacturing systems have been subjects of research for almost half a century. In recent years, with the emergence of customized mass production as a goal of Industry 4.0 (I4.0), Reconfigurable Manufacturing Systems (RMSs) have experienced a renaissance. In this paper, core characteristics of Reconfigurable Safety Systems (RSSs) are proposed, which extend the characteristics of RMSs. Furthermore, we explore use cases of discrete manufacturing and identify service groups and services that are needed for the technical implementation of rapid safety reconfiguration within RMSs in order to allow flexible, reliable, and safe operation of machinery.

Conference paper at Int. Conference on Emerging Technologies and Factory Automation (ETFA) 2022 [20]

1.6 Goals and Objectives

The primary goal of this thesis is a system design and software architecture for a functional safety system capable of rapid reconfiguration aiming at minimizing planned and unplanned downtimes of production facilities. Currently, no comparable system exists, and therefore, the successful realization of the plan is a major technological innovation.

Additionally, the proposed system design and software architecture consider modularity, flexibility, and interoperability as secondary goals to address the requirements of Industry 4.0 production facilities.

The objectives to achieve the defined goals are analyzing functional safety systems and use cases to derive requirements, preparing a safety communication platform, and creating a system design and software architecture for a Flexible Safety System (FSS).

As the follow-up activities of this work, the system design and the software architecture will be realized as a technology demonstrator and subsequently in a proof-of-concept implementation. However, this is outside the scope of this thesis.

CHAPTER 2

State of the Art

> THE ARGUMENT THAT THE SAME RISK WAS FLOWN BEFORE WITHOUT FAILURE IS OFTEN ACCEPTED AS AN ARGUMENT FOR THE SAFETY OF ACCEPTING IT AGAIN. BECAUSE OF THIS, OBVIOUS WEAKNESSES ARE ACCEPTED AGAIN AND AGAIN, SOMETIMES WITHOUT A SUFFICIENTLY SERIOUS ATTEMPT TO REMEDY THEM, OR TO DELAY A FLIGHT BECAUSE OF THEIR CONTINUED PRESENCE. [51]
>
> – Richard P. Feynman, 1986

The topic functional safety in smart manufacturing, especially regarding machinery, comprises various aspects. This section shall give a brief overview about the concept of machinery safety, some administrative and legal aspects, the role of standardization, the importance of risk assessment, the definition of functional safety, and the specific characteristics of safety communication. Furthermore, it explains why flexibility and interoperability are indispensable for smart manufacturing. The section concludes with requirement on base technologies with a listing of some promising candidates.

2.1 Machinery Safety

In an industrial setting, people interact with machines that are designed to process or transport materials. In such a workplace, hazards occur when the working environment can cause injury, illness, or death. These hazards can result from many aspects of the working environment, including equipment, dangerous materials, unsafe working practices, and the negligent behavior of people. Additionally, workplaces change over time and continoue to change due to new technologies, automation, and workflow. For example, the emergence of collaborative robots in recent years, the increased use of Automated Guided Vehicles (AGVs), or the flexible reconfiguration of production facilities, introduced

new risks and hazards. The often-discussed peopleless factory is still far in the future. Therefore, it is important to understand the issues surrounding risks and hazards at work and create means of mitigating the risks from them [4].

Hazards in industrial environments are categorized in Table 2.1 [4][52].

Category	Hazards
Safety-related hazards	▪ Mechanical hazards ▪ Slips, trips, falls, and vision ▪ Electrical hazards ▪ Automation and robots
Physical hazards	▪ Temperature ▪ Pressure ▪ Radiation ▪ Noise and vibration
Chemical hazards	▪ Liquids ▪ Flammable substances ▪ Harmful gases
Biological hazards	▪ Viruses, bacteria, insects, and animals ▪ Plants ▪ Bloodborne pathogen
Ergonomic hazards	▪ Poor posture ▪ Heavy lifting ▪ Improper work station
Psychosocial hazards	▪ Work load ▪ Stress ▪ Violance

Table 2.1: Hazard categories

The concept of machinery safety determines all possible hazards that can emanate from a machine and considers the ability of a machine to perform its intended function(s) during its life cycle where risk has been adequately reduced [53]. In other words, safety is the "freedom from unacceptable risk of physical injury or of damage to the health of people, either directly, or indirectly as a result of damage to property or to the environment", IEC 61508 [10]. Regardless of the development in different industries, this main perspective remains the same, but boundary conditions and applications differ. Whilst safety systems began with mechanical applications to prevent processes from leaving a desired operating range, electronic safety systems like Safety Instrumented Systems (SISs) have come to the fore in the last 20 years. Implementations in discrete manufacturing or process industry move towards smart, modular, and interconnected ecosystems [48].

Regarding functional safety systems for reconfigurable production facilities in smart manufacturing, the categories of hazards can be limited to the safety-related hazards: mechanical, automation, and robots. These hazards, and the appropriate safeguards, will be elucidated in the following. All other hazards, such as noise, vibration, or temperature, are not subject to reconfiguration and therefore are not further addressed here.

2.1.1 Mechanical Hazards

Concerns about mechanical hazards that are associated to power-driven machines date back to the first industrial revolution and the earliest days of mechanization. In an industrial environment, people interact with machines that are designed to drill, cut, shear, punch, chip, staple, stitch, abrade, shape, stamp, and slit different kinds of material. These procedures can cause mechanical injuries to humans and therefore require safeguarding to minimize the risk of accidents at machine-operator contact [4].

When considering the most appropriate method of safeguarding to provide the required degree of protection, a distinction is made between solutions using guards (passive safeguards) and protective devices (active safeguards) [52]:

1. the provision of a guard or physical barrier, and
2. the provision of protective devices that allow access into the machine but ensure that the dangerous parts are no longer dangerous when they can be reached.

Guards, or passive safeguards, prevent people accessing or intervening in a machine's danger zone. There is a range of guard types, from which to make a selection, that will provide the required degree of protection within the operational requirements of a machine [53]. The most commen one are [52]:

- A **fixed guard** is one that is held in place by devices that require a tool to release them. The securing devices can be bolts, screws, nuts, key operated deadlocks or other device acceptable to the enforcing authority.
- A **movable guard** is a fixed guard that can easily be moved out of position but cannot be detached from the machine. It must be designed so it can be consistently returned to its proper location.
- A **removable guard** is a fixed guard that can be completely removed from the machine. It must be designed so that it can, when replaced, be located properly in its correct position and fixed securely to the machine by captive bolts or other locking means requiring a tool to release them.
- An **adjustable guard** is a fixed guard whose position as a whole can be adjusted to suit the particular operation being carried out. Once the adjustment has been made the guard elements must be firmly secured.
- An **interlocking guard** is a movable or removable guard which when the guard is moved from its safe position actuates a device that causes the machine to go to a safe state. It should be positioned to cover those parts of the machine to which immediate access is necessary as part of the normal operating process.

- A **control guard** is a guard that, when it reaches its closed or safe operating position, initiates the machine operating cycle. Its application should be restricted to cyclic processes with a short cycle time, typically single stroking press brakes, where there is a single operator only.
- A **self-adjusting or self-closing guard** is a movable guard which covers the dangerous part of the machine but is moved by the material being worked on to allow the particular process to be carried out. It then reverts automatically to the safe position when the operation has been completed.
- An **automatic guard** is used on single-cycle operations and is actuated by the operator striking on. This causes the guard to move to the safe position when it trips a switch initiating machine movement.
- A **tunnel guard** consists of a fixed guard in the form of a duct between the feed or take-off point and the dangerous part of a machine. It must be of a cross sectional size to accommodate the product being processed and long enough to prevent the operator reaching the dangerous part.
- A **spiral guard** is a special purpose guard designed for application to revolving shafts to accommodate longitudinal movement along the shaft of parts of the machine. It is secured at each end to parts of the machine and can be fitted to a shaft in situ.
- The term **fence** is applied to a guard that is permanently fixed to the floor, is of a height and is set at a distance from the dangerous parts of the machine such that they cannot be reached. The fence can protect part of a machine or completely surround it.

Protective devices, or active safeguards, belong to the category of barrier-free safety solutions and secure the danger zone without any physical separation. Control solutions are often used here, such as Electrosensitive Protective Equipment (ESPE) in conjunction with a safety controllers, and a safety actuators. ESPE devices can implement detection of persons through various principles. In practice, optoelectronic protective devices (e.g. light curtains, laser scanners or camera-based protection systems) and safe radar systems are used. The most common protective devices are [52]:

- **Interlocking devices** that are actuated as the guard moves from the safe position, can be provided to increase the level of protection on machinery to which movable guards have been fitted.
- Interlocking with **guard locking** should be applied where it is essential that the power is isolated before the guard can be opened.
- An **interlocking system**, whether electrical, pneumatic or hydraulic or a combination of these, can be either control interlocking (machin control brings machine in a safe state) or power interlocking (machine isolated from its source of power).
- A **two-hand control device** can provide protection on single-operator machines where guarding of the feed position is not feasible.
- Special **hold-to-run controls** should be provided at machines which, of necessity, need to be run with the guards open to facilitate adjustment and setting.

- **Limited movement control** is an alternative arrangement to hold-to-run whereby a machine may be run with the guards open but moves only a limited distance for each control actuation.
- **Person-sensing devices** or rather immaterial or intangible barriers can be applied to machines where for operational reasons the provision of physical guards is not feasible. Known as electro-sensitive protective equipment (ESPE in IEC 61496 parts 1 to 4) or personnel sensing protective equipment (PSPE in IEC 62046). The sensing medium can be photo-electric, ultrasonic, radar, laser or other electromagnetic emitter with a suitable detection capability and level of safety integrity.
- **Emergency stop switches** should override all other controls to bring the machine to a stop. All machines should be provided with an emergency stop switch unless it can be shown that such a device would not contribute to minimizing the risk.
- **Grab wires** can provide a degree of protection on long machines, such as flat belts and troughed belt conveyors, where hazardous points occur at frequent intervals along the whole length of the machine, and neither fixed guarding nor safeguarding devices, such as photo-electric curtains, are feasible.
- A **pressure sensitive mat** laid across the approach floor to a machine provides protection against inadvertent entry to the hazard zone.
- **Pressure sensitive edges and wires** can be used in applications where there is a risk of trapping between powered closing doors.

Unlike fixed guards or physical barriers, protective devices enable temporary barrier-free entry into a potential danger zone. The protection is not based on the physical separation of persons at risk from the risk itself, but on the localization of people or parts of the body. This approach offers many advantages, especially for dynamic work environments like those in smart manufacturing. These advantages are: reduced access time (operator does not have to wait for the protective device to open), increased productivity (time savings when loading the machine), and improved workplace ergonomics (operator does not have to operate a physical guard) [54].

2.1.2 Automation and Robots

Each introduction of new technologies or tools has changed the workplace to make it more effective and efficient. But it also brought along new safety and health hazards. Since the advent of automation in the 1960s, the emphasis changed from physical to automated mechanical and mental work. This led to increased levels of stress experienced by workers due to the increased rate of change in the automated production process.

In modern industry robots are widely used. Consequently, there is plenty of human-robot interaction. The potential dangers of robots originate in their ability to acquire intelligence through programming, their flexibility and range of motion, their speed of movement, and their power. The principal hazards associated with robots are being struck by a moving robot while inside the work envelope, being trapped between a moving part of a robot and another machine or object, and being struck by a workpiece dropped by a robot. The best guard against these hazards is to erect a physical barrier

around the entire perimeter of a robot's work envelope. However, such a solution restricts the scope of the application and therefore requires a flexible approach with protective devices. Additionally, collaborative robots, which interact with humans directly, have to be considered as well.

As human workers continually increase the amount of their interaction with automated machines and robots, the potential for maladaptation also increases. Therefore, automated systems have to be designed around the needs of humans in order to prevent maladaptation, which also includes the application of worker assistance systems [4].

2.2 Administrative and Legal Aspects

Dangerous machinery that could cause serious injuries to workers has been used in factories since the first industrial revolution in the 18th century. It was only in the 1970s, in the third industrial revolution, that safety regulation was introduced. Today, legal and regulatory frameworks are in place in all industrialized countries in order to ensure a common safety level in machinery placed on the market or put in service.

Within the lifetime of a machine, the legal responsibility regarding safety is assigned to two roles, the manufacturer and the operator. Until delivery and commissioning of a machine, the manufacturer has to ensure the compliance with safety regulations. For the rest of the machine's lifetime, the operator is accountable for the compliance with safety and health regulations, this includes commissioning, operation, decommissioning, and disposal of the machine. Figure 2.1 shows the roles over the lifetime.

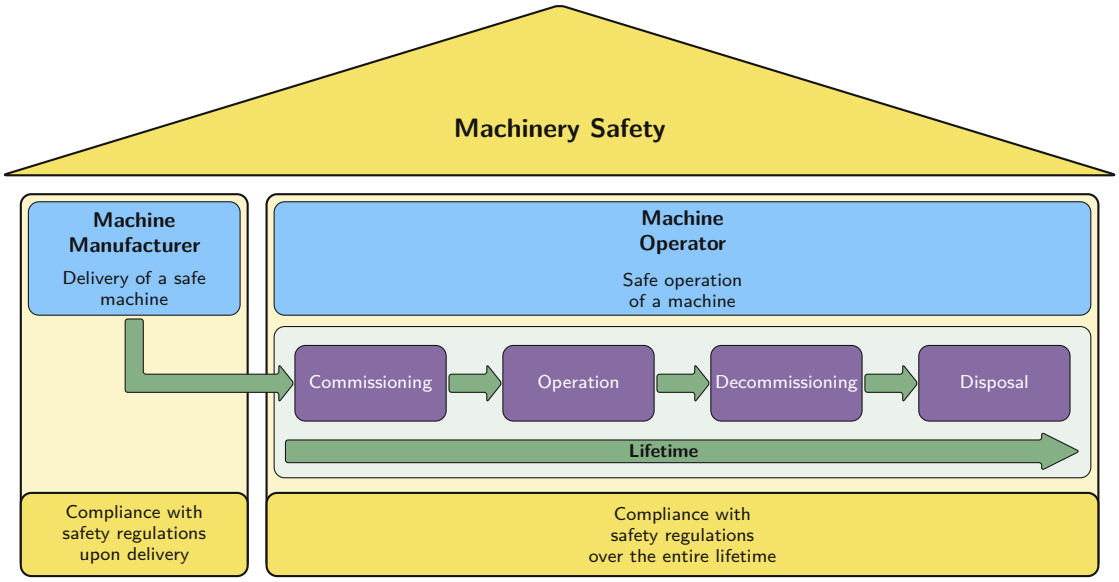


Figure 2.1: Responsibilities

The manufacturer's and operator's task is to make a machine, production line, or plant as safe as possible by applying the relevant standards and using state-of-the-art technology. In this way, the manufacturer and the operator have fulfilled their responsibility for ensuring the safety of the used production facilities and resources.

The applicable laws depend primarily on the location where a machine will be operated. The legislation differs significantly between continents, countries, or even cities. Therefore, the manufacturer, the exporter/importer,the distributor, and the operator of a machine have to know the legal requirements and which laws and regulations are to be applied, especially in the global market.

Laws regarding machinery safety can be divided into three kinds:

- **Administrative law**, determine consequences without suffering any damage including the enforcement of regulations by administrative agencies. (e.g. for Austria: Gewerbeordnung (GewO), Maschinen-Sicherheitsverordnung (MSV), ArbeitnehmerInnenschutzgesetz(ASchG), Arbeitsmittelverordnung (AM-VO))
- **Criminal law**, determine consequences after suffering a damage. (e.g. for Austria: Strafgesetzbuch (StGB))
- **Civil law**, define compensation after an individual has been injured physically. (e.g. for Austria: Allgemeines bürgerliches Gesetzbuch (ABGB), Allgemeines Sozialversicherungsgesetz (ASVG))

In general, requirements for machine and plant safety on a global view can extremely vary. The machine and plant manufacturer must therefore deal with the special country-specific laws, directives, and standards relevant for the product. This thesis is mainly concerned with European standards, directives and laws. However, the following section provides also a brief overview of the situation outside of Europe.

2.2.1 Europe

For any manufacturer or distributor that wants to sell their products in the EU or the European Economic Area (EEA), it is obligatory to comply with the relevant European directives or regulations. Failure to comply can result in fines and product recalls and could lead to being banned from distributing in Europe.

European Machinery Directive 2006/42/EC

Within the EEA the Machinery Directive 2006/42/EC regulates a uniform level of safety for machines in order to enable free movement of machinery within the single market and guarantees a high level of protection for EU workers and citizens. It applies to manufacturers and distributors of machinery and devices, but not to operating companies. This directive aims at the free market circulation on machinery and at the protection of workers and consumers using such machinery. It defines essential health and safety requirements of general application, supplemented by a number of more specific requirements for certain categories of machinery. It applies to machinery, interchangeable

equipment, safety components, lifting accessories, chains, ropes and webbing, removable mechanical transmission devices and partly completed machinery as defined in Art. 2 of the directive [55]. On 15th December 2022, the Council and the European Parliament negotiators have reached a provisional agreement on the regulation for machinery products. The proposed legislation transforms the 2006 Machinery Directive into a regulation. The new regulation aims to harmonise health and safety requirements within the EU for machinery products and tackle the challenges arising from new technologies. The regulation covers machinery products intended for consumers and industrial machinery, ranging from heavy-duty construction machines to entire industrial production lines, as well as highly digitalised products like robots or manufacturing 3D-printers. The aim of transforming the directive into a regulation is to constitute a legal framework that is directly applicable in all members states and is clear for all economic operators. The provisional agreement is subject to approval by the Council and the European Parliament. After the formal steps of adoption have been completed, member states will have 42 months to apply the rules of the Regulation [56].

European Directive 2009/104/EC – use of work equipment

The aim of Directive 89/391/EEC - OSH "Framework Directive" is to introduce measures to encourage improvements in the safety and health of workers at work. It applies to all sectors of activity, both public and private, except for specific public service activities, such as the armed forces, the police or certain civil protection services. This framework directive contains several specific directives. One of these specific directives is Directive 2009/104/EC of the European Parliament and of the Council of 16 September 2009 concerning the minimum safety and health requirements for the use of work equipment by workers at work which lays down minimum safety and health requirements for the use of work equipment by workers at work. It provides definitions of terms “work equipment”, “use of work equipment”, “danger zone”, “exposed worker” and “operator” [57].

It regulates the obligations of the employer and logically requires that the employer ensures that safety and the protection of health are guaranteed with the operation of the work equipment provided. This includes inspection of the work equipment before first use and throughout the period of usage. This directive and the implementation in local laws, like local regulations, applies to operating companies of machinery and devices.

CE Marking

Irrespective of the place and date of manufacture, all machinery used in the EEA for the first time from 1st January 1995 is subject to the EU Machinery Directive and as such must be "CE" certified. The letters "CE" are the abbreviation of "Conformité Européenne" (French for "European conformity"), they signify that products sold in the EEA have been assessed to meet safshall be used to protect persons whenever an inherently safe design measure does not reasonably make it possible either to remove hazards or to sufficiently reduce risks.ety, health, and environmental protection requirements.

The product's manufacturer bears sole responsibility for declaring conformity with all relevant EU-wide requirements. Therfore, the manufacturer has to perform the following tasks: categorise the product, check the application of additional EU directives, ensure that safety regulations are met, perform the risk assessment, carry out the validation, compile the technical documentation, issue the declaration of conformity, affix the CE mark as depicted in Figure 2.2.

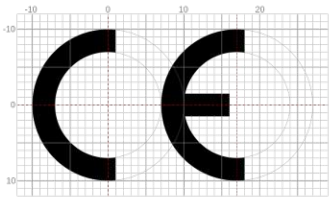


Figure 2.2: CE mark

2.2.2 United States

In Northern America, machinery safety is ensured mainly by a mixture of product standards, fire codes, electrical codes, and national laws. There is no uniform standards system in the US with one issuer of standards as is the case in Europe with CEN/CENELEC and the EN standards [58]. However, there are a number of accredited drafters that can develop and publish standards. These organizations can be categorized broadly as follows: the government, professional organizations, technical/trade organizations, and testing organizations. The most prominent regarding machinery safety are: Occupational Safety and Health Administration (OSHA), American National Standards Institute (ANSI), National Fire Protection Association (NFPA), and Underwriters Laboratories (UL) [4].

Occupational Safety and Health Administration (OSHA)

Occupational Safety and Health Administration (OSHA) is part of the United States Department of Labor. The administrator for OSHA is the Assistant Secretary of Labor for Occupational Safety and Health. OSHA's administrator answers to the Secretary of Labor, who is a member of the cabinet of the President of the United States. The most widely applicable OSHA standards are the General Industry Standards. These standards are found in 29 CFR 1910 and consist of Subparts A-Z [4, p. 139-149].

American National Standards Institute (ANSI)

The American National Standards Institute (ANSI) is a private nonprofit organization that accredits Standards Developing Organizations to develop voluntary consensus standards according to rigorous procedures and approves those resultant American National Standards.[1]

[1] https://www.b11standards.org

B11 Standards, Inc. is one of the ~230 ANSI-accredited Standards Developing (ASD) organizations in the United States that administers the process through which the ANSI B11 series of American National Standards and Technical Reports on machine/machine tool/machinery safety are developed and approved. In total, there are 26 ANSI B11 standards that specify requirements for both the manufacturers (suppliers) and users of the machines.[2]

National Fire Protection Association (NFPA)

The National Fire Protection Association (NFPA) is a US-based international non-profit organization devoted to eliminating death, injury, property and economic loss due to fire, electrical and related hazards. The NFPA publishes more than 300 consensus codes and standards intended to minimize the possibility and effects of fire and other risks. All NFPA codes and standards are periodically reviewed.[3]

Underwriters Laboratories (UL)

The Underwriters Laboratories (UL) is a US-based international safety science company composed of three organizations, UL Research Institutes, UL Standards & Engagement and UL Solutions. The company is one of several companies approved to perform safety testing by the US federal agency OSHA.

Regarding standardization and certification of machinery UL operates two unique entities, one being an independent body focused on standards development and a second Nationally Recognized Testing Laboratories (NRTL) that is capable of issuing a US product certification. In the US, UL certification is voluntary for all electrical control devices or systems and is obligated on component parts that are part of a larger product or system. UL certification is a document confirming that a product is safe to be used and complies with UL regulations. Upon successfully completing UL certification, products can carry the a corresponding UL mark.[4]. Examples for such UL marks are shown in Figure 2.3.

Figure 2.3: Examples of UL functional safety marks

2.2.3 Russian Federation

Since 2009 the decree "N753 Decree of the Government of the Russian Federation – Technical Regulation (TR) on safety of machines and equipment" is into effect. It stipulates

[2] https://blog.ansi.org/2020/07/ansi-b11-standards-safety-of-machinery/
[3] https://www.nfpa.org
[4] https://marks.ul.com/about/ul-listing-and-classification-marks/

basic minimum requirements for safety for machines and equipment as well as a mandatory conformity assessment procedure combined with a certification procedure. With an obligation for certification, the machine must be checked by a locally accredited test laboratory and a TR certificate must be issued [58]. Additionally, Eurasian Conformity (EAC) marking is the prerequisite for the initial placing on the market in all member-states of the Eurasian Economic Union. The EAC mark, shown in Figure 2.4, indicates products that conform to all technical regulations of the Eurasian Customs Union.

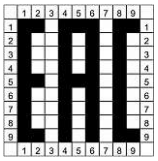


Figure 2.4: EAC mark

2.2.4 China

In China the State Administration of Work Safety is responsible for defining and monitoring health and safety measures. Chinese machine safety standards are used for plant and machinery. The Standardization Administration of China (SAC) is responsible for the independent national standards system. However, in the area of machine safety, the SAC generally adopts international ISO and IEC standards [58].

China has introduced CCC certification, which stands for China Compulsory Certificate. Certain technical product groups are subject to mandatory certification through a national approval body. The CCC mark, depicted in Figure 2.5 is used to mark certified products.

Figure 2.5: CCC mark

2.2.5 Japan

The Japanese Industrial Safety and Health Law specifies that hazardous machinery or machinery that is used in hazardous environments must be equipped with appropriate protective measures, as defined by the national Ministry. The Japanese Industrial Safety & Health Association (JISHA) is responsible for the introduction of risk evaluations and health and safety management systems in Japan.

Due to Japanese law, each employer/machine user must ensure that industrial safety measures are in place. The methodology for risk assessment is stated in the Japanese

Industrial Standards (JIS), which are harmonised with international standards. Japan generally adopts the IEC and ISO standards as national JIS standards. [58].

The Product Safety Electrical Appliance and Material Safety Law (PSE) and its regulations specify mandatory electrical safety and Electromagnetic Interference (EMI) requirements for electrical products sold in Japan [59]. Products with a history of accidents in the marketplace, or products which are likely to cause injury, are termed "Specified Electrical Products". These products must display the diamond PSE mark depicted in Figure 2.6.

Figure 2.6: PSE mark

2.3 Standardization

Standards and standardization are broad terms that are covering many entities and that are sometimes differently defined. We define standardization as the process of rendering things uniform, and standard as both the means and outcome of standardization. [60]

Standards can be seen as generalized and formalized rules that serve to prescribe and document efficiency and control within and across organizations. [61]

The economic importance of technological standards has grown tremendously over the past two decades. The growing recognition of the importance of the standardization process has been attributed in large part to the growth of the information technology and communications industries, for which standards are critical [62]. The rapid developments in Internet of Things (IoT) and Machine-to-Machine (M2M) communication make it necessary to design interoperable communication systems. In this context, standardization is essential because various machines from different vendors need to cooperate and communicate with each other.

2.3.1 European and International Standard Organizations

Standard-Setting Organizations (SSOs) present one form of a multifirm environment, where companies proactively and voluntarily collaborate to develop standards. SSOs provide the arena for multifirm collaborative interaction by encompassing the entire ecosystem to coordinate the technical interoperability between various system components eventually reducing uncertainty and spurring industry growth [63].

There are several european and international standard organizations which are responsible for maintaining the uniformity of standards in all areas across Europe and the globe. A selection of well-known SSOs is shown in Table 2.2.

European and International Standard Organization	Abbreviation
European Committee for Standardization	CEN
European Committee for Electrotechnical Standardization	CENELEC
European Telecommunications Standards Institute	ETSI
International Electrotechnical Commission	IEC
Institute of Electrical and Electronics Engineers	IEEE
Internet Engineering Task Force	IETF
International Organization for Standardization	ISO
Object Management Group	OMG
Underwriters Laboratories	UL
Amarican National Standards Institute	ANSI
International Telecommunication Union	ITU
World Standards Cooperation (collaboration between IEC, ISO, ITU)	WSC

Table 2.2: European and international standard organizations

2.3.2 Safety Standards

The structure of safety standards is, amongst others, defined in ISO/IEC GUIDE 51 by providing the following types of standards [64]:

- **basic safety standard**, comprising fundamental concepts, principles and requirements with regard to general safety aspects applicable to a wide range of products and systems
- **group safety standard**, comprising safety aspects applicable to several products or systems, or a family of similar products or systems, dealt with by more than one committee, making reference, as far as possible, to basic safety standards
- **product safety standard**, comprising safety aspects for a specific product or system, or a family of products or systems, within the scope of a single committee, making reference, as far as possible, to basic safety standards and group safety standards
- **standards containing safety aspects**,but which do not deal exclusively with safety aspects, making reference as far as possible to basic safety standards and group safety standards

Furthermore, it defines that the risk associated with a particularly hazardous situation depends on the following elements: the severity of harm that can result from the considered hazard and the probability of occurrence of that harm. ISO/IEC GUIDE 51 also includes a procedure that should be used to reduce risks to a tolerable level, and provides definitions and guidance regarding tolerable risk, risk reduction, and validation.

Based on ISO/IEC GUIDE 51, ISO 12100 provides a structure for safety standards in the field of machinery which is divided into type-A, type-B, and type-C standards [53]:

- **type-A standards**(basic safety standards) giving basic concepts, principles for design and general aspects that can be applied to machinery
- **type-B standards** (generic safety standards) dealing with one safety aspect or one type of safeguard that can be used across a wide range of machinery
  - **type-B1 standards** on particular safety aspects (for example, safety distances, surface temperature, noise)
  - **type-B2 standards** on safeguards (for example, two-hand controls, interlocking devices, pressure-sensitive devices, guards)
- **type-C standards** (machine safety standards) dealing with detailed safety requirements for a particular machine or group of machines

When provisions of a type-C standard are different from those which are stated in type-A or type-B standards, the provisions of the type-C satndard take precendence.

type-A: Basic Safety Standards

type-A standards are basic safety standards covering basic concepts, design principles and general aspects that can be applied to all machinery. These standards are general, widely applicable, and have no specific area of application. Examples of type-A standards are shown in Table 2.3.

Standard	Description
ISO/IEC GUIDE 51	*Safety aspects – Guidelines for their inclusion in standards.* Provides requirements and recommendations for the drafters of standards for the inclusion of safety aspects in standards. It is applicable to any safety aspect related to people, property or the environment, or to a combination of these [64].
ISO 12100	*Safety of machinery – General principles for design – Risk assessment and risk reduction.* Provides an overall framework and guidance for decisions during the development of machinery. Specifies the principle strategy for safety of machinery [53].
ANSI B11.0	*Safety of Machinery.* Specifies basic terminology, principles, and a methodology for achieving acceptable risk in machinery design and use. It specifies general requirements and principles of risk assessment and risk reduction [65].
IEC 61508	*Functional safety of electrical/electronic/programmable electronic safety-related systems.* A standard that sets out a generic approach for all safety life cycle activities for systems comprised of electrical and/or electronic and/or programmable electronic components that are used to perform safety functions [10].

Table 2.3: Examples for type-A safety standards

type-B: Generic/Group Safety Standards

type-B standards are generic safety standards covering safety aspects or one type of safeguard that can be used across a wide range of machinery. However, there are two types of B standards, type-B1 standards for particular safety aspects and type-B2 standards for safeguards. Examples of type-B standards are shown in Table 2.4.

Standard	Description
ISO 13849	*Safety of machinery – Safety-related parts of control systems.* Provides safety requirements and guidance on the principles for the design and integration of safety-related parts of control systems, including the design of software. [12].
ISO 14120	*Safety of machinery – Guards – General requirements for the design and construction of fixed and movable guards.* Specifies general requirements for the design, construction, and selection of guards provided to protect persons from mechanical hazards..
ISO 13850	*Safety of machinery – Emergency stop function– Principles for design.* Specifies functional requirements and design principles for the emergency stop function on machinery, independent of the type of energy used [66].

Table 2.4: Examples for type-B safety standards

type-C: Machine/Product Safety Standards

type-C standards provide detailed safety requirements for particular machinery or group of machinery. Their scope is determining the limits of the machinery and the significant hazards covered. Examples of type-C standards are shown in Table 2.5.

Standard	Description
ISO 23125	*Machine tools - Safety - Turning machines* Specifies the requirements and/or measures to eliminate the hazards or reduce the risks in the following groups of turning machines and turning centres, which are designed primarily to shape metal by cutting.
IEC 61496-3	*Safety of machinery - Electro-sensitive protective equipment* Specifies additional requirements for the design, construction and testing of electro-sensitive protective equipment designed specifically to detect persons or parts of persons as part of a safety-related system, employing active opto-electronic protective devices responsive to diffuse reflection for the sensing function.
ISO 16090-1	*Machine tools safety — Machining centres, milling machines, transfer machines* Specifies the technical safety requirements and protective measures for the design, construction and supply of milling machines, machining centres, and transfer machines, which are intended to cut cold metal and other non-combustible cold materials, except wood.

Table 2.5: Examples for type-C safety standards

2.4 Risk Assessment

Risk assessment is the process of quantifying the level of risk associated with the operation of a given machine. It should be a structured and systematic process that answers the following four specific questions [4]:

1. How *severe* are potential injuries?
2. How *frequently* are employees exposed to the potential hazards?
3. What is the *possibility* of avoiding the hazard if it does occur?
4. What is the *likelihood* of an injury should a safety control system fail?

For the risk assessment of plant and machinery, operators and manufacturers require a concept that meets the specifications of national legislation as well as international directives and standards. The harmonized standard ISO 12100 [53] defines important procedures for safety-related systems and safety-related parts of machinery and plant control systems. Figure 2.7 shows a simplified schematic representation of the risk reduction process provided in ISO 12100.

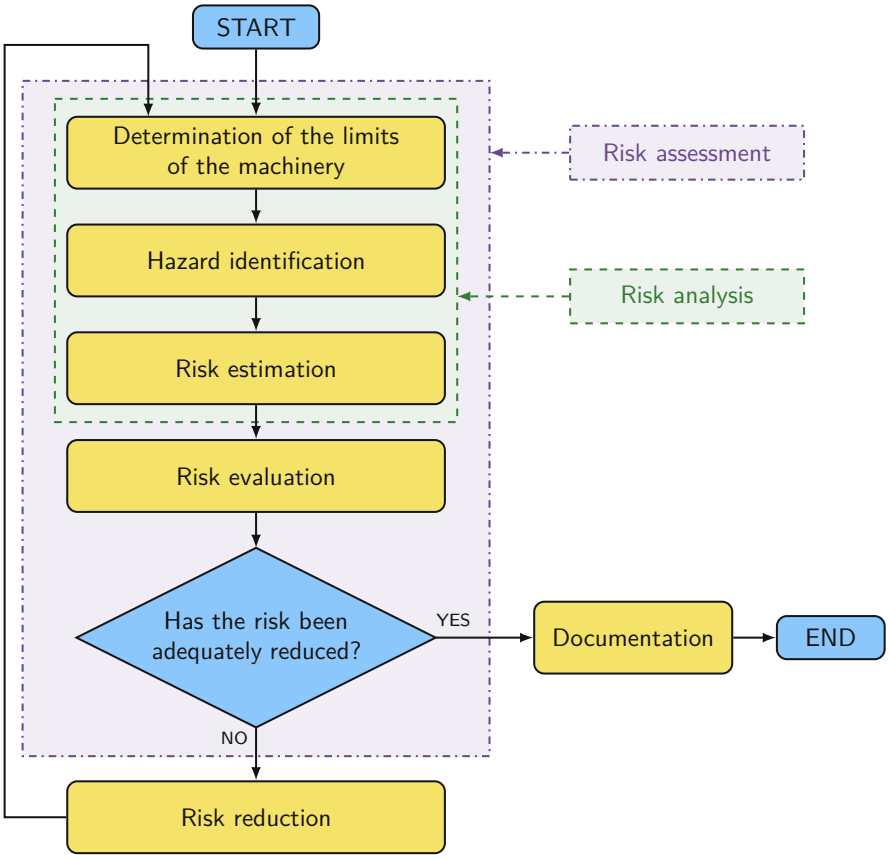


Figure 2.7: Simplified risk reduction process according to ISO 12100

Risk assessment begins with the determination of the limits of the machinery, taking into account all the phases of the machinery life. For this purpose, ISO 12100 differentiates between limits in use, space, time, and other. The next step is a systematic identification of reasonably foreseeable hazards, hazardous situations, and hazardous events during all phases of the life cycle. Techniques for identifying hazards include [67] [68]:

- **Failure Mode and Effects Analysis (FMEA)** is a method designed to identify and understand potential failure modes and their causes, and the effects of failures on systems or processes. It is possible to assess qualitatively the risks, the effects and causes, and then prioritize corrective actions.
- **Hazard and Operability (HAZOP)** is a systematic approach to determining potential problems that may be uncovered by reviewing the safety of designs and revisiting existing processes and operations.
- **Fault Tree Analysis (FTA)** and **Event Tree Analysis (ETA)** are deductive analytic techniques to analyze faults and accidents, determining their causes using tree structures.
- **Cause Consequence Analysis (CCA)** is traditionally used to model the causes of subsystem failures in a critical system and their potential consequences using FTA and ETA dependability modeling techniques, combined in a graphical Cause-Consequence Diagram (CCD).
- **What-If Analysis (WIA)** or **Structured What If Technique (SWIFT)** is a structured brainstorming method. An experienced team reviews each operation or process step utilizing a form where What-If questions are asked and answers generated. The review team then makes judgments regarding the likelihood and severity of the "What-If" answers.
- **Check Lists (CLs)** are mainly applied to processes covered by standards and engineering practices. They are easy and practical to use, as well as suitable to identify ordinary known hazards. However, CLs are highly dependent on the team experience and, depending on their completeness, hazards may not be identified.
- **Layer of Protection Analysis (LOPA)** is a relatively easy, semi-quantitative method for assessing process risks and for defining additional protective measures or measures to reduce the identified risk.

After hazard identification, risk estimation shall be carried out for each hazardous situation by determining the elements of risk. These elements are the severity of harm and the probability of occurrence of that harm as illustrated in Figure 2.8.

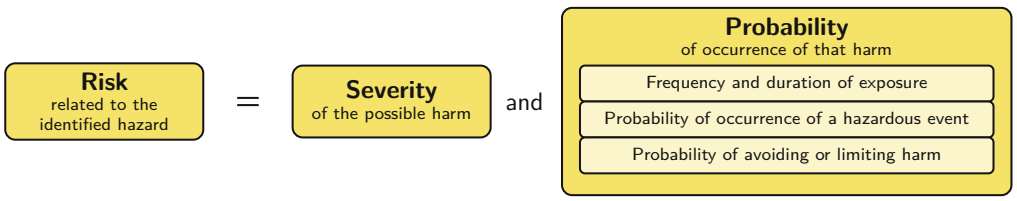


Figure 2.8: Parameters used in risk estimation according to IEC 62061

After risk estimation has been completed, risk evaluation shall be carried out to determine if risk reduction is required. If it is required, then appropriate protective measures shall be selected and applied. When the iterative risk assessment process is in a state where risk has been adequadely reduced, documentation shall demonstrate the procedure that has been followed and the results that have been achieved [53].

2.4.1 Risk Reduction

The objective of risk reduction can be achieved by the elimination of hazards, or by separately or simultaneously reducing the severity of harm, or the probability of occurrence of that harm. All protective measures intended for reduction of risk shall be applied in the following sequence [53]:

1. **Inherently safe design measures** are achieved by avoiding hazards or reducing risks through a suitable choice of design features and interaction between the exposed persons and the machine. Among other things, this includes geometrical factors, physical aspects, appropriate technology, and ergonomic principles.
2. **Safeguarding and complementary protective measures** shall be used to protect persons whenever an inherently safe design measure does not reasonably make it possible either to remove hazards or to sufficiently reduce risks. Based on the risk assessment, appropriate safeguards or other measures can be chosen.
3. **Information for use** consists of communication links, such as texts, words, signs, signals, symbols or diagrams, used separately or in combination to convey information to the user. Within this context, the following should be considered: location and nature of information use, signals and warning devices, markings, signs (pictograms), written warnings, and instruction handbook.

2.4.2 Hazards

ISO 12100 defines hazards as potential source of harm. The term hazard can be qualified in order to define its origin or the nature of the potential harm. In addition, distinction is made between hazards that are permanently present during the intended use of the machine and hazards that can appear unexpectedly.

The risk assessment has to take into account all dangers arising from a machine or a safety component. As a basis, ISO 12100 lists the following hazards:

- mechanical hazards
- electrical hazards
- thermal hazards
- noise hazards
- vibration hazards
- radiation hazards
- material/substance hazards
- ergonomic hazards
- hazards associated with the environment in which the machine is used

This lists of hazards given is not exhaustive. Therefore, any other hazard, hazardous situation, or hazardous event that exists in a machine has to be identified and documented by the designer.

Most of the stated hazards do not change during the lifetime of a machine. However, some hazards may change as a result of substantial modification or the reconfiguration, either hardware or software, of a machine. In that case, a re-assessment of hazards and their risks has to be carried out. Furthermore, it must be verified whether the existing protective measures are still sufficient.

2.4.3 Safety Function

ISO 12100 defines a safety function as a function of a machine whose failure can result in an immediate increase of the risk(s). Whether and to what extent the risk on a machine must be reduced is determined from the risk assessment. The formulation of a safety function generally includes three items of information: (1) triggering event, (2) safety-related reaction, and (3) dangerous part of the machine [69]. The relationships of those items are illustrated in Figure 2.9.

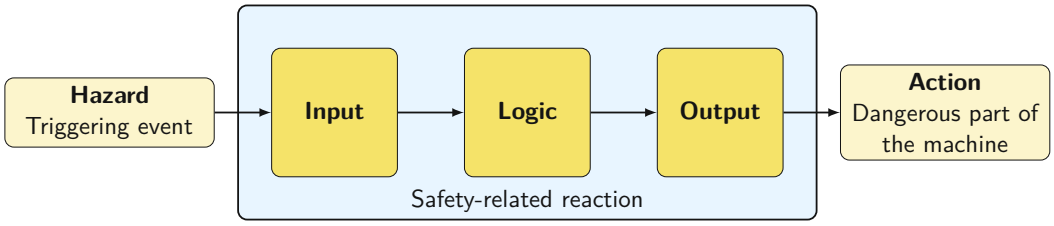


Figure 2.9: Safety function

For each identified hazard a safety function, which provides an adequate risk reduction, must be defined. These safety functions are performed by a control system to ensure risks are kept at an accepted level.

Similar to the limits of the machinery in risk assessment, the limits of the safety functions must also be considered [69] [70]:

- **Use limits** of a safety function include analysis of the machine's operating modes.
- **Space limits** define any geometric shape which may be used as a danger zone, and considers the movements of machine or people and safe distance requirements.
- **Time limits** include mission time and the lifespan of the components, which have an influence upon the probability of failure. Furthermore, the response time of the safeguarding device, the control system, and the mechanical portion of the machine are important parameters for the determination of risk reduction.

2.5 Functional Safety

IEC 61508 defines functional safety as "part of the overall safety that depends on a system or equipment operating correctly in response to its inputs" [10]. The IEC further states in [71] that "Functional safety is the detection of a potentially dangerous condition resulting in the activation of a protective or corrective device or mechanism to prevent hazardous events arising or providing mitigation to reduce the consequence of the hazardous event."

Functional safety is implemented by systems that ensure freedom from unacceptable risk of injury or damage to the health of people. This is achieved by the proper implementation of one or more automatic protection functions or safety functions. Such systems are often based on safety-related electrical, electronic, and programmable electronic (E/E/PE) control systems that are considered to be tolerant to hardware fault. The concept of hardware fault tolerance ensures that a single fault does not cause a loss of the safety function, and it can also be applied to subsystems such as sensors [72].

The most important functional safety standard in continental Europe is IEC 61508, which is a generic standard that can be used as a template for application-specific standards, or it can be applied directly. Furthermore, two international standards for machinery safety, namely ISO 13849-1 [12] and IEC 62061 [11], use the concept of functional safety by specifying safety requirements in terms of functional requirements, by providing guidance, and by defining the amount of risk reduction. Figure 2.10 gives an overview of safety standards with their relationships.

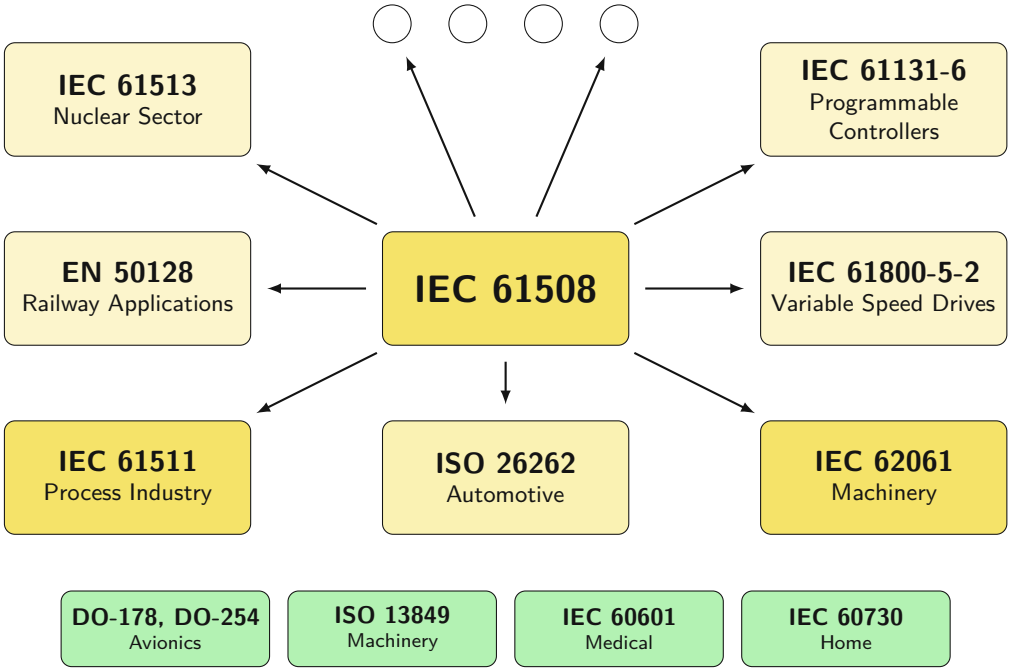


Figure 2.10: Functional safety standards

Based on the fundamental standard ISO 12100 other harmonized standards, such as IEC 61508 [10] with its sector-specific standards IEC 62061 [11] and IEC 61511 [73] and ISO 13849-1/-2 [12], describe the design, construction and integration of Safety-Related Parts of Control Systems (SRP/CSs).

These standards do not cover electrical hazards or other safety requirements necessary at the machine level such as safeguarding. For that purposes, the standards IEC 60204-1 [74], ISO 13850 [66], and ISO 13855 [70] can provide the necessary foundations.

ISO 13849-1 and IEC 62061 provide safety requirements and guidance on the principles for the design and integration of SRP/CS , including the design of software. However, ISO 13849-1 is applicable to hydraulic, pneumatic and electromechanical systems, while IEC 62061 is only applicable to electrical, electronic, and programmable electronic systems. Although, these standards use different methodologies for the design of SRP/CS, their intention is to mitigate the risk to equivalent levels. Both standards require the user to assess the risks by calculating either the average Probability of a dangerous Failure per Hour (PFH_{avg}) or the average Probability of Failure on Demand (PFD_{avg}) due to the different modes of operation (low/high demand mode, continuous mode) defined for a safety function.

IEC 61508 and IEC 62061 assign the PFH_{avg} or PFD_{avg} to Safety Integrity Levels (SILs), ISO 13849-1 uses Performance Levels (PLs) instead.

2.5.1 Safety Integrity Level

A SIL is the pivotal factor in determining the level of safety required for a safety function and also gives a performance measure for the functioning of safety control systems. SILs are derived from IEC 61508 which specifies them as nominal numbers, ranging from 1 to 4, given to specific probability failure ranges [52].

In order to calculate the SIL footherr a specific machine, IEC 61508 provides the structure, while IEC 62061 defines the relevant risk parameters based on the severity of the potential harm (Se) and the probability of the harm occurring. Unlike IEC 61508, IEC 62061 defines SIL only in the range from 1 to 3 [75].

The severity of potential harm is given a score from 1 to 4, with 4 being the most severe. The probability of harm occurring is broken down into three parameters: the frequency and duration of exposure (Fr), the probability of an event occurring (Pr), and the probability of avoiding or limiting the harm (Av). These risk classification parameters are arranged in Table 2.7. Each of these parameters is scored from 1 to 5, with 5 being the worst, or least safe situation, and their scores are summed to determine a class (Cl). The SIL rating is then chosen from a matrix, depicted in Table 2.6, that plots the severity scores (Se) and classes (Cl).

Table 2.6: SIL assigment matrix according to IEC 62061

Severity (Se)	Class (Cl) = Fr + Pr + Av				
	3-4	5-7	8-10	11-13	14-15
4	SIL 2	SIL 2	SIL 2	SIL 3	SIL 3
3			SIL 1	SIL 2	SIL 3
2				SIL 1	SIL 2
1					SIL 1

Table 2.7: Risk classification according to IEC 62061

(a) Severity (Se)

Consequences	Se
Irreversible: death, losing an eye or arm	4
Irreversible: broken limb(s), losing a finger(s)	3
Reversible: requiring attention from a medical practitioner	2
Reversible: requiring first aid	1

(b) Frequency and duration of exposure (Fr)

Frequency of exposure	Fr
$\leq 1h$	5
$> 1h$ *to* ≤ 1 day	5
$> 1day$ *to* $\leq 2weeks$	4
$> 2weeks$ *to* $\leq 1year$	3
$> 1year$	2

(c) Probability (Pr)

Probability of occurrence	Pr
Very high	5
Likely	4
Possible	3
Rarely	2
Negligible	1

(d) Probabilities of avoiding harm (Av)

Probabilities of avoiding harm	Av
Impossible	5
Rarely	3
Probable	1

2.5.2 Performance Level

The PLs specifies the capability of safety circuits to execute a safety function under foreseeable conditions. It is specified as a discrete level from PL "a" to PL "e". With "a" the control function's contribution to risk reduction is low, with PL "e" it is high. ISO 13849-1 uses a qualitative risk graph, to assign a PL for the described safety function, based on three criteria:

- severity of injury
- frequency and/or exposure time to the hazard
- possibility of avoiding the hazard or limiting the harm

The determination of the performance level using severity, frequency, and possibility is shown in the risk graph in Figure 2.11. The values for severity, frequency, and possibility according to ISO 13849-1 are shown in the Tables 2.8a, 2.8b and 2.8c.

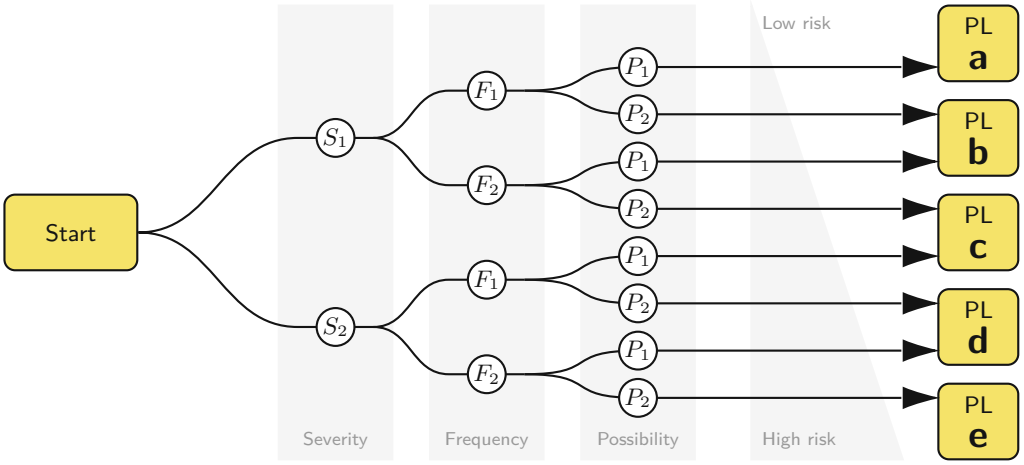


Figure 2.11: PL risk graph according to ISO 13849-1

Table 2.8: Risk classification according to ISO 13849-1

(a) Severity (S)

S	Severity of injury	Possible assessments
S_1	Slight (generally reversible injury)	Injuries that require nothing more than first aid or do not lead to more than two days of absence from work
S_2	Serious (usually irreversible injury or death)	Injuries that require treatment by a doctor or lead to more than two days of absence from work

(b) Frequency (F)

F	Frequency and/or duration of the exposure to hazard	Possible assessments
F_1	Seldom to less often and/or the time of exposure to hazard is short	No more than twice per shift (8 working hours) and shorter than 15 minutes in total per shift
F_2	Frequent to continuous and/or the duration of the of exposure to hazard is long	More than twice per shift (8 working hours) or longer than 15 minutes in total per shift

(c) Possibility (P)

P	Possibility of avoiding the hazard or limiting the damage	Possible assessments
P_1	Possible under certain conditions	In certain cases, the hazard can be reduced
P_2	Scarcely possible	Hazard cannot be avoided

2.5.3 Safety Life Cycle

The safety life cycle is provided by the various specifications in order to give designers a framework for creating safe systems. It addresses the analysis, design, installation, operation, maintenance, and disposal of equipment. IEC 61508 divides the life cycle, illustrated in Figure 2.12, into three main parts: analysis, realization, and operation.

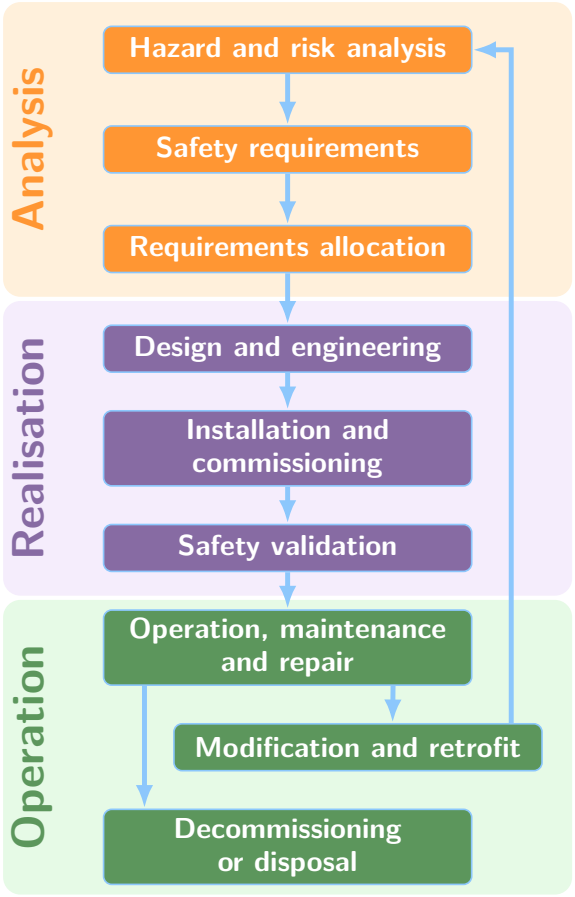


Figure 2.12: Simplified safety life cycle accoring to IEC 61508

The safety requirements are identified in the analysis phase based on a hazards and risk analysis. Furthermore, safety functions are specified to allocate the appropriate safety integrity levels for each safety system. In the realization phase, technology and architecture are selected which meet the safety requirements identified in the analysis phase. This phase ends with a safety validation after design, engineering, and installation. During operation, the final phase, the systems are maintained and repaired as specified in the requirements. Modifying machinery in this phase can result in new assessments, leading back to the analysis phase. Finally, and not to be overlooked, decommissioning or disposal of a system can also occur in this phase.

The System Development Life Cycle (SDLC) is a systematic approach that generates a structure for the developer to design, create and deliver software based on customer requirements. It is a process for planning, creating, testing, and deploying a software-based system. Several life cycle models are available such as waterfall model, agile model, V-shaped model, or spiral model.

In the field of functional safety, IEC 61508 gives detailed guidance on many aspects of the development of safety-related software and suggest a range of techniques. It uses a generic V-model to describe the development of software which is shown in Figure 2.13. ISO 13849-1 applies a more simplified V-model.

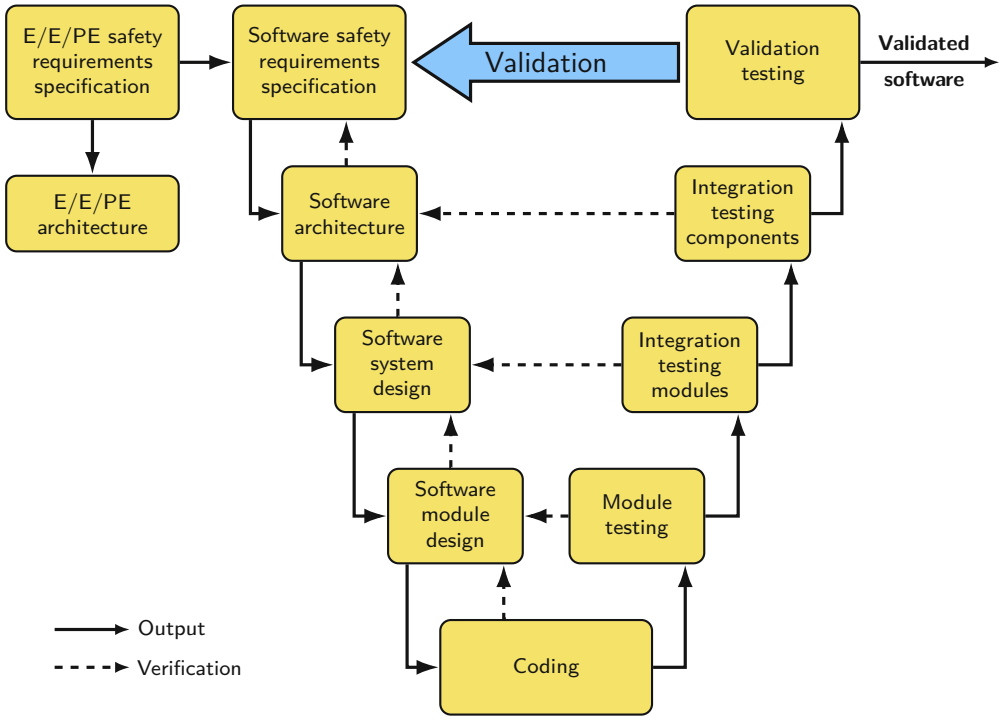


Figure 2.13: Software safety life cycle according to IEC 61508 and ISO 13849-1

Safety-related software is developed using standard software life cycle models, such as the V-model. However, it is usually necessary to obey additional requirements on the methods and functions applied in order to meet the technical requirements of IEC 61508 or other safety-related standards. For that reason, it is important to separate safety-related and non-safety-related functions within a system clearly. Otherwise, the whole software development must be compliant with IEC 61508. Furthermore, it is advisable to clearly separate the safety functions of different SIL or PL levels. Otherwise, all functions must be developed compliant to the highest SIL or PL level in the whole system [76].

2.6 Safety-related Communication

A crucial part of a safety function (see also Figure 2.9) is the communication between input, logic, and output. Besides detecting a hazard, processing the logic, and activating a protective measure, a safety-related reaction to a triggering event for a hazard depends essentially on communication. Therefore, particular attention must be paid to safety-related information transfer to ensure a machine's safe and reliable operation.

The classic approach for building a functional safety system is to connect safety devices using cables with line monitoring. This discrete wiring concept involves efforts in wiring each sensor and actuator to the safety logic (e.g., relays or a safety Programmable Logic Controller (PLC)). It is an inflexible solution as every change in the safety configuration may imply a change in cabling, which leads to a long downtime of the machine or production line. This reliable and proven approach is still used very often today but is a major obstacle to rapid reconfiguration.

2.6.1 Requirements for Data Communications

In recent years, automation networking technologies, including wireless technologies, have been enriched with safety features, particularly those dedicated to the industrial automation domain. As the platform for all communications within a production line, an industrial network must support the two paradigms of deterministic real-time traffic and best-effort traffic, which allows the transmission of process or safety data and configuration or parametrization data on the same network. For this to achieve, one pre-requisite of a safe and stable operation of a functional safety protocol based on an industrial network is the deterministic transmission of data, which includes a low latency, minimal jitter, and minimal packet loss. Compared to the discrete wiring solution, this approach can be reconfigured without rewiring cables. However, additional techniques and measures have to be applied to protect safety-relevant data [77] [78] [79].

IEC 61508 – Part 2 formulates requirements for data communications used in the implementation of a safety function. The failure measure, such as the residual error rate, of the communication process shall be estimated. In doing so, potential threats such as transmission errors, repetitions, deletion, insertion, resequencing, corruption, delay, and masquerade must be taking into account. Therefore, IEC 61508 offers two possible architectures for data communication: "White Channel" and "Black Channel".

IEC 61784-3 defines common principles for transmitting safety-relevant messages among participants within a distributed system using control network technology in accordance with the requirements of the IEC 61508 series for functional safety. Furthermore, IEC 61784-3 recommends the use of the "Black Channel Principle" which is based on the requirement that the transmission of safety-related data is performed independently of the characteristics of the transmission system. Therefore, an additional layer – the safety layer – is placed on top of the application layer. This safety layer considers that safety data is subject to various threats, and for each one, a set of defence measures is defined in order to protect this data [80, chapter 46.1.2].

White Channel

With this paradigm, the communication protocol, including the hardware required for transmission, has to be included in the safety assessment. Furthermore, the communication channel must be designed, implemented and validated according to the IEC 61508 series and IEC 61784-3 or IEC 62280 series. Figure 2.14 illustrates this approach which leads to increased effort in development and is no longer state of the art.

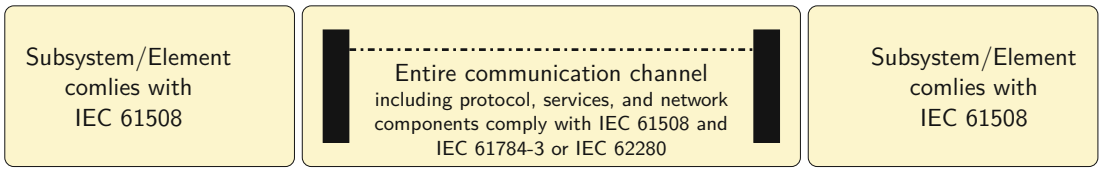


Figure 2.14: White channel according to IEC 61508-2

Black Channel

This approach uses a superimposed safety layer where the safety-critical participants communicate on a higher level, the so-called black channel. The primary purpose of this additional safety layer is to transmit safety-related data between nodes. Important properties from the underlying transport layer, such as determinism and channel availability, are taken over. Figure 2.14 depicts this approach where parts of the communication channel are not designed or validated according to the IEC 61508 series. In this case, the measures to ensure the failure performance of the communication process are implemented in the subsystems, which are in accordance with the IEC 61784-3 or IEC 62280 series.

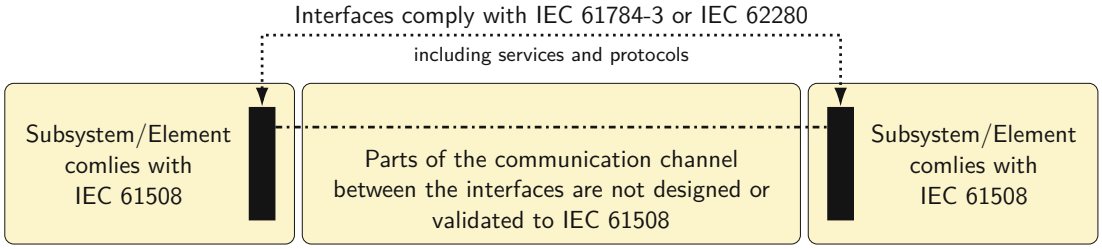


Figure 2.15: Black channel according to IEC 61508-2

Safety-related data transmission using a black channel does not necessarily require deterministic or real-time behaviour of the underlying transport layer. However, regarding reliability, it is paramount for a safety system to use a data transport channel with guaranteed capabilities such as low latency, minimal jitter, and minimal packet loss. The lack of those guarantees would result in an unreliable operation due to the unpredictable transmission of safety-critical data leading to a violation in the safety layer and consequently bringing the machine to a safe state [81] [82] [83].

2.6.2 Wireless Communication for Safety Applications

Recently, the idea of replacing wired industrial communication platforms with wireless technology has gained interest. Some advantages are flexibility, scalability, lower operational costs, and ease of integration. Although there is a research activity in the field of wireless safety (e.g., wirelessHART, IO-Link Safety), there is not yet an industry-accepted wireless system that guarantees safety function requirements [84].

However, a technology which aims at providing wireless and wired connectivity is Time-Sensitive Networking (TSN). Its goal is to enable the coexistence of time-critical and best-effort traffic over Ethernet, WIFI, and 5G [85], [86].

2.6.3 Performance of Safety-critical Systems

An additional index to assess the performance of safety-critical automation systems is the Safety Function Response Time (SFRT) which has been introduced for a specific functional safety protocol but shows, in general, the time-critical behaviour of a safety function. In particular, it is defined as "worst case elapsed time following an actuation of a safety sensor connected to a fieldbus, before the corresponding safe state of its safety actuator(s) is achieved in the presence of errors or failures in the safety function channel" [87]. It represents the time that elapses between detecting a hazard and the system's transition into a safe state. Figure 2.16 illustrates the context of Worst Case Delay Times (WCDTs) and Watchdog Times (WDTimes) relating to a safety function. The Total Worst Case Delay Time (TWCDT) results from the WCDTs of all entities of the safety function. For safety reasons every entity has its superposed watchdog timer, which takes the necessary actions to activate the safe state whenever a failure or error occurs within that particular entity [88].

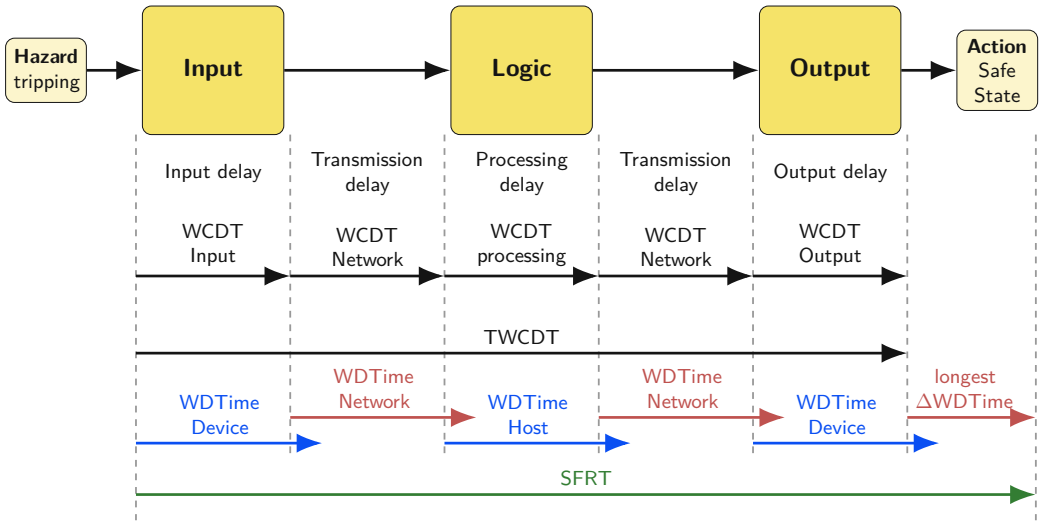


Figure 2.16: Safety function context of delay times and watchdog times

Based on WCDT and WDTimes of all components of a safety function, the SFRT can be calculated as shown in Equation 2.1.

$$SFRT = \sum_{i=1}^{n} WCDT_i + \max_{i=1,2,...n} (WDTime_i - WCDT_i) \tag{2.1}$$

One of the most important metrics for safety-critical applications is the time between a detected error and the transition to a safe state. Therefore, the SFRT specifies the worst-case time before a safe state is achieved in the presence of errors or failures within a safety function. Depending on the application, the requirements of the SFRT range from milliseconds to seconds [77].

2.6.4 Safety Protocols

Although all safety protocols defined in IEC 61784-3 are transport layer agnostic using the "Black Channel Principle", almost every Industrial Ethernet protocol comes with its own safety protocol. Some of the most prominent protocols are listed in Table 2.9 with specific reference to the Communication Profile Families (CPFs) defined in IEC 61784-1 [89].

CPF	Commercial Name	Safety Protocol
1	FOUNDATION Fieldbus	FF-SIS
2	Common Industrial Protocol (CIP)	CIP Safety
3	PROFIBUS & PROFINET	PROFIsafe
6	INTERBUS	INTERBUS Safety
8	CC-Link	CC-Link Safety
12	EtherCAT	Fail Safe over EtherCAT (FSoE)
13	Ethernet POWERLINK	Ethernet POWERLINK Safety
14	EPA	EPASafety
16	SERCOS	CIP Safety
17	RAPIEnet	RAPIEnet Safety
18	SafetyNET p	SafetyNET p

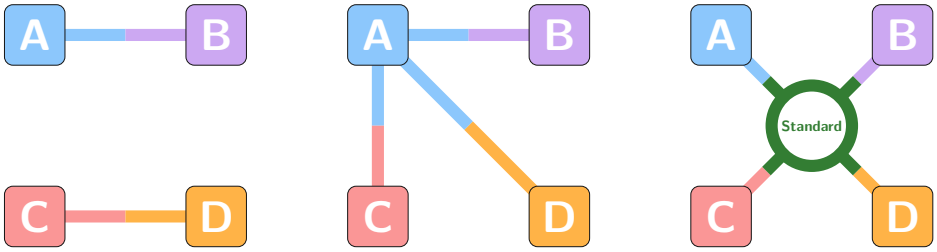
Table 2.9: Safety protocols defined in IEC 61784-3

An additional safety protocol which has been recently defined in agreement with the guidelines of IEC 61784-3, but not yet included in the standard, is OPC UA Safety [90]. It is a manufacturer-independent standard for safe communication between controllers. It describes services and protocols for the exchange of data using OPC UA mechanisms. The choice of an Industrial Ethernet solution almost always determines the safety protocol which has to be used within a machine. But the advantage of a well integrated safety protocol entails difficulties when it comes to interoperability of machines of various manufacturers within a production line [46].

2.7 Interoperability

Interoperability is defined as the ability of two (or more) systems or components to exchange information and use the information that has been exchanged. There are four levels of interoperability. These are technical, syntactic, semantic, and organizational. Technical interoperability is typically associated with components that enable machine-to-machine communication. It focuses on the physical aspects of interconnections and their communication protocols. Syntactic interoperability is defined as the ability to exchange data. It considers the structure and format of data. Semantic interoperability involves M2M communication and humans. It is related to the definition of content and explains how data can be interpreted. Organizational interoperability refers to the capability of organizations to exchange data effectively. It relies on the successful interoperability of the technical, syntactic, and semantic aspects [91].

In the manufacturing domain, interoperability represents a characteristic of a manufacturing system in which its components are capable of exchanging information with one another, using the information that has been exchanged [92]. Information exchange can be based on three different paradigms: compatibility, de facto standard, and interoperability. Figure 2.17 illustrates these paradigms. 1.2.1.9



(a) Compatibility (b) De facto standard (c) Interoperability

Figure 2.17: Information exchange

In the industrial automation (Industry 3.0) era, machines were built as closed systems. There were only minimal interfaces to access data or influence the operation of the machine. For that purpose, compatibility between two machines or devices was satisfactory. This approach is no longer sufficient for machines as part of production lines in smart factories. Fundamental prerequisites for smart manufacturing are production facilities that are able to adapt to a changed production environment or production flow and to exchange data using unified M2M communication. Therefore, these facilities depend on technical, syntactic, semantic, and organizational interoperability.

Production facilities in the context of smart manufacturing become Cyber-Physical Production Systems (CPPSs). A CPPS is a network of interconnected Cyber-Physical Systems (CPSs) specialized for manufacturing. The challenge of a new vision of CPPSs is to change the architecture from a hierarchy to a modular and interconnected system implemented either as services or agents. With respect to technical interoperability, there are some standards available (e.g., OPC Unified Architecture (OPC UA), IEC 61499,

TSN). Semantic interoperability is based on models that need to cover three main elements: products and services; production and process; production resources. Using such models makes it possible to provide a consistent view on the product along the different stages of its development and production process across the supply chain. AutomationML and Systems Modeling Language (SysML) are two semantic modelling approaches that allow to connect different views and parts of manufacturing. Furthermore, OPC UA not only aims to provide technical interoperability but also interoperability of the semantic layer shall be addressed based on so-called Companion Specifications. Organizational interoperability in smart manufacturing requires horizontal and vertical connectivity in order to connect machines and business processes. An initiative for interoperable connectivity across the supply chain is GAIA-X[5] [93].

2.8 Operating Mode Selection

In principle, a machine's functional safety system configuration has to be certified and, therefore, cannot be altered after the machine has been commissioned. However, in some cases, machinery may be designed with specific control modes, for example, for setup or maintenance operations.

For that purpose, the European Machinery Directive 2006/42/EC explicitly states in "Annex I – Section 1.2.5" the possibility of a "selection of control or operating modes" [94]. It, therefore, describes the conditions under which a mode selector is permitted to change the safety-relevant functions of a machine.

In general, it is required that the different control or operating modes be exclusive of each other, except for the emergency stop function, which must be available whichever control or operating mode is selected. Certain modes may enable specific functions of the machinery to be controlled with guards open or with protective devices muted. In these cases, each position of the mode selector must correspond to a single control or operating mode. Additionally, it must be possible to lock the mode selector device in each position. Furthermore, indicators must clearly show which control or operating mode has been selected. As an alternative to a physically lockable selector, the selection of a control or operating mode may be restricted to specially trained and authorised operators by other means, such as an access code or RFID-based selector switches with coded keys.

The application of a mode selector enables a minimal degree of flexibility in an otherwise very rigid system. It allows the operation of a machine in predefined and certified modes. That also means that the risk of each operating mode and switchover must be determined and analyzed within the risk assessment separately before commissioning a machine.

[5] https://www.data-infrastructure.eu/GAIAX/Navigation/EN/Home/home.html

2.9 On-line and Off-line Software Tools

Generating safety-related software or configurations places additional requirements on the process and toolchain. It needs to be ensured that the tools used comply with predefined criteria. The first standards that introduced the concept of tools qualification were DO-178B and ED-12B (Software Considerations in Airborne Systems and Equipment Certification) in the aeronautical field. Developed jointly by the safety-critical working group RTCA SC-167 of the Radio Technical Commission for Aeronautics (RTCA) and WG-12 of the European Organisation for Civil Aviation Equipment (EUROCAE), they were published in 1992 as DO-178B by the RTCA and as ED-12B by the EUROCAE.

The safety standard IEC 61508 has been applied in discrete manufacturing and process industry for many years. Since its 2008 version, it has required the qualification of software tools. IEC 61508 – Part 3 and 4 define a software on-line support tool as one that can directly influence the safety-related system during its run time. It has to be considered to be a software element of the safety-related system. In contrast, a software off-line support tool is defined as one that supports a phase of the software development life cycle and cannot directly influence the safety-related system during its run time. It has to be selected as a coherent part of the software development activities, and the selection has to be justified.

Software off-line tools may be divided into the following classes:

T1 generates no outputs which can directly or indirectly contribute to the executable code (including data) of the safety related system

T2 supports the test or verification of the design or executable code, where errors in the tool can fail to reveal defects but cannot directly create errors in the executable software

T3 generates outputs which can directly or indirectly contribute to the executable code of the safety related system

Off-line software tools include compilers, editors, static analysis tools, and requirements management tools. Off-line software tools classified as T2 or T3 need a specification or product documentation clearly defining the behaviour. Moreover, an assessment to determine the level of reliance placed on the tools and the potential failure mechanisms of the tools that may affect the executable software has to be carried out.

The activity of demonstrating that a tool can be used as part of the realization of a software application with a determined safety goal is called tool qualification. Its purpose is to associate the risks induced by tools on the final product [95].

CHAPTER 3

Methodology

> SCIENTISTS ARE NOT IN THE HABIT OF THINKING ABOUT MATTERS OF METHODOLOGICAL POLICY. ASK A SCIENTIST WHAT HE CONCEIVES THE SCIENTIFIC METHOD TO BE, AND HE WILL ADOPT AN EXPRESSION THAT IS AT ONCE SOLEMN AND SHIFTY-EYED: SOLEMN, BECAUSE HE FEELS HE OUGHT TO DECLARE AN OPINION; SHIFTY-EYED, BECAUSE HE IS WONDERING HOW TO CONCEAL THE FACT THAT HE HAS NO OPINION TO DECLARE. [96]
>
> – Peter B. Medawar, 1984

This research constitutes an exploratory study on functional safety and safety engineering challenges in smart manufacturing. A methodological concept is developed that addresses the challenges of Industry 4.0 machinery safety requirements and assists the safety engineer in applying configuration changes during operation. Therefore, the objectives of flexibility and reconfiguration in a typically static environment are addressed, which helps to reduce engineering efforts.

In principle, the overall research methodology follows the System Development Life Cycle (SDLC). Additionally, the SDLC is enhanced and refined into the Flexible Safety System Development Life Cycle (FSS-DLC) explained in detail in this chapter.

The applied research methodology follows, on a higher level, an adapted systems development life cycle defining the phases of the research. On a lower level, an individual methodology is utilized in each phase of the development life cycle. The overall development life cycle and the subjacent methodologies for the individual phases are briefly discussed in the following. The methodological approaches applied in the individual phases are explained in detail in the corresponding chapters.

3.1 Systems Development Life Cycle

The SDLC is a step-by-step engineering process for analyzing, planning, designing, creating, testing, deploying, and maintaining an information system. It is the oldest formalized methodology for building information systems. It pursues the development of information systems in a deliberate, structured, and methodical way, requiring each life cycle phase to be carried out rigidly and sequentially. The traditional SDLC originated in the 1960s with the purpose of developing large-scale functional business systems and is divided into four evolutionary phases: systems analysis, systems design, systems implementation, and systems evaluation. Over time, variations of the traditional SDLC have been adopted for the development of hardware, software, or other complex systems [97].

In today's literature, additional phases are added to the traditional four phases, resulting in a life cycle consisting of analysis, design, development, testing, deployment, maintenance, evaluation, and disposal, as illustrated in Figure 3.1.

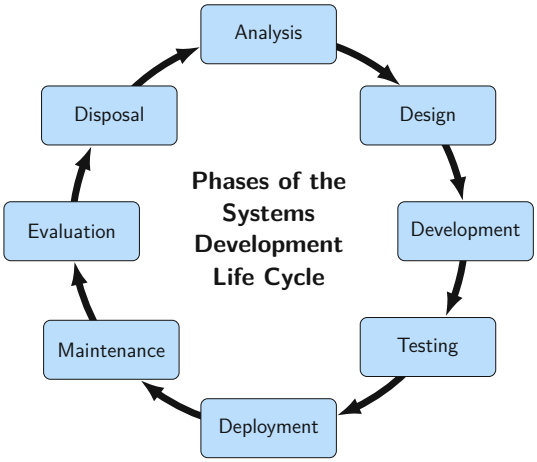


Figure 3.1: Phases of the Systems Development Life Cycle

In the **analysis phase**, all relevant information from stakeholders is gathered and analyzed to determine the goals and the required specifications to fulfil those goals. The **design phase** covers components such as the system architecture, interfaces, and toolchain and will end with a design specification. Based on this specification, a prototype or sample framework can be created. The system is built in the **development phase** based on an approved design. The integrated and completed components are tested in the **testing phase**. In the **deployment phase**, the system is put into operation in a production environment. During the **maintenance phase**, continuous improvement is performed regularly, including hardware and software updates. In the **evaluation phase**, a post-implementation review identifies whether the system meets the initial requirements and objectives. The last phase is the **disposal phase**, where data and information are retrieved for preservation and the physical disposition of an asset.

3.2 Adapted Systems Development Life Cycle

The methodology used for this research is based on the previously described SDLC following the life cycle from the analysis to an architecture evaluation as a precursor to the development phase. However, the SDLC does not provide all needed phases and, therefore, is enhanced with the three phases: technology integration, software architecture, and architecture evaluation. In the following, this adapted life cycle is referred to as the FSS-DLC. Figure 3.2 gives an overview of the phases of the FSS-DLC.

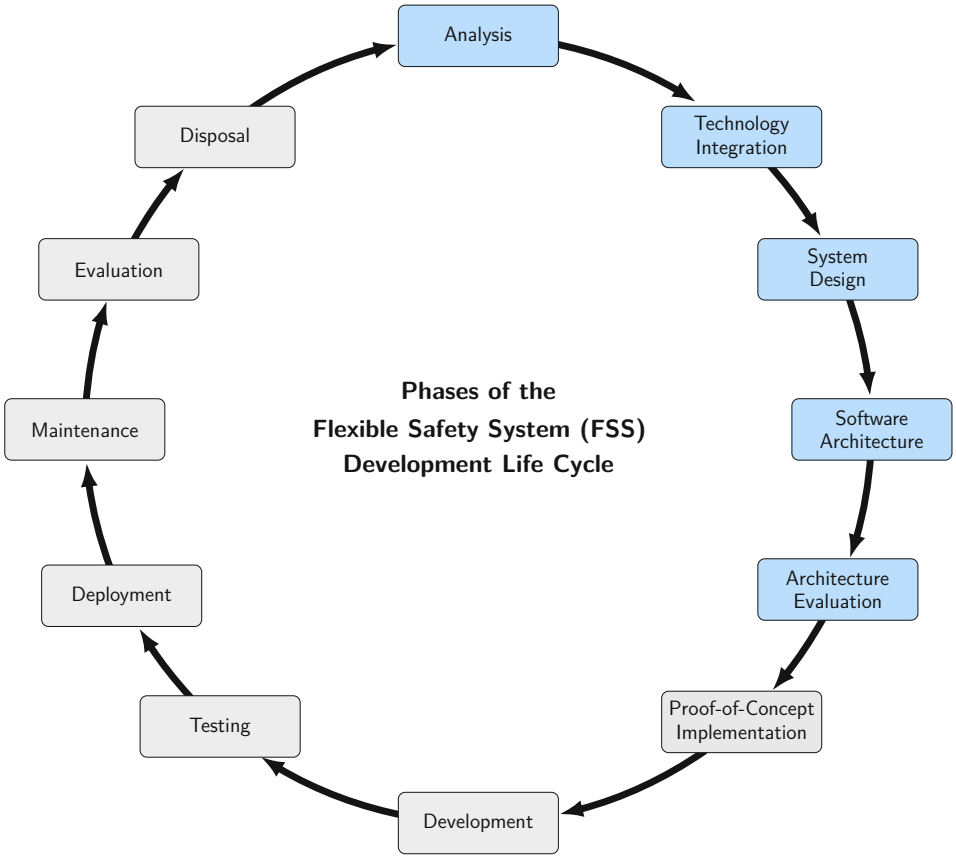


Figure 3.2: Flexible Safety System Development Life Cycle (FSS-DLC)

This thesis follows the path analysis, technology integration, system design, software architecture, and architecture evaluation. The phases from proof-of-concept implementation until disposal are outside the scope of this thesis.

The first phase in the FSS-DLC is an analysis in the field of functional safety in machinery used in the domain of discrete manufacturing that is carried out with the involvement of various stakeholders and comprises an abstract system model, a use case analysis, and a requirements determination. The analysis results lay the foundation for the intended objective of a design for a flexible safety system. Technological issues are resolved before

addressing the system's design by choosing suitable base technologies. After the system design, a software architecture is created, which will be evaluated in the next phase, the architecture evaluation.

The results of this research will serve as a basis for a proof-of-concept implementation, which should demonstrate the basic function of the concept. The theoretical part of this research takes place at the Automation Systems Group at TU Wien, whereas the Center for Digital Production at Pilotfabrik of TU Wien serves as a playground for practical activities and proof-of-concept implementation. The conceptual foundation for the system design will be based on the answers to the three research questions.

The following course of the thesis is aligned to the phases of the FSS-DLC. These phases, with their inputs and outputs, have various dependencies that shape the course of the research and are illustrated in Figure 3.3.

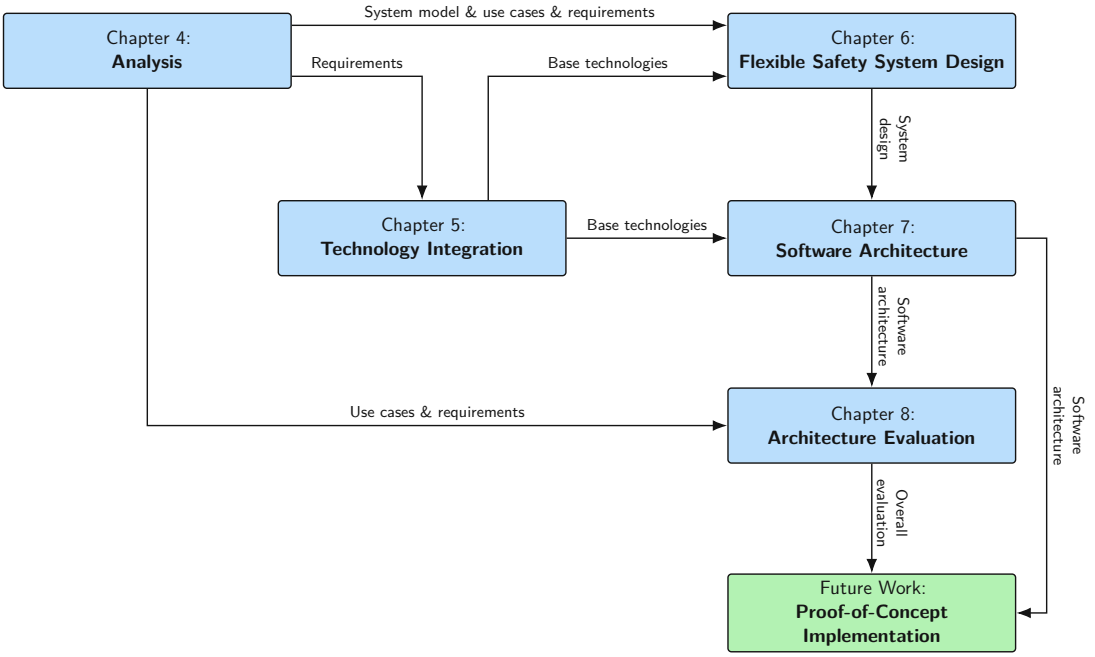


Figure 3.3: Dependencies between individual phases of the FSS-DLC

As depicted in Figure 3.3, and taking into account the sequence of phases of the FSS-DLC shown in Figure 3.2, several artifacts are created and shared between phases. These artifacts are the system model, use cases, and requirements in the **analysis phase**. The output of the **technology integration phase** is a set of base technologies that fulfill the requirements of the previous phase. Subsequently, during the **system design phase**, the system model, uses cases, requirements, and base technologies are transformed into a system design for a Flexible Safety System (FSS). The software architecture is derived from the system design and the base technologies in the **software architecture phase**.

Finally, the architecture is evaluated in the **architecture evaluation phase** using the use cases, requirements, and software architecture as inputs, creating an overall evaluation. The software architecture and the overall evaluation results serve as inputs for the subsequent **proof-of-concept implementation phase**, which ensures the architecture's correct function and is outside this thesis's scope.

3.3 Methodolodical Approaches in the Individual Phases

In the individual phases of the FSS-DLC, various methodological approaches are applied, which are briefly described in the following and, when needed, explained in detail in the corresponding chapters. The phases are divided into tasks that build on one another using the output of their respective predecessors. Additionally, the outputs of the tasks are available for the subsequent phases.

3.3.1 Analysis Phase

In the first phase of the FSS-DLC, an analysis is carried out consisting of three tasks: abstract system model creation, use case analysis, and requirements determination. These tasks are performed sequentially and are depicted in Figure 3.4.

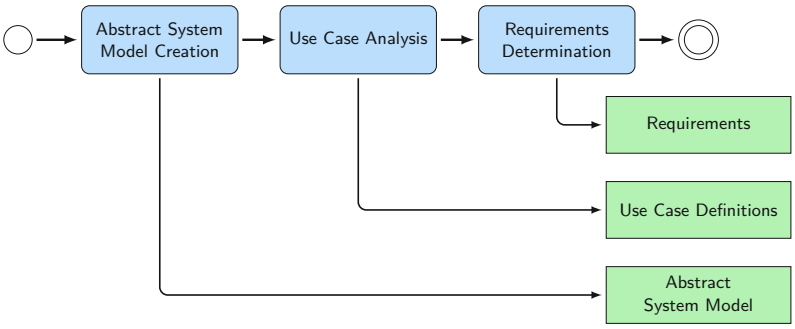


Figure 3.4: Methodological approach of the analysis phase

The abstract system model provides a vendor-neutral and technology-agnostic model of a generic production system used for discrete manufacturing. It includes the safety system's specific components, characteristics, and configurations. In order to understand the purpose of a production system, use cases are defined to explain and document the interaction between the user and the system to accomplish a desired task. After studying the existing system and gathering details of its use, requirements are derived that serve as a basis for creating models, designs, and architectures.

3.3.2 Technology Integration Phase

Existing technologies that fulfill the requirements from the previous analysis phase are chosen to serve as a safe communication platform during the technology integration

phase, which comprises three tasks. First, a base technologies review on communication technologies that can serve as the foundation for safety-critical, standardized, and vendor-neutral communication. Second, the compilation of a safe communication platform that combines the technologies from the previous task. Third, an investigation regarding safety-related configuration tools considering the specific aspects of the safe communication platform compiled in the previous task. These tasks are performed sequentially and are depicted in Figure 3.5.

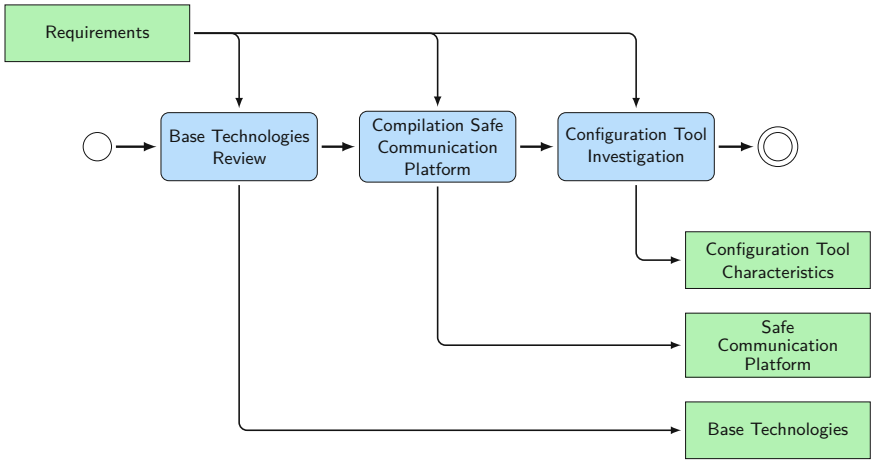


Figure 3.5: Methodological approach of the technology integration phase

The created artifacts of this phase are a description of suitable base technologies, a specification for combining the base technologies into a safe communication platform, and the characterization of a safety-related configuration tool.

3.3.3 System Design Phase

Based on the created artifacts from the analysis and technology integration phases, a system design is created in this phase, which is split up into four tasks. First, an overall concept creation, including the system components, communication relations, a communication stack, and the system configuration. Second, developing a safety communication establishment specification involving safety device communication and deterministic data transport. Third, creating a safety network management design consisting of a configuration procedure and the required system services. Fourth, the conception of control interfaces comprising a Human-Control Interface (HCI) and a Machine-to-Machine-Control Interface (M2MCI). These tasks are performed sequentially and are depicted in Figure 3.6.

The created artifacts of this phase are an overall system concept, a safety communication scheme, a safety network management design, and a control interface definition.

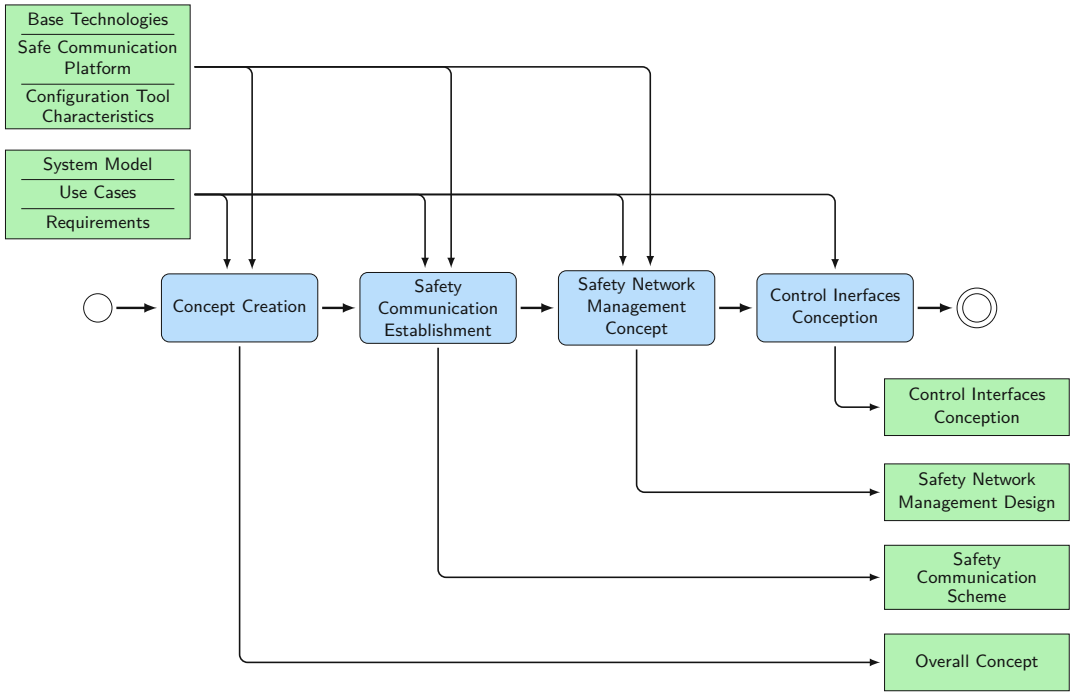


Figure 3.6: Methodological approach of the system design phase

Additionally to this methodological approach, the system design follows the design principles stated by Tim Berners-Lee in 1998 as a personal note[1]. These principles are simplicity, modular design, being part of a modular design, tolerance, decentralization, test of independent invention, and the principle of least power. The design of a flexible safety system takes into account all these principles. **Simplicity** refers to the user experience since the objective of this system is to assist the engineer in making a very complex task simple. In order to create a flexible solution, the design has to be **modular**, and it has to consider that it will be **part of a bigger system**. The principle of **tolerance** can be applied in that way that the system accepts not precisely defined commands, but it creates configurations that strictly conform to the defined specifications. The proposed safety system will be designed as a distributed system and therefore addresses the principle of **decentralization**. The principle of **test of independent invention** asks, "If someone else had already invented your system, would theirs work with yours?". Therefore, emphasis must be placed on interoperability in all parts of the system and the use of existing technology. Finally, the principle of **least power** suggests choosing the least powerful language suitable for a given purpose. In this research, the definition will also be expanded to include communication protocols.

[1] Principles of Design https://www.w3.org/DesignIssues/Principles.html

3.3.4 Software Architecture

The purpose of this phase is the creation of a software architecture for a FSS utilizing the C4 model for visualizing software architecture[2]. In doing so, several artifacts are needed from two previous phases. First, the description of the base technologies, the specification for a safe communication platform, and the characterization of a safety-related configuration tool from the technology integration phase. Second, the overall system concept, the safety communication scheme, the safety network management design, and the control interface definition from the system design phase. The phase is divided into four tasks: context creation, container creation, components creation, and code creation, which are depicted in Figure 3.7.

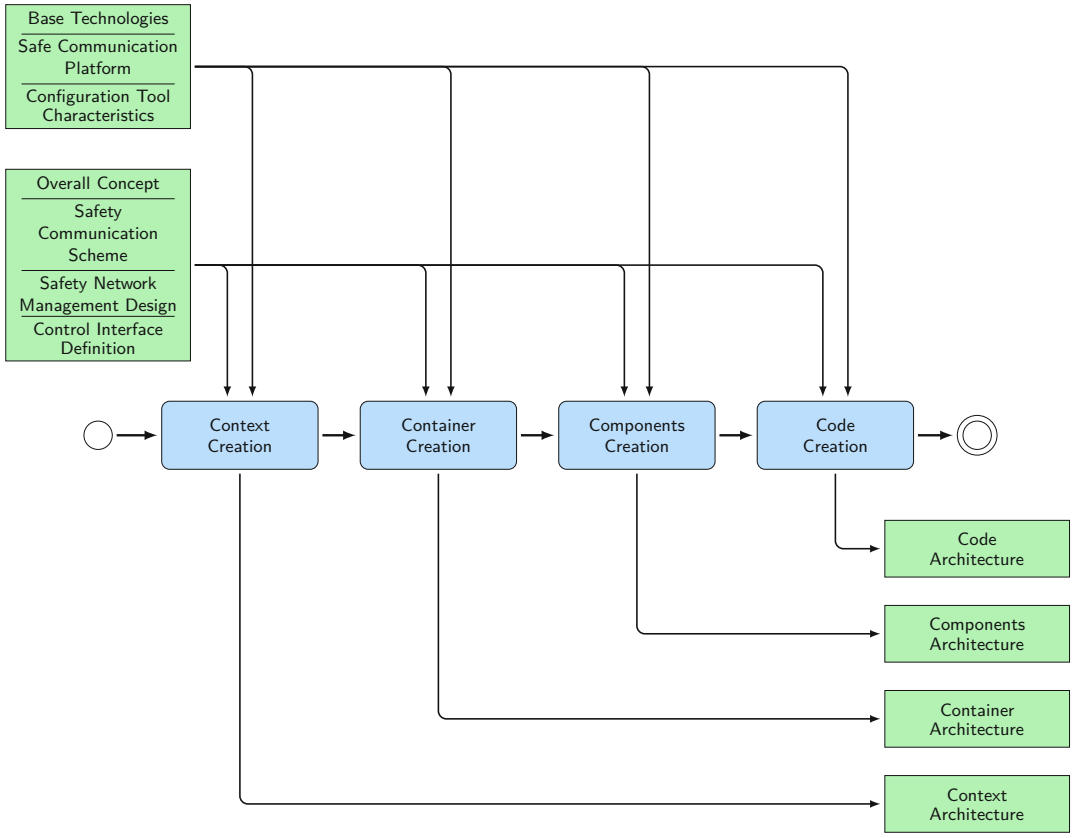


Figure 3.7: Methodological approach of the software architecture phase

The created architecture artifacts of this phase are diagrams for context, container, and components according to the C4 methodology. Furthermore, code artifacts are produced in related scientific work.

[2] https://c4model.com

3.3.5 Architecture Evaluation

Based on the use cases and requirements from the analysis phase and the architecture artifacts from the software architecture phase, a suitable evaluation method is chosen in the first task of the architecture evaluation phase. The second task applies the Software Architecture Analysis Method (SAAM) as outcome of the previous task on the software architecture from the preceding phase. The tasks in this phase are performed sequentially and are depicted in Figure 3.8.

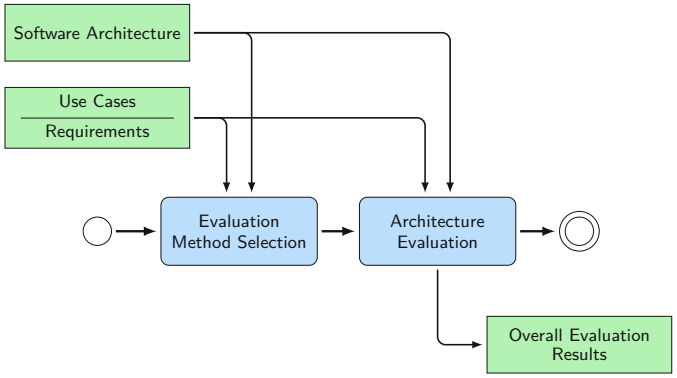


Figure 3.8: Methodological approach of the architecture evaluation phase

The outcomes of this phase are the overall evaluation results, which are used besides the software architecture artifacts in the proof-of-concept implementation phase.

CHAPTER 4

Analysis

> We all know what it means in ordinary life to be safe. I am safe in my room, when I cannot be run over by an omnibus. I am safe if I have had whooping cough and cannot therefore get it again. To be safe essentially means that it is physically impossible that certain things should happen to me and therefore it's nonsense to say that I am safe whatever happens. ... this is a misuse of the word "safe" ... [98]
>
> – Ludwig Wittgenstein, 1929

In this chapter, all relevant information from stakeholders, such as machine manufacturers, machine operators, operating and service personnel, and safety engineers, is gathered and analyzed to determine the goals and the required specifications to fulfil those goals. Those goals are broken down into defined functions and operations of the desired application. In doing so, the topic is approached from different angles and perspectives.

The analysis starts with an abstract system model of a generic production system with an embedded safety system. The model includes and examines properties of safety functions, safety devices, safety communication, and safety configuration. In the next step, a use case analysis reveals the machinery operators' needs in smart manufacturing within several production scenarios. Requirements for a flexible safety system will be derived from the outcomes of the use case analysis, combined with the requirements on functional safety systems. In doing so, the three requirement categories operational, functional, and non-functional are considered. The system model and the requirements will be the foundation for the subsequent technology integration and the system design.

4.1 Abstract System Model

Before starting the design for a flexible safety system, the environment of a production system with its components and their relationships have to be analyzed. Figure 4.1 shows the structure of a generic production system with a safety system embedded into it.

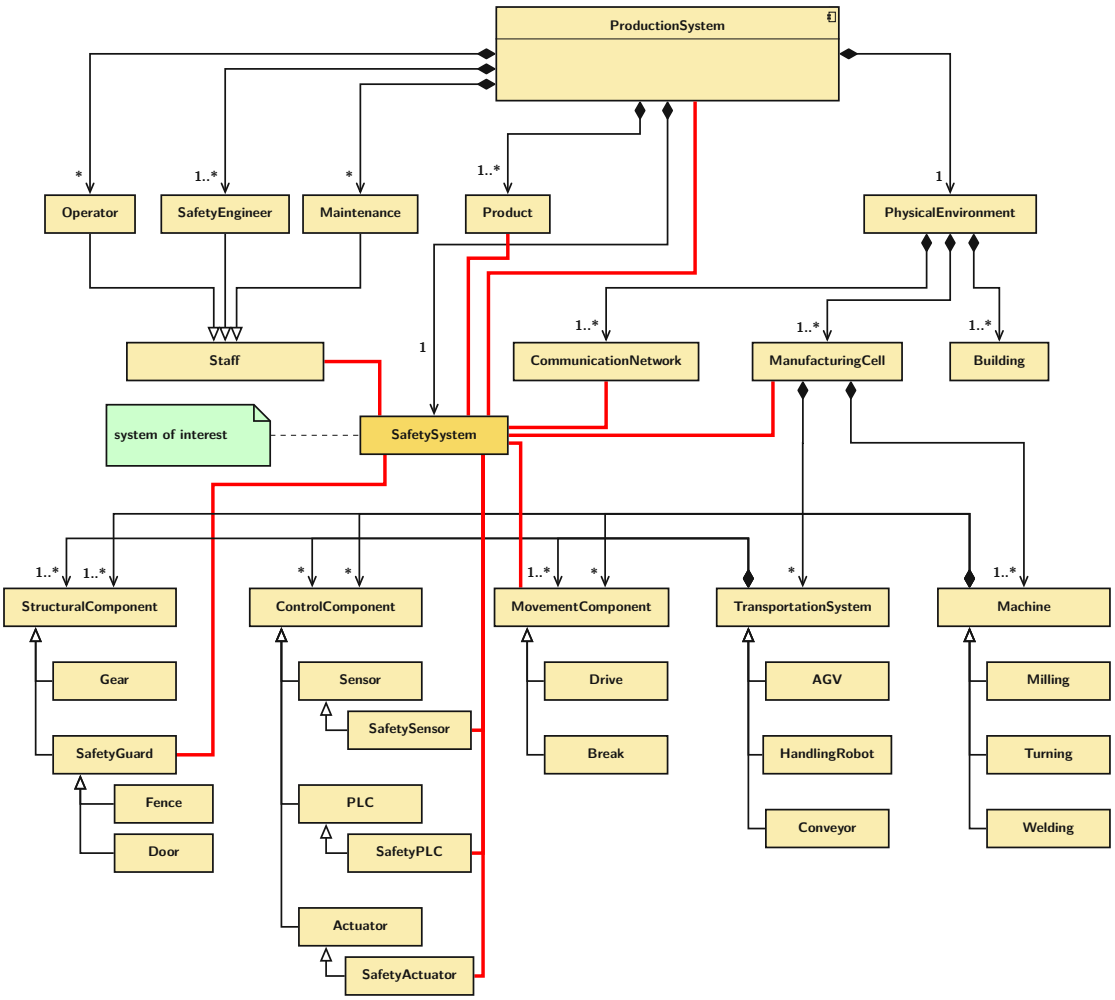


Figure 4.1: Production system model

The red lines in the production system model illustrate the relationships between the safety system and the system components accordingly. These relationships are either for gathering information from the component or for configuration purposes.

4.1.1 Safety Function Properties

Machinery control systems are usually realized by a combination of physical components and software. A safety-related control system implements the required safety functions necessary to achieve a safe state for the equipment under control. Its purpose is to reduce the probability and consequences of a hazard or a set of hazards.

Defining a safety function always includes two components as a fundamental basis:

- Required action – what must be done to reduce the risk
- Safety performance – Safety Integrity Level (SIL) or Performance Level (PL)

Safety functions are the basic building blocks of a safety-related control system. These building blocks, or Safety-Related Parts of Control System (SRP/CS), can be split into four basic functions: input or sensing the hazardous triggering event, processing or logic solving, actuation or bringing the dangerous part of the machine into a safe state, and the communication or connections in between the SRP/CSs. A safety function with its building blocks is depicted in Figure 4.2.

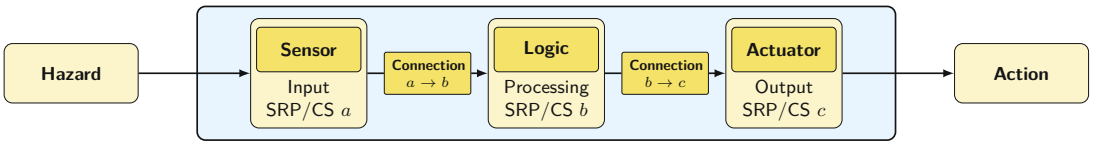


Figure 4.2: Safety function building blocks

ISO 13849-1 defines five distinct designated architectures of SRP/CS for realizing a safety function. The categories range from B, 1, 2, 3 and 4 where category B represents the basic architecture with the least fault tolerance and category 4 represents the designated architecture with the highest fault tolerance. In this context, fault tolerance refers to the ability of the safety system to provide a continued performance of the safety function, which means that even if an SRP/CS is faulty, the safety function will still work [99]. Figure 4.3 illustrates a two channel safety function with monitoring and cross-monitoring.

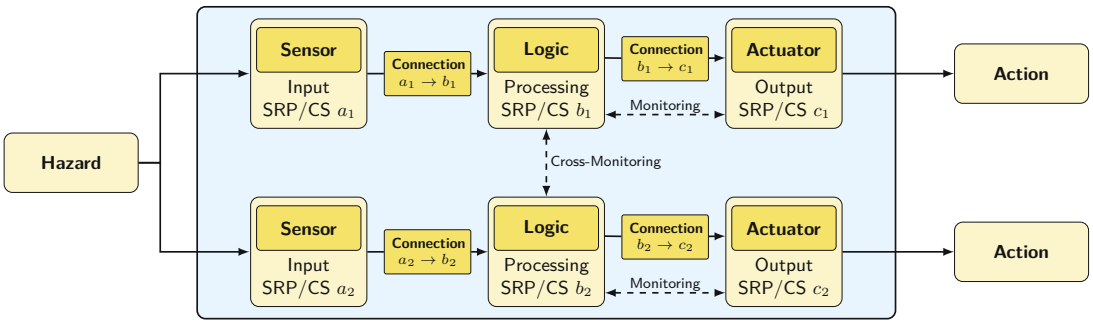


Figure 4.3: Two channel safety function with monitoring

For the implementation of a safety function, the results of the risk assessment must be available in order to derive a suitable architecture. In doing so, the following characteristics of the safety function have to be defined at least:

- Use limits – the machine's operating modes
- Space limits – danger zones
- Time limits – Safety Function Response Time (SFRT)
- Safety system rating – SIL or PL
- Redundancy – number of channels (single/double/multi)
- Fault Detection
  - Diagnostic Coverage (DC) (None/Low/Medium/High)
  - Monitoring
  - Cross-Monitoring

In addition to the overall properties of a safety function described above, the basic building blocks of a safety function sensor, logic, actuator, and communication also have their own properties, which are discussed in the following.

4.1.2 Safety Device Properties

Generally, three types of safety devices can be part of a safety function within a machine: sensors, logic, and actuators. However, for brownfield applications, there is the need for a gateway device to bridge the gap between a legacy system and the Flexible Safety System (FSS). In the following, the specific properties of these devices will be analyzed.

Safety Sensors

Safety sensors are electronic devices that detect the presence and position of objects or people or measure physical characteristics such as temperature or pressure. The detected or measured value is exposed through an interface for further processing. For a safety-relevant sensor, this can be accomplished using Industrial Ethernet and a safety protocol as depicted in Figure 4.4.

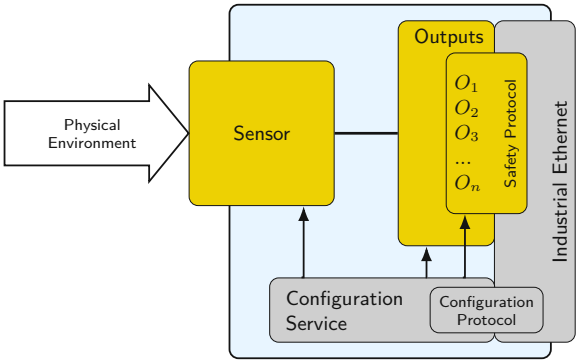


Figure 4.4: Safety sensor with Industrial Ethernet interface

Before the safety-relevant process data can be transferred, there is a need for configuration on several layers. First, the sensing part might need parametrization, such as range. Second, the type and number of output channels have to be defined. Third, the safety protocol and the underlying network have to be configured. These configuration tasks are handled via a configuration service using a configuration protocol, as shown in Figure 4.4.

A distinction can be made based on the underlying technology: mechanical, magnetic, inductive, optical, or radar. Depending on the application, the following devices can be used to detect hazardous situations:

- Emergency stop device
- Safety switches (mechanical, magnetic, inductive, photoelectric)
- Safety light curtains
- Safety laser scanner
- Safety radar systems
- Safety camera systems

These sensors provide binary signals to be used in the safety logic. Sometimes, it is necessary to measure an object's physical properties, such as temperature, pressure, or velocity and process this information in the safety logic. In this case, an analog value is transmitted from the sensor to the logic. Therefore, the output of an input or sensor SRP/CS must support two types:

- Binary output
- Analog output

Certain safety sensors also support additional functions such as muting, which is the safe, automatic, and temporary suspension of electrosensitive protective equipment during operation. For example, it allows material to be transported into and out of a danger zone. Bypassing a portion of the sensing field of a presence-sensing safeguarding device is called blanking. In this case, the protected field is only partially but continuously muted. A distinction is made between two types of blanking in IEC 61496-2: fixed blanking and floating blanking. Floating blanking allows objects of a defined minimum and maximum size to move within the protective field. Muting and blanking are available at certain safety light curtains and safety laser scanners. Table 4.1 summarizes the minimum set of properties which have to be defined for each safety sensor as part of a safety function.

Property	Description
sensor type	technology and operation purpose
safety performance	SIL or PL
channel type	type of output channels (digital/analog)
channel number	number of output channels (redundancy)
communication specific parameters	network and safety protocol
sensor specific parameters	sensor parametrization (e.g., sensing field, muting, blanking)

Table 4.1: Safety sensor properties summary

Safety Logic

In contrast to a standard PLC, a safety PLC has redundant microprocessors, flash memory and RAM that are continuously monitored by a watchdog circuit and a synchronous detection circuit. Also, the input and output circuits are equipped with additional self-checking features. Inputs and outputs to the logic can be implemented as remote signals from and to other devices using Industrial Ethernet technologies and safety protocols. Figure 4.5 illustrates a setup where all inputs and outputs of the logic solver are realized with Industrial Ethernet and a safety protocol.

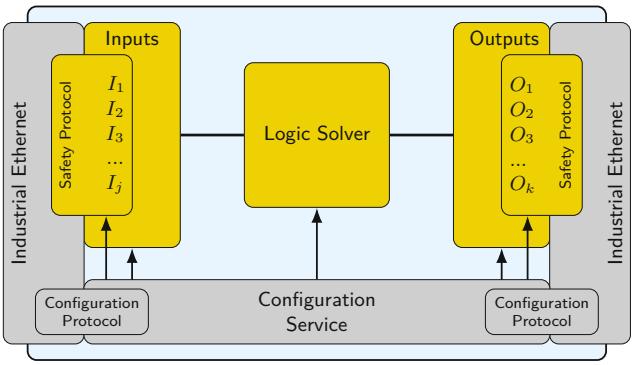


Figure 4.5: Safety logic device with Industrial Ethernet interfaces

A device serving as safety logic needs configuration on different parts and several layers before receiving safety-relevant process data from sensors and sending safety-relevant process data to actuators. First, the logic solver part needs combinatory logic. Second, the type and number of input and output channels have to be defined. Third, the safety protocol and the underlying network has to be configured. These configuration tasks are handled via a configuration service using a configuration protocol, as shown in Figure 4.5. Similar to the outputs of sensors the safety logic must support the two types of signals:

- Binary input/output
- Analog input

Table 4.2 summarizes the minimum set of properties which have to be defined for each safety logic as part of a safety function.

Property	Description
safety performance	SIL or PL
channel type	type of input/output channels (digital/analog)
channel number	number of input/output channels
communication specific parameters	network and safety protocol
logic specific parameters	logic parametrization (e.g., cycle time, combinatory logic)

Table 4.2: Safety logic properties summary

Safety Actuators

The final element in a safety function is the safety actuator. Safety actuators are electronic, electric, or electromechanical devices that bring a hazardous device to a predetermined safe state and ensure that the device remains in a safe condition. The trigger signal for the actuator to bring the device to a safe state and remain there comes from the safety logic through an interface. For a safety-relevant actuator, this can be accomplished using Industrial Ethernet and a safety protocol as depicted in Figure 4.6.

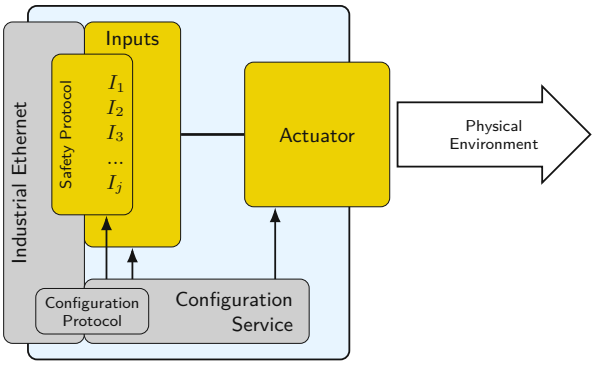


Figure 4.6: Safety actuator with Industrial Ethernet interface

Generally, an actuator is part of a device or machine responsible for achieving physical movements by converting energy, often electrical, air, or hydraulic, into mechanical force. Examples of actuators include electric motors, drives, servos, pneumatic and hydraulic cylinders, and electric muscular stimulators in robots.

Safety actuators are equipped with additional hardware for monitoring to detect the incorrect operation of an actuator which can be used to trigger emergency action. Furthermore cross-monitoring of multiple actuators is used detect faults in actuators by comparing the results. If a discrepancy occurs, emergency action is taken.

One of the simplest safety actuators are safety contactors that are used to remove power from an actuator. Special monitoring features are added to the safety contactor to provide the safety rating. Usually, these are mechanically linked and normally closed contacts that are used to feed back the status of the safety contactors to the safety logic device, thus ensuring the safety function.

For adjustable speed electrical power drive systems with integrated safety functions, IEC 61800-5-2 defines several safety-related functions and divides those into stop functions, shown in Table 4.3, and miscellaneous safety functions such as safe motion functions or safe brake functions, shown in Table 4.4.

Safety actuators reveive binary signals from the safety logic. Therefore, the input of an output or actuator SRP/CS must support one signal type:

- Binary input

Function	Short	Description
Safe torque off	STO	a mechanism that prevents the drive from restarting unexpectedly
Safe stop 1	SS1	power to the actuators is maintained to guarantee a controlled stop
Safe stop 2	SS2	the motor is braked in a controlled manner

Table 4.3: Safe stop functions in accordance with IEC 61800-5-2

Function	Short	Description
Safe direction	SDI	prevents the motor from moving in an invalid direction
Safely limited increment	SLI	allows the motor to travel a permitted distance following a start command
Safely limited speed	SLS	defined transition from the operating speed to the reduced speed
Safe operating stop	SOS	monitors the achieved stop positions of an axis
Safe speed range	SSR	the maximum speed must not exceed a certain value, and the minimum speed must not drop below a certain value
Safely limited position	SLP	ensures that the motor does not exceed a preset position limit value
Safely limited acceleration	SLA	limits the maximum acceleration of the drive and/or prevents the motor from exceeding the set acceleration limit
Safe acceleration range	SAR	monitors the safely monitored acceleration to ensure it is within specified limit values

Table 4.4: Safe motion functions in accordance with IEC 61800-5-2

Furthermore, the safety actuator uses a watchdog timer to react on missing or delayed input signals to bring the device autonomously in a safe state. Depending on the used actuator technology, safety actuators also support additional functions such as double acting actuators, self-lock to avoid actuator drifting, mechanical endstops, quick release, and breaking systems.

Table 4.5 summarizes the minimum set of properties which have to be defined for each safety actuator as part of a safety function.

Property	Description
actuator type	technology and operation purpose
safety performance	SIL or PL
channel type	type of input channels (digital/analog)
channel number	number of input channels (redundancy)
communication specific parameters	network and safety protocol
actuator specific parameters	actuator parametrization (e.g., safe stop/motion function)

Table 4.5: Safety sensor properties summary

Safety Gateway

A gateway device is required to integrate legacy systems, which are part of each brownfield application. Figure 4.7 illustrates a gateway with two Industrial Ethernet interfaces and safety protocols, where the legacy system is on the left and the FSS on the right.

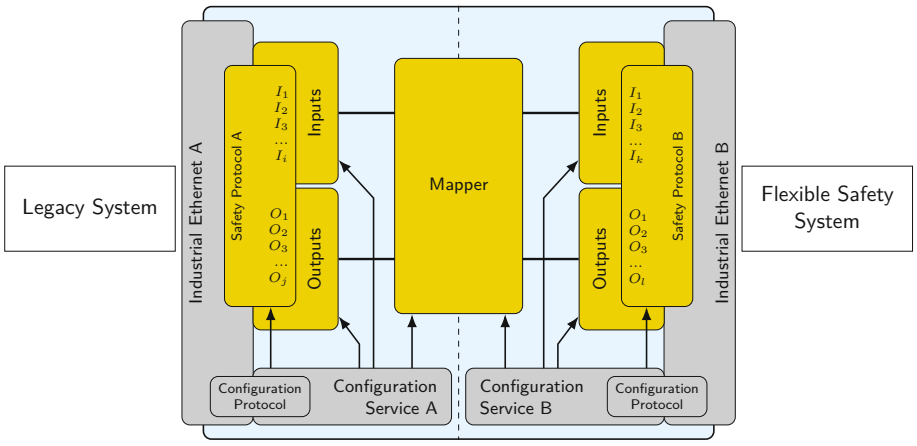


Figure 4.7: Safety gateway device with two seperate Industrial Ethernet interfaces

A device serving as a safety gateway needs configuration on different parts and several layers before receiving and sending safety-relevant process data to and from other safety devices. First, the gateway mapper part needs the mapping of signals on both sides but does not allow any interaction between safety protocols. Second, the type and number of input and output channels have to be defined. Third, the safety protocols on both sides and the underlying network has to be configured. These configuration tasks are handled via the configuration services using a configuration protocol, as shown in Figure 4.7. Similar to the inputs, outputs of sensors, actuators, and logic, the safety gateway must support the two types of signals:

- Binary input/output
- Analog input/output

Table 4.6 summarizes the minimum set of properties which have to be defined for each safety gateway.

Property	Description
safety performance	SIL or PL
channel type	type of input/output channels (digital/analog)
channel number	number of input/output channels
communication specific parameters	network and safety protocol
gateway specific parameters	gateway parametrization (e.g., mapper)

Table 4.6: Safety gateway properties summary

4.1.3 Safety Communication Properties

Besides the safety devices, the safety-relevant communication between those devices is also part of a safety function. Figure 4.8 illustrates the safety-relevant communication using Industrial Ethernet.

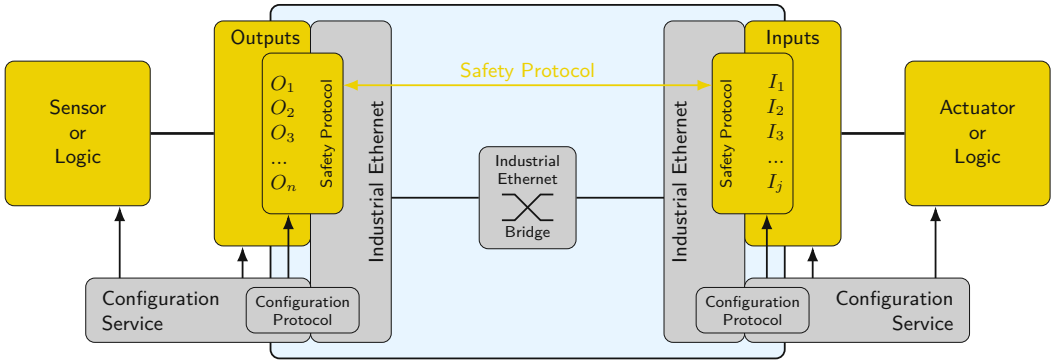


Figure 4.8: Safety communication using Industrial Ethernet

The primary purpose of safety-related communication is to transmit data between safety nodes where additional techniques and measures are applied to protect safety-relevant messages from various threats such as loss, repetition, insertion, and incorrect order using the black channel principle. The foundation for a stable and reliable operation of such a safety protocol is an industrial network that provides deterministic data transmission that includes a low latency, minimal jitter, and minimal packet loss. Furthermore, the same network is used for deterministic real-time traffic and best-effort traffic, which allows the transmission of process or safety data and configuration or parametrization data.

The connection establishment and release for safety-relevant messages require knowledge about the physical network topology, network bridges, and end devices (sensor, logic, actuator). Therefore, the configuration of a connection for safety-relevant communication has to consider the properties of the underlying Industrial Ethernet devices and the properties of the interfaces of the black channel implemented on both end devices.

Table 4.7 summarizes the minimum set of properties which have to be defined for safety communication as part of a safety function.

Property	Description
communication type	technology
safety performance	SIL or PL
channel type	type of input channels (digital/analog)
channel number	number of input channels (redundancy)
communication specific parameters	network and safety protocol

Table 4.7: Safety communication properties summary

4.1.4 Safety Configuration Properties

During risk evaluation, as part of the risk assessment, a set of safety functions has to be defined for all identified hazards to reduce risk adequately. The realization of these safety functions entails configuration on several devices: safety sensors, safety logic, safety actuators, and communication links.

IEC 61511 defines the term Safety Instrumented System (SIS) which is composed of and performs one or several Safety Instrumented Functions (SIFs). This set of SIFs has to be seen as a whole, meaning that a change in one SIF affects the whole SIS and, therefore, its certification. For this reason, the configuration of a SIS must be an atomic composition of all configurations of all included SISs. Figure 4.9 illustrates a SIS with its SIFs and the elements which need configuration.

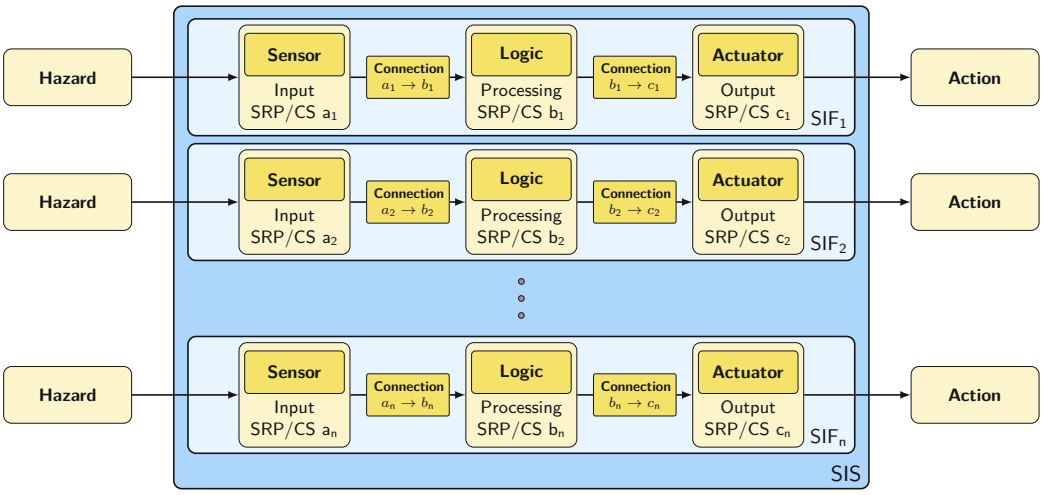


Figure 4.9: Safety system configuration structure

Since the safety configuration of an SIS has to be assessed as a whole, all SIFs need to be operational in order to put the machine into operation. That is especially important when the safety configuration of a machine shall be changed very quickly during operation.

Table 4.8 summarizes the minimum set of properties which have to be defined for a safety system configuration.

Property	Description
ID	uniqe identifier for this configuration
safety function number	number of safety functions as part of this configuration
safety performance	SIL or PL

Table 4.8: Safety configuration properties summary

4.2 Use Case Analysis

Use cases are very helpful tools for understanding user requirements. However, it is incorrect to assume that the use case is all that is needed to fully define what the system must do. Use cases are used to explain and document the interaction that is required between the user and the system to accomplish the user's task and can be used to derive more detailed functional requirements for a system [100].

A use case is characterized by attributes describing how the user envisions deploying, operating, supporting, or disposing of a system, product, or service to achieve a desired performance-based outcome [101].

In the following, use cases and scenarios in the domain of discrete manufacturing are presented with a focus on the convenience of use, rapid reconfiguration, and machinery safety. All the presented use cases are based on the outcomes of several workshops with scientific and industry partners of the Center for Digital Production (CDP) and mapped on the production facilities in the manufacturing area in the Pilotfabrik of TU Wien. The layout of this manufacturing area is depicted in Figure 4.10.

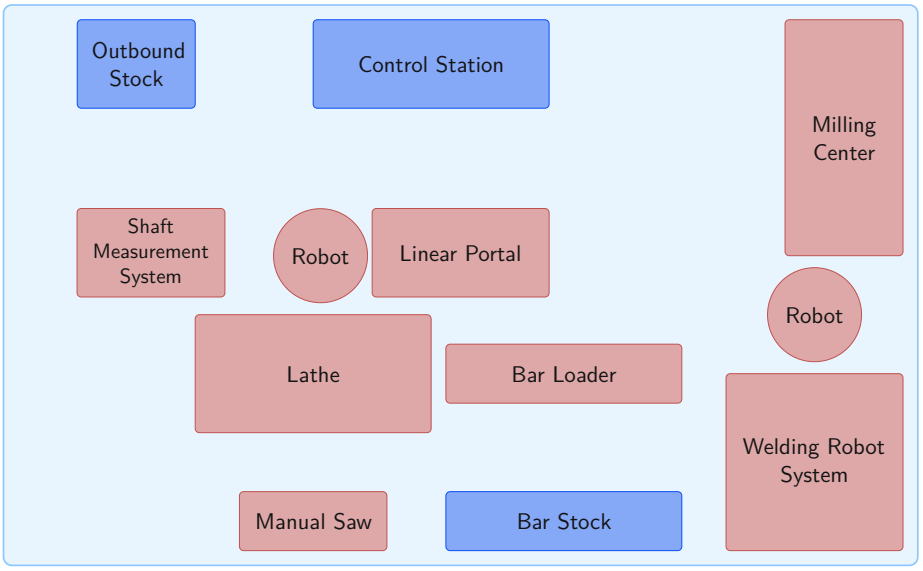


Figure 4.10: Manufacturing area of the Pilotfabrik of TU Wien

The manufacturing area consists of machinery and robots marked in red and passive elements such as control station and stocks marked in blue. The machinery is composed of a milling center (EMCO MAXXMILL500), a welding robot system (IGM), a lathe (EMCO MAXXTURN45) including a fully automatic bar loader, an optical shaft measurement system (Jenoptik Opticline C 308), a linear portal for handling palettes (Festo), an outbound stock station, and a manual saw. Two 6-axis robots complete the setup. One robot arm has 20 kg payload and 1650 mm reach (ABB IRB2600), the other has 150 kg payload and 2200 mm reach (ABB IRB6620).

4.2.1 Plug-and-Produce Manufacturing Cell (UC-01)

This use case is motivated by the objective of rapid reconfiguration of production facilities which is a fundamental condition for the realization of customized mass production and desired by Industry 4.0 manufacturing. Besides the configuration of machinery and their programs, this also affects the configuration of the safety system [20].

A Reconfigurable Manufacturing System (RMS) is designed for rapid change in structure, as well as in hardware and software components [19]. Individual components of an RMS, such as machines, cells, or material handling units, and their software configuration, can be rapidly added, removed, modified, or interchanged in order to respond to changing market requirements, technologies, or regulatory requirements [102][103]. The key feature of rapid reconfiguration of production facilities is also known under the term ”plug-and-produce”. Thereby, a component can be any equipment, from an individual sensor to complex machinery or even complete production cells.

The plug-and-produce manufacturing cell use case consists of a lathe with an automatic bar loader and two optional components, a handling robot and a linear portal. The robot and the portal can be added and removed for machine-loading tasks depending on the produced workpieces. The layout of the cell is depicted in Figure 4.11 with the optional components marked with dashed lines.

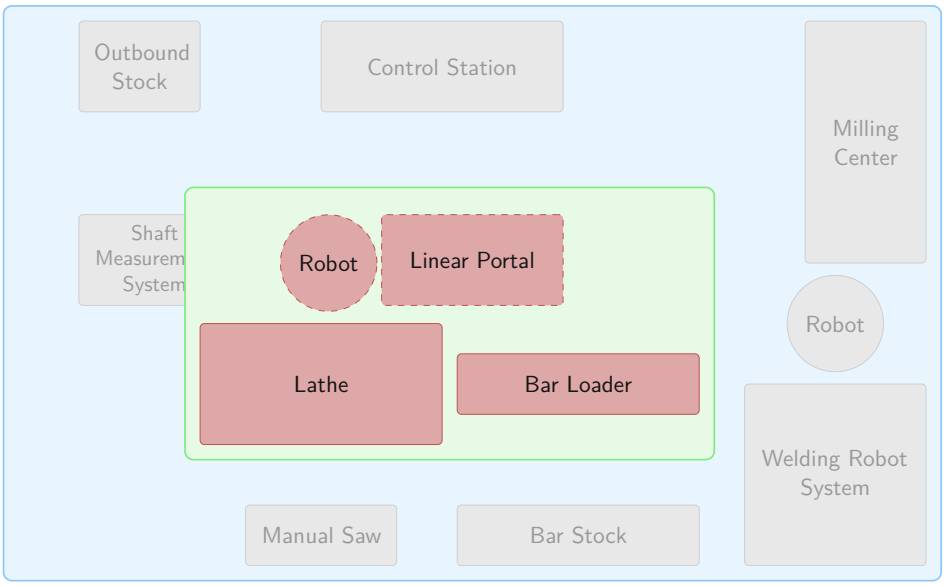


Figure 4.11: Plug-and-produce manufacturing cell

While the production cell is not equipped with the robot or the robot is deactivated, the cell is in manual mode, where the machine operator can manufacture individual parts. By adding or activating the robot, the cell can be operated in semi-automatic mode, meaning that the robot can load and unload the machine using a storage area. In this

scenario, small lot sizes can be handled as the storage area limits it. When the robot and portal are in operation, the cell can produce in a full-automatic mode where an Automated Guided Vehicle (AGV) collects and delivers workpieces at the pickup station, which is part of the portal. In this scenario, the cell can also handle larger lot sizes.

Scenario: Manual Mode

Goal
The operator of the manufacturing cell produces individual parts manually.

Workflow
The operator places the part or raw material to be processed in the machine's work area and performs the configuration and programming directly on the machines' Human-Machine Interface (HMI). The operator is present during configuration and programming and the whole processing procedure.

Scenario: Semi-Automatic Mode

Goal
The operator of the manufacturing cell produces a small batch size of workpieces with the aid of the robot arm.

Workflow
The operator places the workpieces to be processed in the storage area and performs the configuration and programming directly on the machines' HMI. The operator is present during configuration and programming, and the processing procedure is executed automatically. Therefore, during the processing procedure, the operator is not present.

Scenario: Full-Automatic Mode

Goal
The operator of the manufacturing cell produces large lot sizes of workpieces without a machine operator but with the aid of the robot arm and the linear portal. Workpieces are delivered and collected automatically using AGVs.

Workflow
The operator performs the configuration and programming of the manufacturing cell from the control station. An AGV collects and delivers workpieces at the pickup station, which is part of the linear portal. The robot arm picks a workpiece to be processed from the linear portal and places it in the machine's work area. During the whole procedure the operator is not present.

Additional Conditions

When using the lathe and the bar loader only, the safety certification of these machines is sufficient. This situation changes when the robot or the linear portal is added. According to the EC Machinery Directive, these are "partly completed machinery" and, therefore,

must not be put into service until the final machinery into which it is to be incorporated has been declared in conformity with the Machinery Directive. Consequently, the whole cell, marked in green in Figure 4.11, has to be certified according to the relevant safety standards, and potentially, additional safety sensors, such as emergency stop buttons or laser scanners, have to be applied accordingly.

As stated before, the robot or portal requires additional safety sensors before being put into service. These need to be integrated into the cell's safety system. This procedure requires reconfiguration on two levels: First, the various safety functions need to be extended on an application level to take all available safety equipment into account and conform to normative and regulatory guidelines. Second, the underlying communication system must be reconfigured to guarantee the timely and reliable delivery of safety-related messages between components.

4.2.2 Flexible Manufacturing Cell (UC-02)

The efficiency and profitability of a manufacturing cell increase, the better it can be adapted to changing market or customer requirements and organizational conditions. Thus, this use case is motivated by the need for frequent operation mode changes of the manufacturing cell caused by organizational conditions. The layout of the cell is depicted in Figure 4.12 with the path or area where humans may be or move marked in orange.

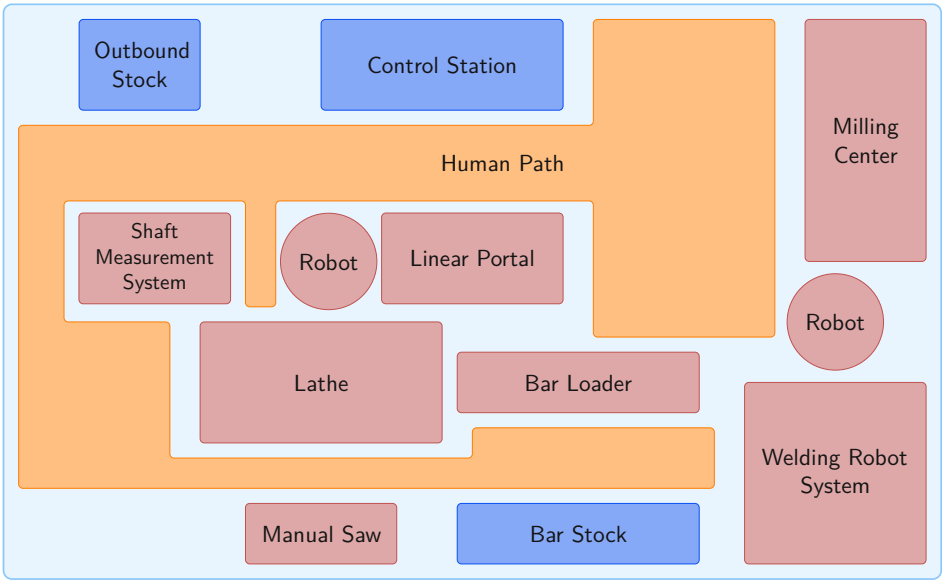


Figure 4.12: Flexible manufacturing cell

This use case considers three manufacturing cell operation modes and derives scenarios accordingly. The cell is operated in three shifts: early shift, day shift, and night shift. In the early shift, the cell is operated in semi-automatic mode, where only the lathe from

UC-01 runs in full-automatic mode. The rest of the machinery is used in manual mode. During the day shift, when most of the personnel is present, all machines in the cell are operated manually. On the night shift, a minimum number of workers is on-site. During that time, the cell operates in a full-automatic mode without human user intervention.

Scenario: Manual Operation

Goal

All machinery of the manufacturing cell is operated in manual mode. All machines must be accessible by workers at any time via the human path marked in orange in Figure 4.12.

Workflow

Several machine operators in the manufacturing cell operate their machines simultaneously. Each machine operator is present during configuration, programming, and the whole processing procedure.

Scenario: Semi-Automatic Operation

Goal

The operator of the manufacturing cell produces workpieces with the lathe and its additional components in full-automatic mode. The milling center, the welding robot system, and the manual saw are operated by workers manually. The manually operated machines must be accessible by workers at any time.

Workflow

Several machine operators in the manufacturing cell operate their machines on the machine's control panel. The lathe is operated from the control station. All machine operators except the lathe operator place the workpieces to be processed in the machine's work area and perform configuration and programming directly on the machines' HMI, respectively. These operators are present during configuration, programming, and the whole processing procedure. The lathe operator configures and controls the machine from the control station.

Scenario: Full-Automatic Operation

Goal

The operator of the manufacturing cell produces large lot sizes of workpieces without machine operators. Workpieces are delivered and collected automatically using AGVs.

Workflow

The operator of the manufacturing cell operates all machinery except the manual saw from the control station. An AGV collects and delivers workpieces to and from the robots. The robot arm picks workpieces to be processed from the AGV or the linear portal and places them in the corresponding machine's work area. In this mode, no machine operator is in the manufacturing cell. Only at the control station, there may be an operator.

Additional Conditions

Similar to UC-01, each individual machine in the cell can be operated safely. The composition of all machines, robots, and linear portal makes it necessary to assess the safety risks as a whole and change the safety configuration of the cell accordingly. Specific attention has to be paid to the area where humans may be or move. That human path or area is marked in orange in Figure 4.12.

4.2.3 Mobile Devices with Safety Features (UC-03)

In a flexible manufacturing cell, mobile devices are indispensable. On one hand, such devices are used for transportation. On the other hand, configuration and monitoring can be accomplished with mobile handheld devices. Mobile devices bring flexibility to the manufacturing cell but also introduce new challenges, especially for the safety system. The motivation for that use case is the additional flexibility mobiles devices entail, consequently enabling more functionality for the manufacturing cell.

The layout of the cell is depicted in Figure 4.13 with the path or area where humans may be or move marked in orange and the AGV path marked in green.

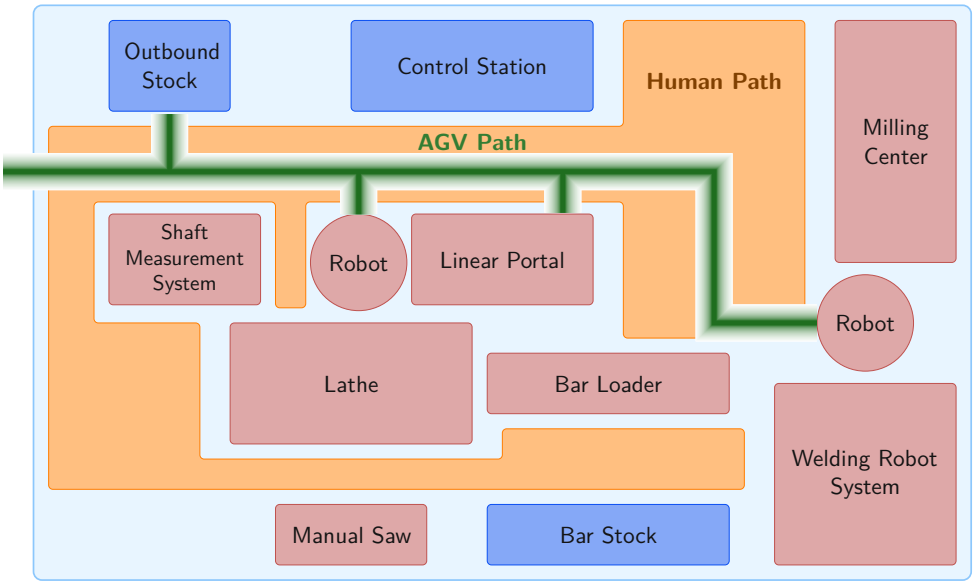


Figure 4.13: Mobile safety devices with safety features

This use case considers two additional components which may be either separately or both added to or removed from or during the operation of the manufacturing cell. Thereby, those mobile devices contain safety-relevant sensors (e.g., emergency stop button), which may change the safety configuration of the whole manufacturing cell upon its discovery within or its removal from the cell.

Scenario: Mobile Operating Panel

Goal

The cell operator enters the cell with a mobile operating panel equipped with an emergency stop button. When that device is outside the cell the emergency stop button is disabled, which is indicated by a grey color of the button. Once the device is inside the cell, the emergency stop button is enabled, indicated by a red and yellow button illumination. When pressing the activated emergency stop button, all machines in the cell shall go to a safe state. When leaving the cell, the mobile operating panel shall be unregistered, and the emergency stop button removed from the cell's safety configuration.

Workflow

Upon entry into the manufacturing cell, the operator is notified on the mobile device about the possibility of connecting the device to the cell, including safety configuration. After the acknowledgement, the emergency stop button is connected to the safety system of the manufacturing cell and consequently illuminated. When leaving the cell, the emergency stop button is disconnected from the safety system of the manufacturing cell, and consequently, the illumination is disabled.

Scenario: Mobile Transportation System

Goal

An AGV equipped with an emergency stop button and a safety actuator that brings the AGV into a safe state enters the manufacturing cell. Once the device is inside the cell, the emergency stop button and the safety actuator are registered and added to the cell's safety system. When pressing the AGV's emergency stop button, all machines in the cell shall go to a safe state. When the cell is brought to a safe state, the AGV shall do that as well. When leaving the cell, the AGV's safety features shall be unregistered and removed from the cell's safety system.

Workflow

Upon entry into the manufacturing cell, the emergency stop button and the safety actuator are connected to the safety system of the manufacturing cell. When leaving the cell, the emergency stop button and the safety actuator are disconnected from the safety system of the manufacturing cell.

Additional Conditions

The composition of all machines, robots, linear portal, and mobile devices makes it necessary to assess the safety risks as a whole and change the safety configuration of the cell accordingly. Specific attention has to be paid to the areas where humans or AGVs may be or move. That human path or area is marked in orange, and the AGV path is marked in green in Figure 4.13. Besides the required reconfiguration of the safety functions, the specific characteristics of the underlying wireless communication system have to be taken into account in order to guarantee the timely delivery of safety-related messages between wirelessly connected components.

4.2.4 Container-Based Safety Applications (UC-04)

A manufacturing cell may have several safety logic controllers, either in a dedicated device or running on a hosted environment (e.g., edge node, fog node, cloud). The possibility of transferring the safety logic between devices increases the reliability of the cell and brings more flexibility at maintenance. This use case is motivated by the idea of moving containerized safety applications, similar to container orchestration systems for automating software deployment.

Scenario: Safety Logic Transfer

Goal
Due to the maintenance or removal of a device that hosts the safety logic controller of a manufacturing cell, the safety logic with its safety connections shall be seamlessly transferred to a different device.

Workflow
The cell operator initiates the transfer by issuing a command to the container orchestration. After a resource check on the destination node, the orchestration transfers the container and its connections to the destination node.

Additional Conditions

Safety applications pose higher requirements on the hardware they are executed on. Also, connection management needs particular attention since the safety-relevant connections require properties for deterministic data transfer. Especially when changing connections during operation timing and sequence of that change are crucial.

4.3 Requirements Determination

Determining the requirements is a critical step in the early stage of the Flexible Safety System Development Life Cycle (FSS-DLC) presented in Section 3.2. Requirements determination involves studying the existing system and gathering details to derive the requirements for the proposed system. Its purpose is to transform high-level business requirements into detailed requirements that can be used as inputs for creating models, designs, and architectures.

Requirements can be categorized in several ways. In the following, requirements are grouped under three primary headings: operations, functions, and performance. Operational requirements pertain to the scenarios within which the system will operate. Functional requirements pertain to the physical plant and the situations within which the system will reside. Performance requirements or non-functional requirements pertain to the metrics and parameters describing the system's capabilities [104].

Requirements analysis involves defining customer needs and objectives in the context of planned customer use, environments, and identified system characteristics to determine

requirements for system functions [105]. Figure 4.14 shows the transformation of use cases with inclusion of technology enablers and controls into system requirements, these requirements serve as a basis for the system model in the next step.

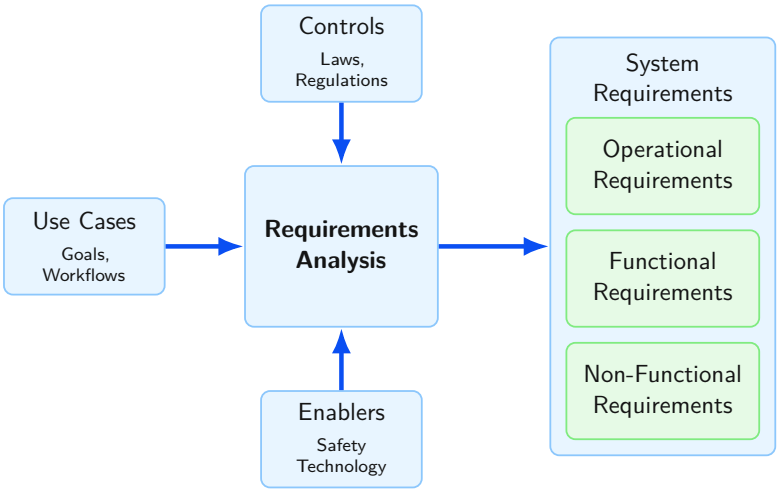


Figure 4.14: Requirements analysis

The requirements analysis is based on use cases, safety regulations, properties of safety functions, safety devices, safety communication, safety configuration, and prior publications [46] [18] [47] [50] [20].

In the following, a non-exhaustive list of operational, functional, and non-functional requirements, which emerged during the work on this thesis at meetings and workshops with scientific and industrial stakeholders, is compiled. Therefore, the listed requirements have to be regarded as a minimal basis that can be extended.

4.3.1 Operational Requirements

OR-01 A manufacturing company wants to produce workpieces using discrete manufacturing cell in an industrial production facility.

OR-02 The manufacturing cell is operated by workers, by orchestration systems, or both.

OR-03 Raw materials or workpieces are delivered and collected by workers and AGVs.

OR-04 The manufacturing cell shall be operated manually, semi-automatically, and full-automatically.

OR-05 Roles needed within the cell are cell operators, machine operators, and machine maintainers.

OR-06 Only trained personnel with knowledge and experience in handling the machinery may work in the manufacturing cell.

OR-07 During maintenance, parts of the cell must be disabled and accessible by maintenance personnel.

OR-08 All components must be operated in an industrial environment with high requirements on conditions such as dust, solvents, light strength, temperature, or noise.

OR-09 The system shall be operated and monitored by manufacturing personnel without expert knowledge (operation mode changes).

OR-10 The system shall be configured only by experts with proper authentication.

4.3.2 Functional Requirements

FR-01 Devices or machines as part of such a unified infrastructure must be able to describe and advertise their capabilities.

FR-02 Devices or machines as part of such a unified infrastructure must be able to discover capabilities of other devices.

FR-03 Devices must support the communication model client/server for data transmission upon request such as configuration, data logging, and analysis.

FR-04 Devices must support the communication model publish/subscribe for process data transmission.

FR-05 The safety communication platform must provide mechanisms to maintain a secure channel ensuring confidentiality, integrity, and availability of data and services.

FR-06 The safety communication platform shall support deterministic real-time data transmission for periodic safety process data.

FR-07 The safety communication platform shall support best-effort non-real-time capabilities for configuration, monitoring, and logging.

FR-08 The safety communication platform must conform to applicable legal regulations (e.g., IEC 62061, ISO 13849-1).

FR-09 The safety communication platform shall use the black channel principle recommended by IEC 61784-3.

FR-10 The safety communication platform must provide mechanisms to prevent potential errors from occurring during data transmission.

FR-11 The safety communication platform must support seamless configuration changes during operation.

FR-12 Safety devices from various manufacturers must communicate safety-related messages among each other.

FR-13 It must be possible to change the configuration of safety devices with one engineering tool.

FR-14 The discovered devices, network topology, and safety links must be visualized on an HMI.

FR-15 It must be possible to edit safety functions, safety links, devices, and network links via an HMI.

FR-16 Upon a safety-relevant change in the manufacturing cell, a new configuration shall be selected, deployed, and activated.

FR-17 All changes and events in the safety system have to be automatically documented.

FR-18 An access right management system must control the access for roles and users.

FR-21 The system shall store predefined safety configurations which can be deployed on request (operating mode selection).

FR-22 The system shall support the user in creating safety functions.

FR-23 The system shall support the user in generating the needed configuration files for deployment.

FR-24 A completed safety configuration must be tagged as verified by the safety engineer.

FR-25 A tool must support the safety engineer by semi-automatically verifying a safety configuration in progress against relevant safety regulations.

FR-26 A deployed and validated safety configuration must be tagged as validated by the safety engineer.

FR-27 A tool must support the safety engineer in the procedure to validate a verified and deployed safety configuration.

FR-28 The system shall support a maintenance mode which can be activated and deactivated only by authorized maintenance personell or a safety engineer.

FR-29 A notification mechanism shall inform registered users about changes in the system.

FR-30 The system shall support hosted container applications (e.g. safety logic on a fog node).

FR-31 The system shall support the transfer of container applications between hosts.

FR-32 Container applications shall be configured in the GUI.

4.3.3 Non-Functional Requirements

NR-01 All safety-relevant devices have to comply with local safety certification.

NR-02 The engineering tool for the safety system has to comply to the relevant tool qualification class.

NR-03 A new device or network link shall be discovered and presented on the HMI within 10 sec.

NR-04 A removed device or network link shall be detected and notified on the HMI within 30 sec.

NR-05 Deployment of a new or changed configuration shall be completed within 1 sec.

NR-06 An interface shall be provided for controlling the safety configuration from an cell orchestration system.

NR-07 An interface shall be provided for a web-based graphical user interface and cloud application.

NR-08 An interface shall be provided for error checking and debugging with authentication.

NR-09 An interface shall be provided for additional services in the future.

NR-10 The system shall be designed in modular and service-oriented way, enabling containerization of all components.

NR-11 It must be possible to transmit data with encryption.

NR-12 The system shall be designed to run 24/7.

NR-13 The transmission delay of the communication platform must be <500ms.

NR-14 All data transmission (e.g., process data, safety-related data, configuration, monitoring) in the manufacturing cell shall use the same communication network.

4.4 Functional Safety System Structure

The structure of a functional safety system is determined in principle by safety standards as described in Section 2.5. Here, a distinction is drawn between safety communication and safety configuration. These fundamental aspects, illustrated in Figure 4.15, need to be taken into consideration by each safety-related system.

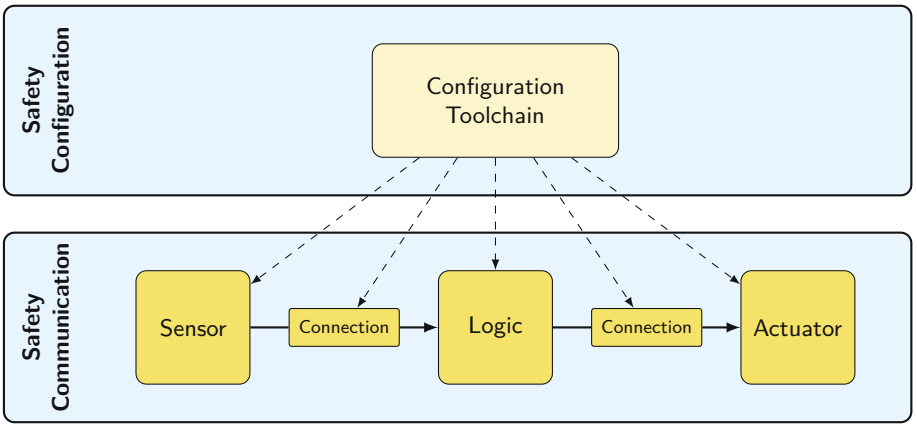


Figure 4.15: Basic functional safety system structure

4.4.1 Safety Communication

Safety communication concerns the exchange of safety-related information between two safety-related subsystems (e.g., sensor, logic, actuator). Specific requirements for this information exchange are posed on the underlying safety-related data transfer, as already stated in Section 2.6, that include measures against potential threats such as transmission errors, repetitions, deletion, insertion, resequencing, corruption, delay, and masquerade. Additionally, the underlying safety-related data transfer should be based on the deterministic transmission of data, which includes a low latency, minimal jitter, and minimal packet loss to ensure the system's reliability.

4.4.2 Safety Configuration

A safety-related system comprises several subsystems that need configuration to fulfill their assigned role. Therefore, qualified tools that fulfill particular requirements are needed to carry out the necessary configuration tasks, as outlined in Section 2.9. These tools are also supposed to help and assist the operator in managing and configuring a safety-related system quickly and safely. Furthermore, configuration deployment, as part of the configuration tasks, places additional requirements on the network, such as best-effort traffic, besides the deterministic real-time traffic used for safety-related data on the same network infrastructure.

CHAPTER 5

Technology Integration

> THE RIPPLE OF A NEW TECHNOLOGY THROUGHOUT THE ECONOMY LEADS TO EFFECTS THAT ARE NOT PREDICTABLE BY EXAMINING EACH INDUSTRY IN ISOLATION. EVERY DECISION TO INTRODUCE TECHNOLOGY COULD BE BASED ON DATA AVAILABLE TO ALL. [106]
>
> – Wassily Leontief, 1985

So far, the topic of functional safety has been considered independently of technology. This chapter examines existing communication technologies that fulfill the requirements from the previous chapter and investigates if these technologies can be used to build a solid bedrock for flexible safety communication.

Three base technologies for deterministic data transmission, interoperable information exchange, and safety-related communication are presented in detail and combined to create a comprehensive safety communication platform. Besides the three base technologies, this communication platform also includes configuration aspects and considers the requirements from the previous chapter. Furthermore, a comprehensive toolchain for safety system configuration is proposed. This toolchain addresses the importance of worker or operator assistance in a complex configuration and deployment process comprising several base technologies and many safety and network devices.

The proposed safety communication platform, including configuration aspects of the three base technologies and a comprehensive configuration toolchain, will serve as a basis for the ensuing system design that also implies services for worker assistance.

5.1 Base Technologies

As part of this research, communication technologies should be evaluated in order to find the best candidates who can serve as a communication platform for safety-critical and configuration data. This platform should be based on standardized, vendor-neutral technologies that support interoperability and are industrially mature.

As stated in Section 2.6, safety-related communication requires specific characteristics of the transport layer on which a safety layer can be based. One of the characteristics is reliable data transfer based on a deterministic real-time data transmission and best-effort data transfer on the same infrastructure [84]. Additionally, there is the need for technical, syntactic, and semantic interoperability as discussed in Section 2.7.

As an alternative to the combined deterministic data transfer for safety-critical process data and best-effort data transfer for configuration data, it is possible to use over-provisioning, isolation, prioritization, Weighted Fair Queuing (WFQ) and similar prioritization schemes, congestion detection, and congestion avoidance. However, these techniques suffer from one or more of the following difficulties: statistical vs. deterministic, predictability, corner cases, dynamism, standardization, expense and flexibility [107]. For this reason, they will not be discussed further.

Three very promising candidates that fulfil the requirements for standardized, vendor-neutral, and interoperable safety communication are Time-Sensitive Networking (TSN), OPC Unified Architecture (OPC UA), and OPC UA Safety [108], [109], [110], [111]. Therefore, these technologies will be described in detail in the following.

5.1.1 Time-Sensitive Networking

TSN is a set of IEEE 802 Ethernet sub-standards that are defined by the IEEE TSN task group. Each of these standards offers a different set of functionality that can be applied to IEEE 802 and other networks, including the well-known 802.3 wired Ethernet, 802.11 wireless local area network (WLAN), and 5G cellular networks defined by the 3rd Generation Partnership Project (3GPP).

The objective of TSN is to provide deterministic services for IEEE 802 and other networks with the following characteristics:

- Guaranteed packet transport
- Low packet loss
- Bounded low latency
- Low packet delay variation

Depending on the application and specific requirements, various TSN standards can be combined and must be supported by the network infrastructure. These standards cover, among other aspects, time synchronization, traffic scheduling, performance optimization, and configuration [17].

Basic Terminology

IEEE 802.1 defines a TSN Stream as a unidirectional flow of data from a Talker to one or more Listeners, as shown in Figure 5.1. During the forwarding process at a bridge, QoS functions are applied to the frames of a TSN Stream (e.g., filtering and policing, shaping, and queuing).

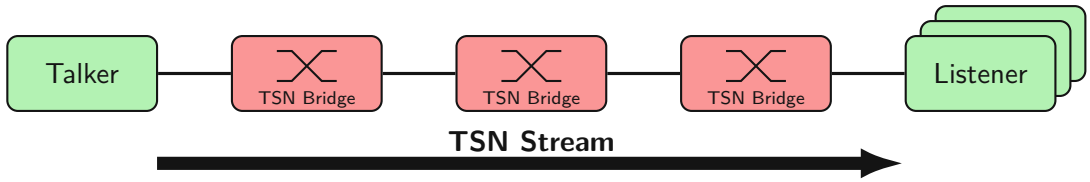


Figure 5.1: TSN stream

The four key components of TSN are synchronization, reliability, latency, and resource management. These components, depicted in Figure 5.2, serve as a foundation to provide best-effort, rate-constrained, and time-triggered traffic on one communication infrastructure. Best-effort traffic has low priority without timing or delivery guarantees. Rate-constrained traffic defines a bandwidth limit for each flow using minimum inter-frame intervals and maximum frame size. Time-triggered traffic transmits each flow at an accurate time.

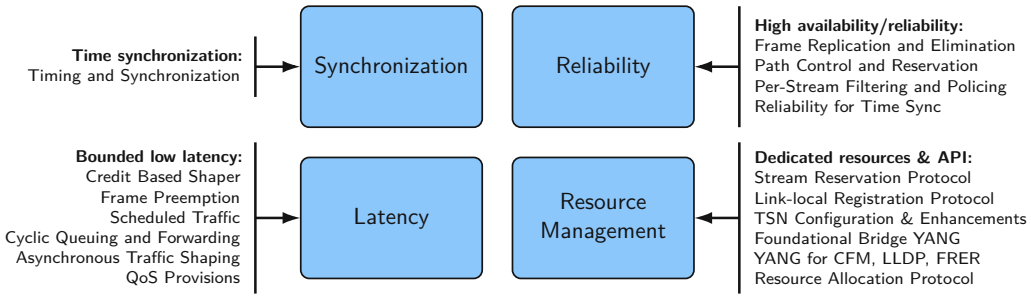


Figure 5.2: TSN components

One of the most critical components for deterministic data transfer is clock synchronization which provides a global notion of time with a predefined precision. Therefore, clock synchronization is the basis for coordinating distributed access to shared resources [112]. TSN mainly addresses reliability and availability through information redundancy (frame replicas) and link redundancy (multiple paths) by transmitting redundant replicas in the network in parallel on disjoint paths [113]. In industrial networks, deterministic delays are essential, especially for real-time control processes and safety-related applications. Deterministic ultra-low latency is provided by TSN, which increases efficiency for operations that require consistent message exchange [114]. Key elements required in the context of resource management include the configuration of communication streams in the network (bridges) and configuration of the end devices (talker and listener) [115].

Standards

Networks for industrial applications need to transport time-sensitive traffic through multiple hops from sensors to controllers and from controllers to actuators. An industrial network may have multiple of these traffic streams in parallel, with varying requirements for latency, packet delay variation, and bandwidth. Additionally, some time-sensitive applications may require extremely high reliability [116].

Features and characteristics of a network using TSN can be composed as needed by the application. The base standards listed in Table 5.1 can be combined for that purpose. Standards, such as IEEE 802.1AS-Rev (Timing and Synchronization), IEEE 802.1Qbv (Enhancements for Scheduled Traffic), and IEEE 802.1Qbu (Frame preemption) provide the necessary extensions to enable deterministic real-time communication on IEEE 802 based networks [117], [118].

Standard	Title
IEEE 802.1Q-2018	Bridges and Bridged Networks
├─ IEEE 802.1Qca-2015	Amendment 24: Path Control and Reservation
├─ IEEE 802.1Qbv-2015	Amendment 25: Enhancements for Scheduled Traffic
├─ IEEE 802.1Qbu-2016	Amendment 26: Frame Preemption
├─ IEEE 802.1Qci-2017	Amendment 28: Per-Stream Filtering and Policing
├─ IEEE 802.1Qcr-2020	Amendment 29: Cyclic Queuing and Forwarding
├─ IEEE 802.1Qcc-2018	Amendment: Stream Reservation Protocol (SRP) Enhancements
├─ IEEE 802.1Qcp-2018	Amendment: YANG Data Model
└─ IEEE 802.1Qcx-2020	Amendment: YANG Data Model for Connectivity Fault Management
IEEE 802.1AB-2016	Station and Media Access Control Connectivity Discovery
IEEE 802.1AS-2020	Timing and Synchronization for Time-Sensitive Applications
IEEE 802.1AX-2020	Link Aggregation
IEEE 802.1BA-2021	Audio Video Bridging (AVB) Systems
IEEE 802.1CB-2017	Frame Replication and Elimination for Reliability
IEEE 802.1CM-2018	Time-Sensitive Networking for Fronthaul
IEEE 802.1CS-2020	Link-local Registration Protocol

Table 5.1: Base standards for TSN

Besides the deterministic data transfer, these standards define additional functions, such as a fault-tolerance mechanism called Frame Replication and Elimination for Reliability (FRER), that offers highly reliable communication for time-triggered traffic [119].

Profiles

A TSN profile narrows the focus, which eases interoperability and deployment. It selects features, options, defaults, protocols, and procedures of bridges, end stations. Furthermore, it describes how to build a network for a particular use and provides configuration guidelines if needed.

Several TSN Profiles have been defined for different application areas as listed in Table 5.2. Some of them are still ongoing projects indicated with a leading 'P'.

Profile	Title
IEEE 802.1BA	Audio Video Bridging (AVB) networks
IEEE 802.1CM	TSN for Fronthaul
└ IEEE 802.1CMde	Amendment on Sync enhancements
IEC/IEEE P60802	TSN Profile for Industrial Automation
P802.1CQ	Multicast and Local Address Assignment
P802.1DC	Quality of Service Provision by Network Systems
P802.1DG	TSN Profile for Automotive In-Vehicle Ethernet Communications
P802.1DF	TSN Profile for Service Provider Networks
P802.1DP	TSN for Aerospace Onboard Ethernet Communications

Table 5.2: TSN profiles

TSN Profile for Industrial Automation (IEC/IEEE 60802)

This standard defines time-sensitive networking profiles for Industrial Automation (IA). The profiles select features, options, configurations, defaults, protocols, and procedures of bridges and end stations, to build industrial automation networks [120].

Industrial network applications typically include control loop operation which are based on three main types of building blocks:

- Sensor applications (providing input measurements indicating the state of a parameter being monitored or controlled)
- IA-controller applications (operate on combinations of measurements and external demand settings to develop output requests)
- Actuator applications (implement output requests as physical changes to the process or machine under control)

A control loop is formed when the process or machine responds to the actuator output and produces a new measured value at the sensor. The complete loop is shown in Figure 5.3 where the application IA-devices are connected as end stations of a TSN infrastructure.

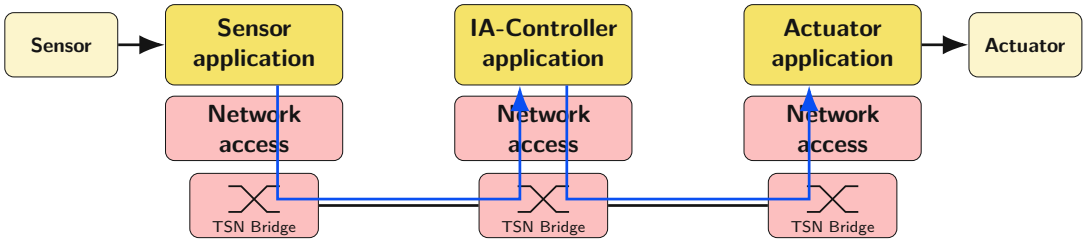


Figure 5.3: Industrial control loop application

Industrial automation applications use different traffic schemes/patterns for different functionalities (e.g., parameterization, control, alarming). The various traffic patterns have different characteristics and, thus, impose different requirements on a TSN network and its features. Table 5.3 shows the industrial automation traffic types defined in IEC/IEEE 60802 with their associated characteristics.

	Characteristics							
Traffic types	Data transmission periodicity	Period	Data transmission time is synchronized to network	Data delivery requirements	Tolerance to interference	Tolerance to loss	Application data size	Criticality
Isochronous	Cyclic/ periodic	100µs to 2ms	Yes	Deadline	0	No	Fixed 30 to 1000 bytes	High
Cyclic-Synchronous	Cyclic/ periodic	500µs to 1ms	Yes	Latency	Yes	No	50 to 1000 bytes	High
Cyclic-Asynchronous	Cyclic/ periodic	2 to 20ms	No	Latency	Yes	1 to 4 frames	Fixed 50 to 1000 bytes	High
Alarms and Events	Acyclic/ sporadic	N/A	No	Latency 100ms to 1s	N/A	Yes	Variable 50 to 1500 bytes	High
Configuration & Diagnostics	Acyclic/ sporadic	N/A	No	Bandwidth	N/A	Yes	Variable	Medium
Network Control	Cyclic/ periodic	50ms to 1s	No	Bandwidth	Yes	Yes	Variable 50 to 500 bytes	High
Best Effort	Acyclic/ sporadic	N/A	No	None	N/A	Yes	Variable 30 to 1500 bytes	Low
Video	Acyclic/ sporadic	N/A	No	Latency	N/A	Yes	Variable	Low
Audio/Voice	Acyclic/ sporadic	N/A	No	Latency	N/A	Yes	Variable	Low

Table 5.3: IEC/IEEE 60802 industrial traffic types and their characteristics

IEC/IEEE 60802 defines several requirements regarding the physical layer (PHY and MAC), the network discovery, the clock synchronization, the network bridges, scheduling, and configuration. Therefore, in the following, the fundamental aspects of clock synchronization, scheduled traffic, and network management will be briefly explained.

Clock Synchronization

Clock synchronization is derived from the network scheduling requirements and the applications built on top of it to provide guaranteed minimum latency for time-critical traffic. IEC/IEEE 60802 require bridge and end station to implement the Precision Time Protocol (PTP) as defined in IEEE 1588. PTP provides several profiles, where IEEE 802.1AS-2020 (Timing and Synchronization for Time-Sensitive Applications) is the standard that has been adopted for TSN operation. The maximum tolerable deviation from the grandmaster time is set by IEC/IEEE 60802 to $0.1 - 1\mu s$ for appropriate network operation in factory automation [85].

Scheduled Traffic

To provide best-effort, rate-constrained, and time-triggered traffic on one shared network, the TSN Task Group defined the standard IEEE 802.1Qbv-2015 (Enhancements for Scheduled Traffic), which is an amendment to IEEE 802.1Q (Bridges and Bridged Networks). It introduces new traffic types, such as time-triggered traffic, which is scheduled via Gate Control Lists (GCLs) specified for each queue of an egress port, as illustrated in Figure 5.4 [121].

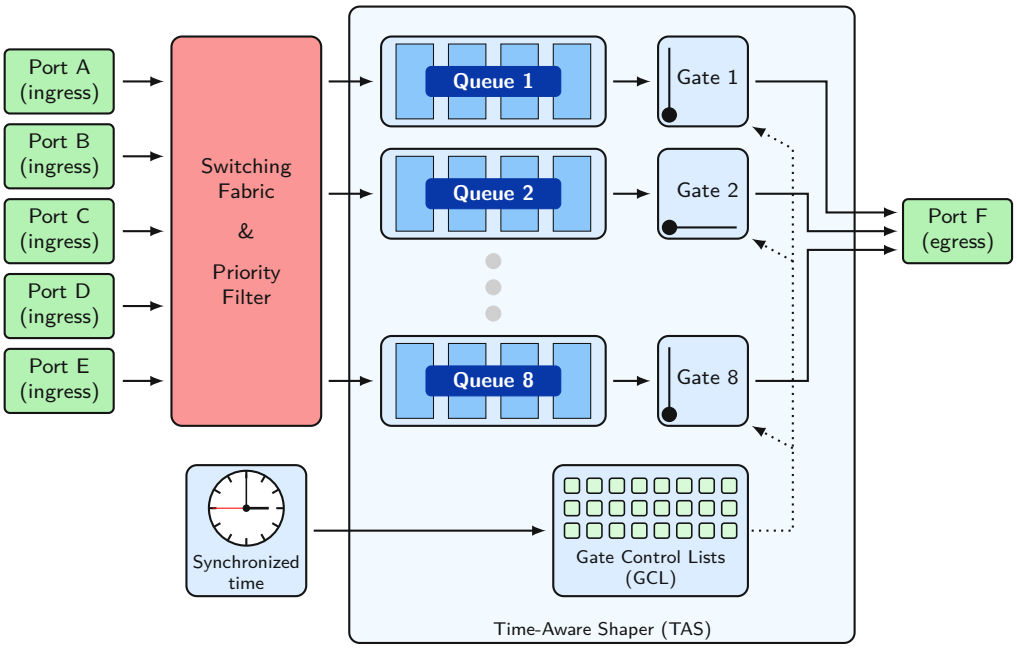


Figure 5.4: TSN traffic scheduling

In order to allow different traffic classes to be transmitted on an egress port, IEEE 802.1Qbv defines a Time-Aware Shaper (TAS) to achieve low latency for time-triggered traffic by establishing completely independent time windows by taking advantage of the globally synchronized clock.

Network Management

The management and configuration of TSN are covered in IEEE 802.1Qcc, which discusses three possible network management approaches. The three models are the fully distributed model, the centralized network/distributed user model, and the fully centralized model. Additionally, managed objects are the configuration aspects of each function defined in the standards and are meant to be adjusted to achieve the desired behavior. For example, in the case of a TAS, managed objects allow to set and retrieve the state of the per port gates that handle the scheduled traffic.

The main elements for the configuration of TSN networks are the endpoints (talkers and listeners) and the network devices (bridges). From a user perspective, the requirements for the streams are specified without having all the details about the network. This information, along with network capabilities, is analyzed, and subsequently, a configuration is generated, which is deployed into the network devices [115].

The decentralized management approach or **fully distributed model** is similar to the Stream Reservation Protocols (SRPs) that has already been in existence with Audio Video Bridgings (AVBs). Talkers announce the streams they offer, and Listeners can subscribe to them. The user transmits its requirements, and the network propagates them through the relevant paths. The network then reserves the necessary resources and time slots along the way. Whether the stream's requirements can be satisfied is detected and computed decentralized within the network. The management of the bridges is performed locally, just with the information available to that bridge. No central management entity is necessary in this approach. Figure 5.5 depicts the fully distributed model.

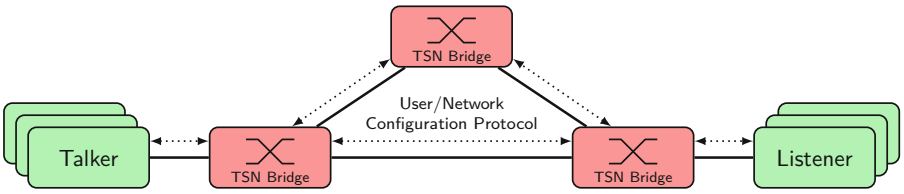


Figure 5.5: TSN fully distributed model

One limitation of the fully distributed model is the lack of a centralized view with a comprehensive overview and complete knowledge of the network devices and topology. This circumstance makes this model not suitable for the configuration of the TAS as part of all egress ports in the network bridges.

The **centralized network/distributed user model** is a hybrid approach that uses a centralized network manager by introducing the Centralized Network Configuration (CNC). However, the data from the end devices is passed to the edge bridge closest to the end device over a standardized protocol and then forwarded directly to the CNC. The CNC has complete knowledge of the network topology and the bridge's capabilities, enabling it to perform the necessary calculations for the required features (e.g., TAS, frame replication, and elimination or frame preemption). Here, only the management of the bridges is performed by the CNC using a network management protocol, excluding the management of end stations. The CNC can exist in an end station, bridge, or a separate device. The centralized network/distributed user model is shown in Figure 5.6.

In the **fully centralized model**, end devices communicate their requirements regarding streams directly towards the centralized management system over a user defined protocol. The centralized management then collects the information and uses them to compute the necessary schedule of streams in the network to satisfy those requirements and configures the network devices accordingly. The fully centralized model is illustrated in Figure 5.7.

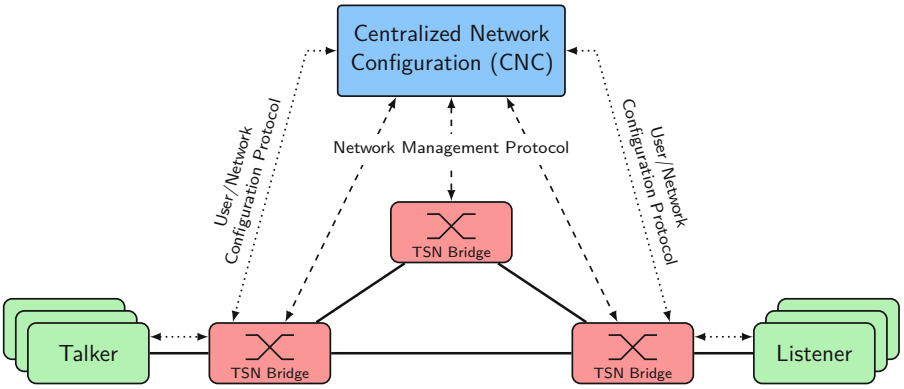


Figure 5.6: TSN centralized network/distributed user model

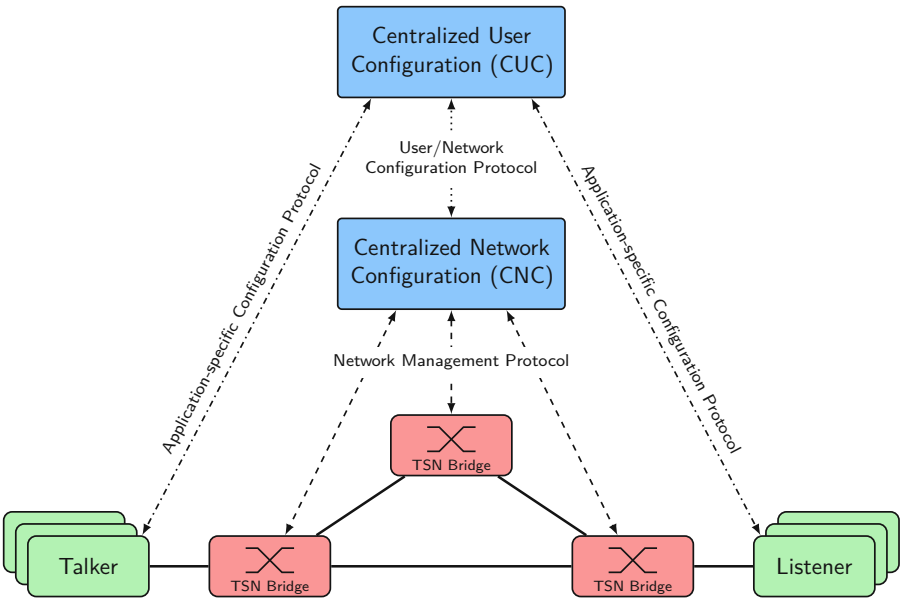


Figure 5.7: TSN fully centralized model

In the two previous models, the configuration of end stations was not covered. However, there are use cases for highly critical applications, such as industrial control, in which complex timing requirements need extra configuration. For those cases, the notion of the Centralized User Configuration (CUC) is introduced in this model. Thereby, the talkers and listeners communicate their requirements to the CUC, which exchanges this information with the CNC.

The CNC and the CUC can be implemented together or separately in an end station, bridge, or a separate device. The definition of the communication protocol between the CUC and the end stations is considered to be out of the scope of IEEE 802.1Qcc.

For remote configuration of TSN bridges IEEE 802.1Qcc assumes the use of a network management protocol such as the Simple Network Management Protocol (SNMP) using Management Information Base (MIB) modules, Network Configuration Protocol (NETCONF) using YANG models, and RESTCONF using YANG models.

5.1.2 OPC UA

OPC UA offers a platform independent service-oriented architecture for Machine-to-Machine (M2M) communication that has been standardized as IEC 62541. The components of OPC UA include transport mechanisms, information modeling capabilities, and services. The transport mechanisms support one-to-one, one-to-many, and many-to-many communication. Information modeling defines the rules and building blocks required to expose managed data with OPC UA. Services allow clients to interact with the application and information model on OPC UA servers. Additionally, OPC UA companion specifications allow to map concepts and technologies to standard models for representation in the OPC UA domain [122].

Components

Besides the data transport, data modeling, and base services, OPC UA defines additional built-in components for Data Access (DA), Alarms and Conditions (AC), Historical data Access (HA), and Programs (Prog) to model more complex and stateful functionality in the system. Furthermore, other organizations can specify their own specific information source, the so-called companion specifications, or a vendor-specific extension. These components are illustrated in Figure 5.8.

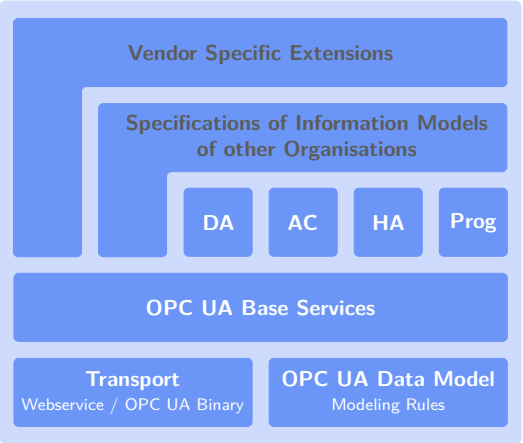
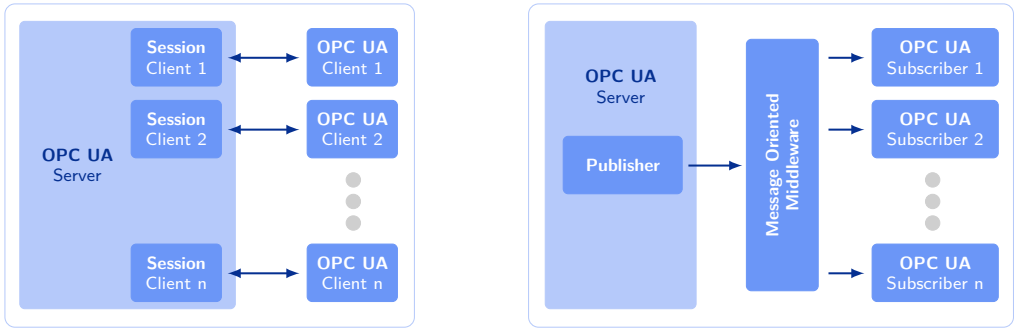


Figure 5.8: OPC UA components

Communication Models

Initially, OPC UA was published based on a client-server communication model. Later, Part 14 of the OPC UA specification extended it by a Publish-Subscribe (PubSub) communication model [123]. For client-server communication, the full range of information model access is available via services following a Service-Oriented Architecture (SOA) design paradigm. Thereby, a client sends a request to a service provider who processes them and sends the results back with the response. The PubSub communication model provides an alternative mechanism for data and event notification. While in client-server communication, each notification is for a single client with guaranteed delivery, PubSub has been optimized for one-to-many or many-to-many configurations [124].

While client-server communication requests and responses are exchanged directly between OPC UA applications on the client and server, this differs from PubSub. At PubSub, publishers send messages to a Message Oriented Middleware without the knowledge about subscribers. Similarly, subscribers express interest in specific types of data and process messages that contain this data without knowing where it originated. This method is especially advantageous for high-speed and cyclic data communication. Both communication models are depicted in Figure 5.9.



(a) Client-Server (b) PubSub

Figure 5.9: OPC UA communication models

For client-server communication, the OPC UA Connection Protocol (UACP) handles the data transport. This abstract protocol establishes a full duplex channel between a client and a server and is designed to work with a secure channel implemented by a layer higher in the OPC UA stack. Implementations of UACP can be based on TCP/IP, HTTPS, and WebSockets [125].

A PubSub connection defines the used protocol and the network address for sending or receiving messages. In the case of a message broker as Message Oriented Middleware, the address represents a TCP/IP connection to the broker endpoint. UDP is used along with an IP multicast group as Message Oriented Middleware. When using a broker as Message Oriented Middleware, the standard application layer protocols Advanced Message Queuing Protocol (AMQP) and Message Queue Telemetry Transport (MQTT) can be used [126].

Security

OPC UA is designed to be deployed in diverse operational environments with varying assumptions about threats and accessibility. Therefore, it provides a flexible set of security mechanisms. The threats covered by the OPC UA security mechanisms are Denial of Service, Eavesdropping, Message Spoofing, Message Alteration, Message Replay, Malformed Messages, Server Profiling, System Hijacking, Rogue Server, Compromising User Credentials, and Repudiation. The OPC UA security architecture allows implementing of the required security features within the OPA UA application architecture. The security objectives are addressed at different levels and can be applied for the communication patterns client-server and PubSub [127].

AddressSpace Model

The OPC UA AddressSpace provides a standard way for servers to represent objects to clients. It defines objects in terms of variables and methods and allows relationships to other objects as illustrated in Figure 5.10 [128].

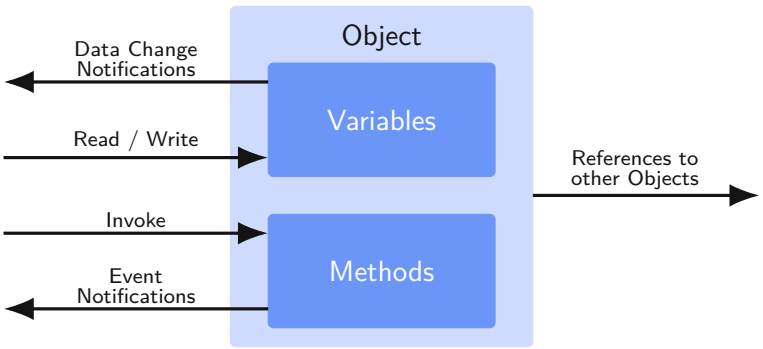


Figure 5.10: OPC UA object

OPC UA Objects consist of nodes within an AddressSpace connected by references. Different classes of nodes convey different semantics. Each node in the AddressSpace is an instance of one of eight defines node classes shown in Figure 5.11.

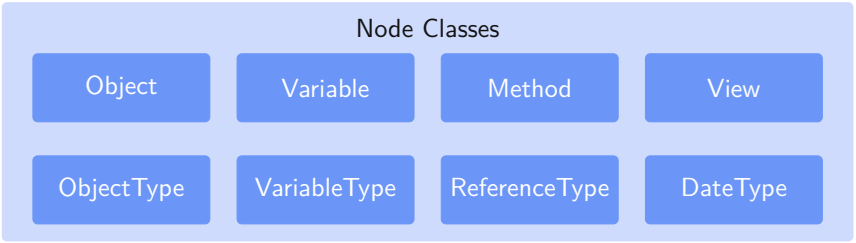


Figure 5.11: OPC UA node classes

Information Model

An OPC UA Information Model describes standardized nodes of a server's AddressSpace as entry points to server-specific nodes, including types and instances. Therefore, it defines the basic data structure of an empty OPC UA Server. Besides standard object types, variable types, and objects with their variables, OPC UA also defines standard methods, views, and reference and data types [129].

This information model must be extended by either an existing companion specification or a machine-specific model for use in a specific application.

Discovery

OPC UA defines services and processes for applications to find other applications on the network and then discover how to connect to them. Discoverable applications are generally servers. However, clients can support reverse connections [130].

A client has several choices for finding servers and discover applications:

- Simple discovery using out-of-band mechanism
- Local discovery
- Subnet discovery
- Global discovery

Every server has one or more discovery URLs that allow access to its endpoints. With a **simple discovery**, a client obtains the discovery URL for the server through some out-of-band mechanism like a web page, graphical user interface, or other mechanisms. Once the discovery URL for the server is known, it reads the endpoint descriptions using standardized services.

If the server's discovery URL is unknown, the client can use a **local discovery** by looking for a Local Discovery Server (LDS) on a host and constructing a discovery URL using well-known addresses. Once the LDS is known, the client can request a list of servers registered on this LDS.

When a client does not know which hosts have servers, it can use a **subnet discovery** by looking for an LDS with multicast extension (ME) on its local host and requesting a list of discovery URLs for servers and discovery servers on the subnet. An LDS-ME server implementation is based upon Multicast DNS (mDNS), which provides the infrastructure for announcing the servers on the multicast network.

In contrast to the previously mentioned discoveries, **global discovery** uses a Global Discovery Server (GDS) whose address space represents OPC UA servers. This discovery allows clients to search for servers but not for the server information. Compared to the LDS, where the host names can be discovered in the local network, a GDS provides an authoritative source for servers that administrators have verified in large systems with multiple servers on multiple subnets.

5.1.3 OPC UA Safety

The specification OPC UA Safety (OPC 10000-15: UA Part 15: Safety) describes the required services and protocols for exchanging safety data using OPC UA mechanisms. It extends OPC UA to fulfill the requirements of functional safety as defined in the IEC 61508 and IEC 61784-3 series of standards. OPC UA Safety specifies a Safety Communication Layer (SCL) allowing safety-related devices to use the services of OPC UA to safely exchange safety-related data suitable for a safety integrity level up to SIL4 according to IEC 61508 and a performance level PL e according to ISO 13849-1 [131].

Architecture

The communication structure of OPC UA Safety is based on an additional safety transmission protocol on top of the standard transmission system of OPC UA. The architecture, illustrated in Figure 5.12, is divided into three parts: the Safety Application Layer, the OPC UA Safety Layer, and the OPC UA Layer.

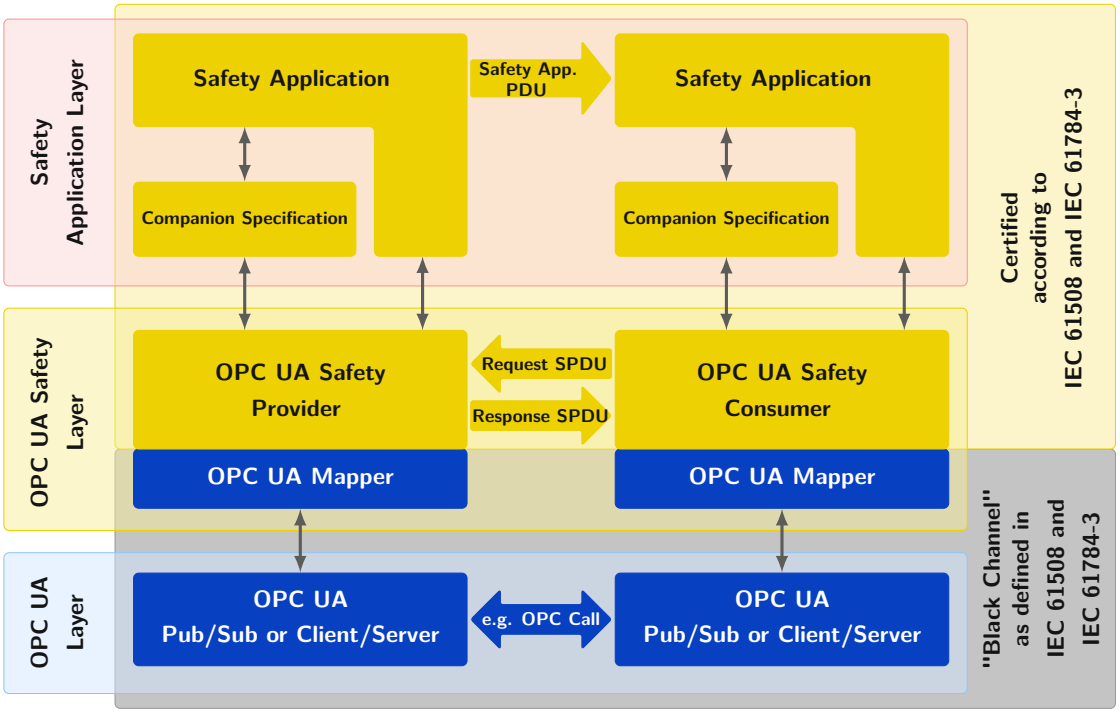


Figure 5.12: OPC UA Safety layer architecture

The **Safety Application Layer** is expected to be designed and implemented according IEC 61508 and is directly or via a machine-specific companion specifications connected to the SafetyProvider or SafetyConsumer of the layer below.

The **OPC UA Safety Layer** defines the two services SafetyProvider and SafetyConsumer as basic building blocks that form the SCL. Safety data is transmitted using unidirectional point-to-point communication. Each unidirectional data flow internally communicates in both directions with a request-response pattern, using request and response Safety Protocol Data Units (SPDUs). The parts of the safety layer specific for the OPC UA communication service in use, i.e., PubSub or client-server, are implemented in the OPC UA Mapper. Therefore, the remaining parts of the safety layer can be implemented independently of the OPC UA service being used.

For the **OPC UA Layer**, both OPC UA communication patterns, either PubSub or client-server, can be used. When using a client-server pattern, the SafetyProvider is implemented using an OPC UA server providing a method. The SafetyConsumer is implemented using an OPC UA client calling that method. When using PubSub, the SafetyProvider publishes the ResponseSPDU and subscribes to the RequestSPDU, while the SafetyConsumer publishes the RequestSPDU and subscribes to the ResponseSPDU.

This modular architecture provides a black channel, as described in IEC 61508, for the safety application without specifying its physical medium underneath. Thereby, it provides End-to-End safety communication, meaning that functional safety data is transported between two safety endpoint devices across a standard network that is not functionally safety compliant. Additionally, it supports dynamic systems where safety communication partners may change during operation.

Communication Links

Generally, three different types of communication links between safety applications can be applied using OPC UA Safety: unidirectional, bidirectional, and multicast. There are two basic building blocks for all these types of communication: a SafetyProvider, which implements the data source of a unidirectional safety link, and a SafetyConsumer, which implements the data sink of a unidirectional safety link. The connection between SafetyProvider and SafetyConsumer can be established and terminated during runtime, allowing different SafetyConsumers to connect to the same SafetyProvider at different times.

The most basic type of communication depicted in Figure 5.13 is **unidirectional communication**. A safety application on one device (Controller A) sends data to another safety application on another device (Controller B) using a SafetyProvider on Controller A and a SafetyConsumer on Controller B.

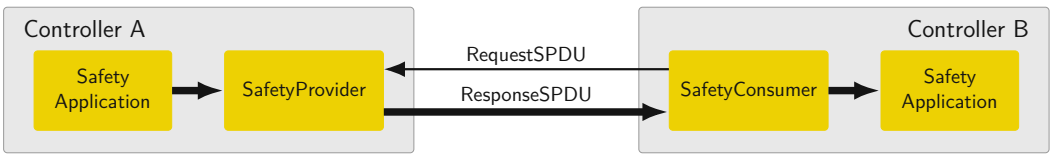


Figure 5.13: Unidirectional communication

Bidirectional communication is realized using two OPC UA Safety connections as illustrated in Figure 5.14. Thereby, the exchange of data in both directions is accomplished by placing a SafetyProvider and a SafetyConsumer on each controller.

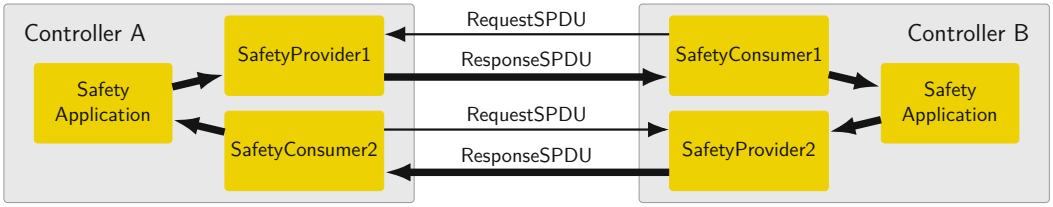


Figure 5.14: Bidirectional communication

Multicast communication is defined as sending the same set of data from one device (Controller A) to several other devices (Controller B1, B2,. . . , BN) simultaneously. It is accomplished by placing multiple SafetyProviders on Controller A and one SafetyConsumer on each of the Controllers B1, B2, ..., BN as shown in Figure 5.15.

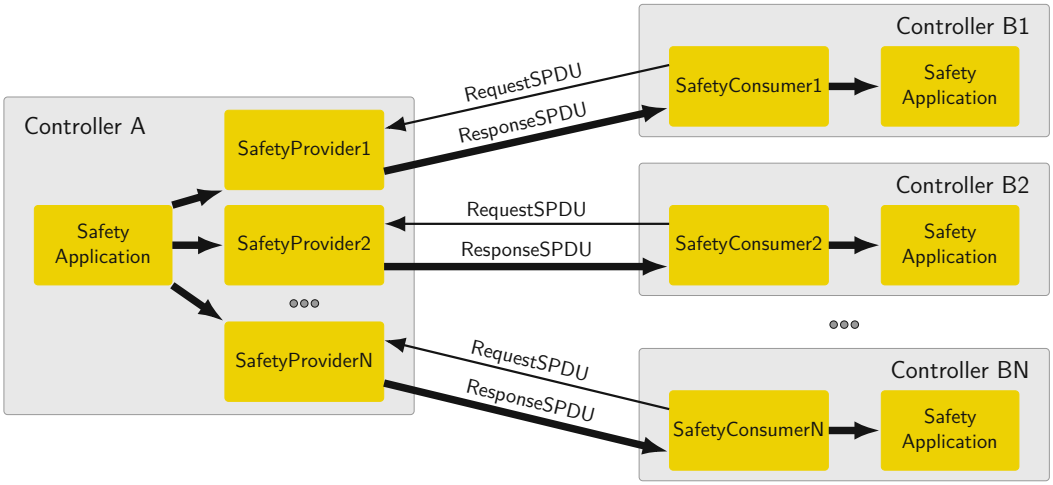


Figure 5.15: Multicast communication

Cyclic and Acyclic Safety Communication

Safety functions must react timely to external events using cyclic safety communication. In this case, the SafetyConsumer is executed cyclically, and the time between two consecutive executions is safely bounded. Some safety functions, such as the transfer of safe configuration data at startup, do not have to react to external events. In this case, it is not required to execute the SafetyConsumer cyclically.

Information Models

For use in safety-related applications, OPC UA Safety extends the information model by identifiers, types, and structure of the objects and methods that are used to implement the OPC UA mappers. These extensions support the safe exchange of SPDUs at runtime, identifying SafetyConsumers and SafetyProviders at online browsing and exporting and transferring in a standardized format for offline engineering.

OPC UA Safety defines several objects. SafetyACSet is the entry point for browsing a component implementing OPC UA Safety and finding its SafetyProviders and SafetyConsumers. SafetyPDU is used by the SafetyProvider to subscribe to the RequestSPDU and to publish the ResponseSPDU. Furthermore, for safety parameters, there are the objects SafetyProviderParameters and SafetyConsumerParameters. Besides the objects, OPC UA Safety also defines methods, namely ReadSafetyData to read SafetyData from the SafetyProvider and ReadSafetyDiagnostics serving as a diagnostic interface.

Safety data is modeled in two data types RequestSPDUDataType depicted in Figure 5.16 and ResponseSPDUDataType depicted in Figure 5.17.

Figure 5.16: Request SPDU

Figure 5.17: Response SPDU

In order to allow multiple versions of OPC UA Safety at the same time on one device, different identifiers are used. Therefore, the same SafetyProvider can be accessed by SafetyConsumers of different versions.

Interfaces

The main function of the OPC UA Safety layer services is the state machine, which handles the safety protocol. For the interaction with the safety application, SafetyProvider, and SafetyConsumer the following interfaces are used:

- Safety Application Program Interface (SAPI)
- Safety Parameter Interface (SPI)
- Diagnostics Interface (DI)
- OPC UA Platform Interface (OPC UA PI)

The SPI is used to commission the settings of safety parameters such as IDs or the timeout value. During runtime, the SAPI is accessed by the safety application for exchanging

safety data, the non-safety related DI for troubleshooting the safety communication, and the OPC UA PI connects the SCL to the non-safe OPC UA stack. The state machines of OPC UA Safety are independent of the actual OPC UA services used for data transmission. This separation is accomplished by introducing a so-called OPC UA Mapper, serving as an interface between the SCL and the OPC UA stack. The mapper can either make use of OPC UA client-server communication and remote method invocation or the publishing of and subscribing to remote variables. Figure 5.18 depicts the OPC UA Safety layer with its interfaces.

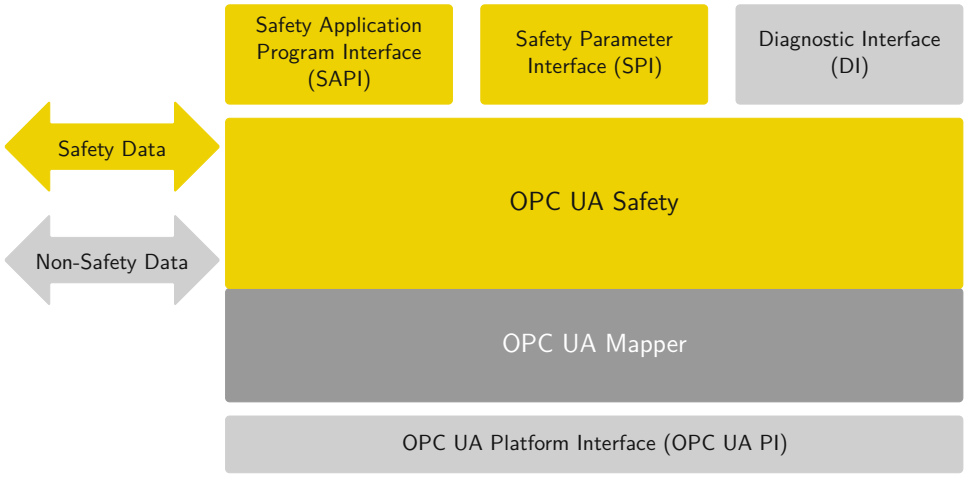


Figure 5.18: Safety communication interfaces

Application Example

Safety communication based on OPC UA Safety utilizes the building blocks SafetyProvider and SafetyConsumer to establish safe communication links. Figure 5.19 shows a simple application example realizing a safety function.

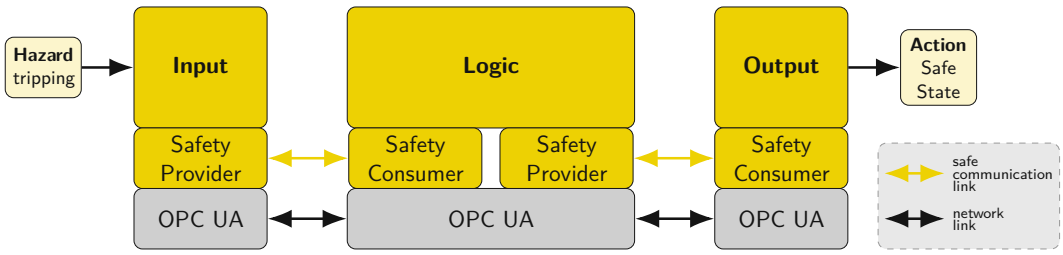


Figure 5.19: OPC UA Safety example

The configuration of this example involves the information models and safety applications of the three devices and the two communication links with their respective SafetyProvider and SafetyConsumer.

5.2 Safety Communication Platform

For an Industry 4.0 (I4.0) conform communication platform, the used technologies should be standardized, vendor-neutral, support interoperability, and be industrially mature. Based on the functional requirements FR-03 ... FR-12, a communication platform consisting of three base technologies is proposed. These three candidates, which are gaining acceptance in the industry, were chosen to serve as a foundation for a safe communication platform: TSN, OPC UA, and OPC UA Part 15: Safety. Figure 5.20 illustrates these three communication technologies combined as a basis for a vendor-neutral and standardized communication platform.

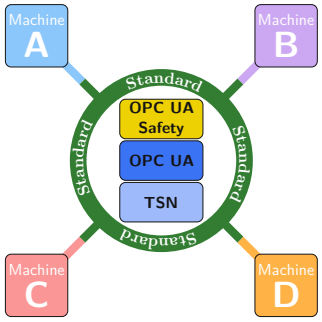


Figure 5.20: Safety communication technologies

5.2.1 Safety Communication Stack Architecture

The architecture of a safety communication stack must fulfill the requirements stated in Section 4.4 to serve as a foundation for the envisioned Flexible Safety System (FSS). Forging a communication platform that supports safe and non-safe transmission of messages requires an integrated architecture that combines both types of data transfer. The proposed architecture uses TSN for deterministic data transport, OPC UA for unified communication, and OPC UA Safety for exchanging safety-related messages. Various combinations of the three base technologies are possible for a safety architecture that combines deterministic transport, safety features, and an integration middleware. Figure 5.21 shows four such variants using TSN, OPC UA, and OPC UA Safety [18].

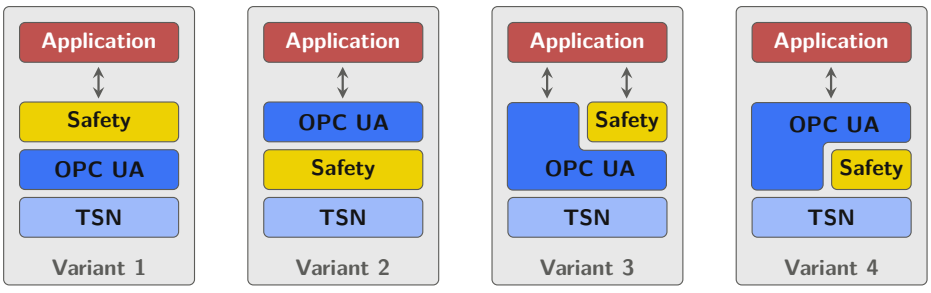


Figure 5.21: Safety communication stack architecture variants

Variant 1 assumes that the application sends and receives only safety-related messages and uses OPC UA as its black channel, while in variant 2, the safety protocol ensures a safe connection between OPC UA nodes. Both variants 1 and 2 allow only safety-related data exchange between safety devices. A best-effort channel for configuration data is not available. Variant 3 is structured in a way that the application is able to transmit safety-related and non-safety-related data between devices. Only safety-related data is transported via the safety protocol with OPC UA as the black channel. Variant 4 organizes the technologies in a way that the devices act as OPC UA and safety devices. The OPC UA stack decides if data may use the safety channel. The application has no access to the safety stack directly [18].

According to EN ISO 13849-1 and IEC 62061, certification covers the path from the safety protocol stack to the physical inputs/outputs. Assuming that the OPC UA stack is not certified, the application requires direct access to the safety stack. Thus, variants 2 and 4 can be excluded. Variant 3 has the advantage over variant 1 in that it can access both data channels with safety and non-safety messages. Evidently, it is the best candidate for further consideration, as shown in Table 5.4.

	Variant			
	1	2	3	4
Direct Safety Data Access	+	-	+	-
Direct Non-Safety Data Access	-	-	+	+
Safe and Non-Safe Data Transfer	-	-	+	+

Table 5.4: Communication stack variants comparison

The envisioned communication platform consists of three layers: deterministic data transfer using TSN, semantic information modeling using OPC UA, and safety-relevant process data exchange using OPC UA Safety. Such an architecture also entails configuration on all three layers. Therefore, an engineering tool is necessary to handle all the configuration tasks easily, quickly, and safely.

5.2.2 TSN Configuration

From the three available TSN network management models defined in IEEE 802.1Qcc, only the fully centralized model can be used because it is the only model that covers all devices in the network, including the end stations. This model is, therefore, suitable for highly critical applications, such as industrial control, in which complex timing requirements can be configured needed by the industrial traffic types defined in IEC/IEEE 60802. Consequently, a CNC and a CUC must be instantiated and configured accordingly.

Communication with the network devices utilizes the NETCONF. It is the most modern network management protocol with some advantages, such as the automatic discovery of YANG modules on the devices. Furthermore, the TSN Task Group continuously publishes new YANG models.

All TSN devices need a global clock synchronization using PTP as required by IEC/IEEE 60802, which has to be configured for the network.

Additionally, for each safety-relevant communication link between a safety provider and a safety consumer, either between safety sensor and safety logic or safety logic and safety actuator, a TSN stream with its relevant parameters has to be established meeting the following requirements:

- calculation of TSN stream route from talker to listener
- TSN schedules for all TAS on the related egress ports of the bridges (IEEE 802.1Qbv)
- TSN stream parameters according to isochronous traffic class (IEC/IEEE 60802)

5.2.3 OPC UA Configuration

OPC UA offers two communication models from which the PubSub model is chosen using a UDP connection with an IP multicast group as message-oriented middleware. This combination offers the best method of transmitting safety-relevant process data with the least overhead and, therefore, with minimal delay.

After a TSN stream between a safety provider and a safety consumer has been established, an OPC UA publisher on the safety provider and an OPC UA subscriber on the safety consumer have to be instantiated, configured, and executed in order to provide the basis for the OPC UA mapper as an interface to the safety protocol. This procedure has to be done for each safety communication link. The following parameters have to be set:

- multicast IP address
- UDP port number
- VLAN ID
- publisherID
- publishingInterval
- writerGroupID

5.2.4 OPC UA Safety Configuration

Once an OPC UA PubSub communication link with its underlying TSN stream is successfully established, a safety provider on the sending side and a safety consumer on the receiving side of the link have to be instantiated, parametrized, and started. The following parameters have to be set on provider and receiver defined in [131]:

- safetyProviderID
- safetyConsumerID
- safetyData structure
- safetyProviderLevel (SIL)
- safetyConsumerTimeOut

5.2.5 Configuration Deployment

Deploying safety communication links requires complex simultaneous chronological configuration sequences on three levels. Therefore, an automatic deployment tool is essential for achieving rapid and safe configuration changes. Before establishing safety communication links, an initial network-related setup has to be carried out to enable global clock synchronization. This initial setup has to be repeated every time a network device is added or removed [132].

5.3 Safety Configuration Tool

A safety communication system that is based on three base technologies requires a sophisticated configuration tool that helps and assists the operator in managing and configuring the system and that fulfills the requirements on tools for safety-related systems as described in Section 2.9 and in Section 4.4.

Due to the complexity of the setup, an effective and comprehensive toolchain for establishing and removing safety connections must be provided. This toolchain has to cover the following aspects:

- Discovery (network devices, network topology, device safety features)
- Deployment (TSN, OPC UA, OPC UA Safety)
- Knowledge representation (information about the safety system and its environment)
- Logging (keeping a logbook or journal of all changes and events that occur)
- Visualization (user interface for monitoring, configuring, and adapting)
- Configuration generators (automatic generation of configuration)
- M2M-Control Interface (for remote control from an orchestration system)

The configuration toolchain is supposed to operate on-line. Every change in the system shall be detected automatically, and if applicable, automatic action (e.g., configuration change, notification) shall be performed. All changes and events within the system shall be written in a logbook for monitoring purposes and future analysis. Additionally, a well-defined Application Programming Interface (API) to the knowledge representation shall allow additional services in the future. Such services could include a compliance check that ensures compliance with specific laws and regulations or a conformance check that verifies the conformance to an industry standard or technical specification.

CHAPTER 6

Flexible Safety System Design

> MACHINES ARE INCREASINGLY CONNECTED WITH INDUSTRIAL PROCESSES AND PERFORM TASKS IN COOPERATION WITH HUMANS. EARLY AUTOMATION AND MASS PRODUCTION HAVE BROUGHT MANY CHALLENGES AND, MOVING FORWARD, TECHNOLOGICAL SOLUTIONS OF THE NEW INDUSTRIAL REVOLUTION WILL CREATE NEW CHALLENGES FOR INDUSTRIAL SAFETY. [133]
>
> – Li Yong, Director General, UNIDO, 2019

According to the Flexible Safety System Development Life Cycle (FSS-DLC), depicted in Figure 3.2, the technology integration phase is followed by the system design phase, which builds on the results of the previous phase. This chapter describes the system design in detail and constitutes the main contribution of this research. The results from the use case analysis, requirements determination, the abstract safety system model, and the technology integration from the two previous chapters are transformed into a modular system design that covers the system architecture, safety devices, interfaces, communication, configuration, and toolchain.

The system foundation is presented after describing the overall system concept, including safety devices, safety-related communication, and the respective configuration. Knowledge representation, interfaces, and system services complete the system design, addressing visualization, discovery, deployment, and automatic configuration generation.

The Flexible Safety System (FSS) design describes the system in sufficient detail so that developers and engineers can create a software architecture, a prototype implementation, and subsequently develop the system in the development phase.

6.1 Overall Concept

In order to address the needs of a machine, production line, or smart factory regarding functional safety, a solution is proposed that provides a flexible safety system based on vendor-neutral technologies to ensure interoperability between devices of various manufacturers. The solution also considers assistance for system configuration. Figure 6.1 illustrates the overall system concept overview for an FSS using the base technologies Time-Sensitive Networking (TSN), OPC Unified Architecture (OPC UA), and OPC UA Safety.

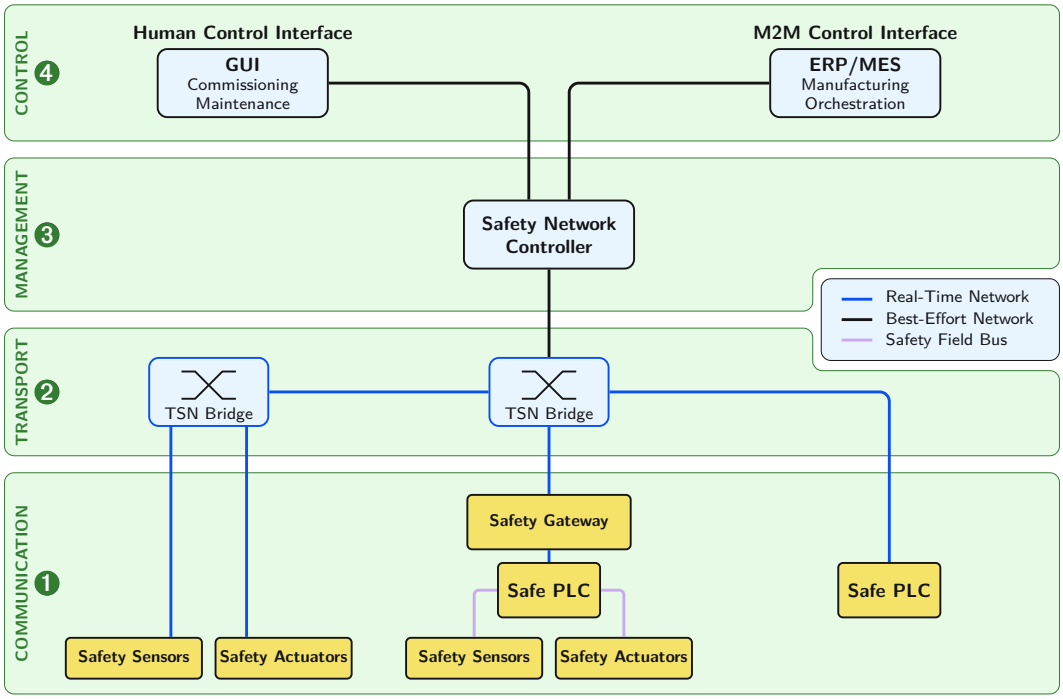


Figure 6.1: FSS concept overview

At the core of the proposed safety system, there are four main segments:

1. **Safety Device Communication – based on OPC UA Safety**
   serving as safe communication platform between different safety devices or domains
2. **Deterministic Transport of Data – based on TSN**
   providing deterministic real-time data transport for reliable data transmission
3. **Safety Network Management**
   a Safety Network Controller (SNC) manages all tasks to configure and monitor the TSN network, OPC UA information models, and OPC UA Safety communication
4. **Control Interfaces**
   providing a graphical user interface (GUI) for commissioning, maintenance, and monitoring, and an M2M interface for manufacturing orchestration

6.1.1 System Components

An FSS consists of several system components that result from subdividing the four main segments: safety communication, data transport, safety network management, and control interfaces. These components and an additional configurations component, a Knowledge-Based System (KBS), are illustrated in Figure 6.2.

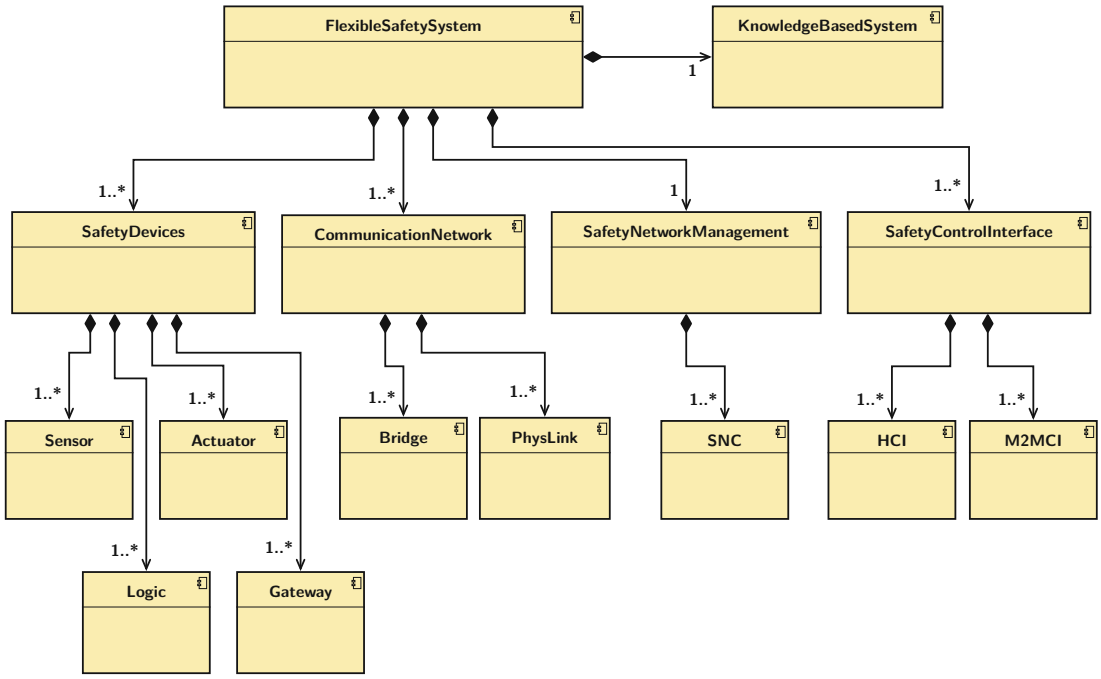


Figure 6.2: FSS components

Safety devices can be safety sensors, safety logic, safety actuators, or safety gateways. A safety gateway bridges safety signals between safety domains by exposing certain safety inputs and outputs from a legacy system into an FSS. This approach can be compared to a discrete wiring solution where terminals in the control cabinet are prepared for external components. When the external part is not used, the terminals are short-circuited with a jumper wire. This safety connectivity on a software level brings maximum flexibility and enables the operator to change the safety configuration with no hardware or wiring change, even during operation.

The communication network comprises TSN bridges and the physical network links in between. These system components provide the basis for combining deterministic and best-effort traffic communication. Deterministic communication is required for the transmission of safety process data, whereas best-effort traffic is used for configuration, maintenance, and monitoring. Furthermore, TSN is a vendor-neutral technology and supports configuration via standardized Network Configuration Protocol (NETCONF) and YANG models.

Network management, including configuration, safety and network device monitoring, and logging of all safety and network devices, is accomplished using a safety network controller that discovers the safety devices, generates a configuration, and deploys it after acknowledgment from the operator into the system.

Two control interfaces are intended for the external control of the system: a Human-Control Interface (HCI) and a Machine-to-Machine-Control Interface (M2MCI). The HCI provides the resources, models, and applications for a Graphical User Interface (GUI). The M2MCI is foreseen for direct access from an Manufacturing Execution System (MES) or an Enterprise Resource Planning (ERP) system to the safety configuration, enabling the reconfiguration of the safety system depending on the product or production line configuration. Additionally, monitoring current and historical events (e.g., dashboard) assists and supports the safety engineer.

A KBS holds all information, safety and network configuration data, safety regulations, and machine models. First and foremost, this includes information about the currently available machinery, focusing on safety-relevant components like safety sensors and actuators and relevant attributes such as type, position, dimensions, kinematics, and weight. Additional information like pre-defined safety configurations, roles of personnel (e.g., safety engineer), their rights, and applicable laws and standards may also be included. A reasoner may be included to infer new knowledge from the information stored in the knowledge base. This additional knowledge can support the FSS in generating new safety configurations and provide valuable insights for the operators.

6.1.2 Communication Relations

Within the system, different data traffic types for safe data (safety process data) and non-safe data (configuration and monitoring) are used, which are illustrated in Figure 6.3.

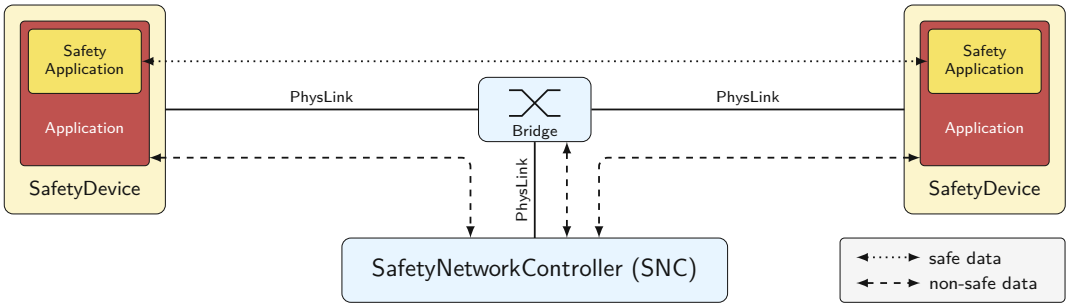


Figure 6.3: FSS communication relations

IEC/IEEE 60802 defines industrial automation traffic types, listed in Table 5.3, with their associated characteristics. All communication relations in the system are assigned to one of these traffic types to allow concurrent transmission of data according to the required characteristics.

The safety-related data exchange between safety devices requires configuration on those devices and the network infrastructure in between. Therefore, a data channel with the best-effort traffic type is used. Once the configuration task is completed, the safety process data is transferred using the same network infrastructure with the isochronous traffic type to ensure the deterministic transmission of safety-relevant messages.

6.1.3 Communication Stack

In order to allow safety devices to communicate concurrently on the same communication platform using several traffic types, a communication stack is required that supports the three base technologies TSN, OPC UA, and OPC UA Safety as depicted in Figure 6.4.

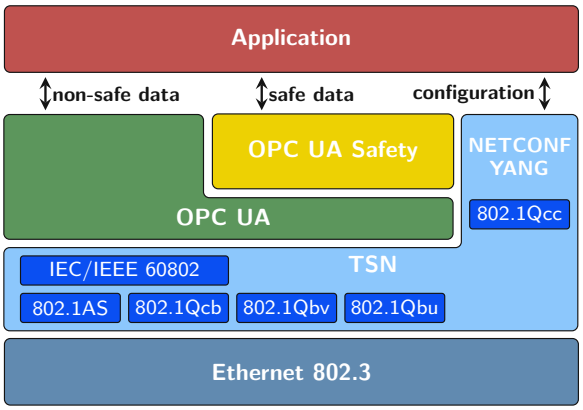


Figure 6.4: Communication stack [18]

Based on the IEEE 802.3 Ethernet standards, the TSN Profile for Industrial Automation IEC/IEEE 60802 with the TSN IEEE standards 802.1AS (synchronization), 802.1Qcb (frame replication and elimination for reliability), 802.1Qbv (enhancements for scheduled traffic), and 802.1Qbu (frame preemption) are applied to realize the traffic types specified in IEC/IEEE 60802. Additionally, the TSN standard 802.1Qcc is utilized for the configuration of TSN specific parameters using the NETCONF protocol with the associated YANG models.

On top of TSN, the communication platform OPC UA is placed, responsible for information modeling and providing the communication patterns Client/Server and PubSub for non-safe data exchange.

Safe data transfer is provided by OPC UA Safety, which is attached to the underlying OPC UA stack. It is responsible for establishing a channel for safety-related message exchange using the black channel principle [18]. Safety messages, therefore, use the OPC UA Safety stack, which is connected via a mapper with the OPC UA stack and uses PubSub as a communication pattern as described in [131].

6.1.4 System Configuration

The realization of a safety function using the proposed communication stack entails configuration on several devices (sensor, logic, actuators, and network devices) and on several layers (TSN, OPC UA, and OPC UA Safety). A safety function requires safe communication from sensor to logic and from logic to actuator. Establishing such a safe communication link involves configuration on three layers. First, TSN streams are created by configuring TSN Talkers and TSN Listeners. Second, a bidirectional OPC UA PubSub connection is set up by configuring OPC UA publishers and OPC UA subscribers. Third, an OPC UA Safety communication link is established by configuring an OPC UA Safety provider and an OPC UA Safety consumer. Figure 6.5 illustrates a safety function with its safety devices, the communication between, and the configuration communication relations to the Safety Network Controller (SNC).

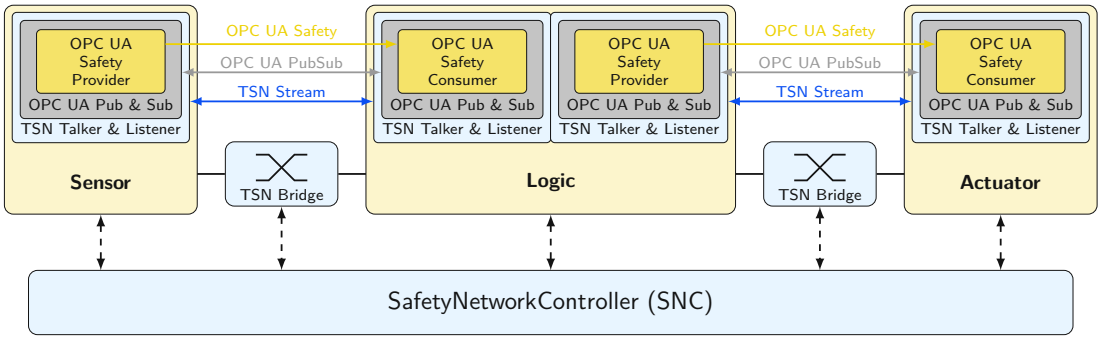


Figure 6.5: Safety function configuration

Establishing a safe communication link requires a number of configuration tasks on multiple devices and several layers and the execution of these tasks in a specific order and with error handling accordingly. When deploying the configuration of a safety function, the SNC has to analyze all occurring errors and create a strategy for a rollback. This aspect is especially important due to the fact that the configuration might be deployed simultaneously on several devices in order to minimize the time needed for deployment. Only after the error-free execution of all configuration tasks the safety function can be put into operation.

During risk evaluation, as part of the risk assessment, a set of safety functions has to be defined for all identified hazards to reduce risk adequately. In an FSS, the SNC manages all resources for safety-related measures to reduce risks. These resources include all elements of the safety functions (sensors, logic, actuators, and communication). IEC 61511 defines the term Safety Instrumented System (SIS), which is composed of and performs one or several Safety Instrumented Functions (SIFs). This set of SIFs has to be seen as a whole, meaning that a change in one SIF affects the whole SIS and, therefore, its certification. For this reason, the configuration of a SIS must be an atomic composition of all configurations of all included SISs. Figure 6.6 illustrates a SIS with its SIFs and the elements which need configuration.

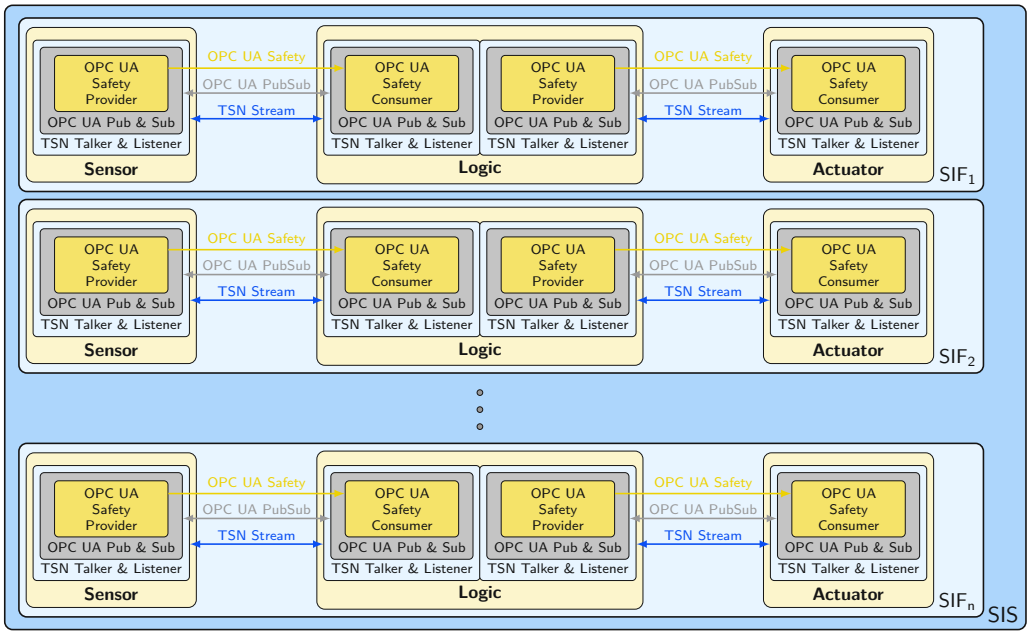


Figure 6.6: Safety system configuration

Combining the communication stack described in Section 6.1.3 with the safety system configuration, safety functions can be implemented in a production system as shown exemplarily in Figure 6.7. In this example, the safety links between the four safety domains can be easily reconfigured without the need for changes inside the domains.

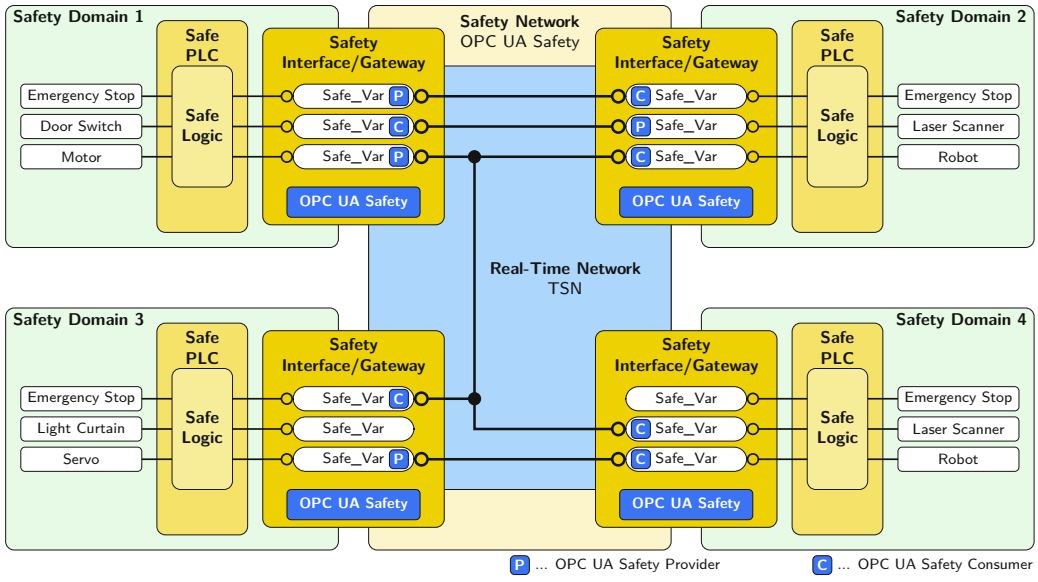


Figure 6.7: Safety connections example

The safe Programmable Logic Controllers (PLCs) in each safety domain exposes certain safety variables through an OPC UA Safety interface directly or using an OPC UA Safety gateway. These variables can be either inputs or outputs, which are mapped to OPC UA Safety provider or consumer, respectively. This structure enables the operator of a flexible production cell to change safety functions by deploying different safety configurations affecting only the TSN network, the OPC UA PubSub connections, and the OPC UA Safety links with their OPC UA Safety provider and consumer.

Removing and adding individual safety-relevant devices within a production cell, such as safety sensors, safety logic, safety actuators, safety gateways, or even a whole safety domain, may become necessary depending on the production process or process step. By use of an FSS with its flexible and vendor-neutral communication based on TSN, OPC UA, and OPC UA Safety and assisted by the SNC, the physical reconfiguration of a production cell, which also entails the reconfiguration of the safety system is made possible, even for operators without functional safety experience. Furthermore, the SNC monitors the status of network and safety devices constantly, which allows it to automatically react to changes in the system, such as network link failure or a newly discovered device. Various sources can trigger the reconfiguration of the safety system. As already mentioned, this could be due to a failure or outage but also due to additional devices or network links. Another reason for reconfiguration can originate from an MES or ERP system. Therefore, an FSS, in particular the SNC, must be designed to accept requests from other systems and transform these into safe configurations.

This section gave a general overview of the structure, communication, and configuration of an FSS. The following sections describe the four main segments of an FSS in detail.

6.2 Safety Device Communication

At the innermost core of an FSS is the communication between two safety-relevant devices, generally from sensor to logic and from logic to actuator. Since this communication contains safety-relevant data, additional techniques and measures must be applied to protect it. Based on Ethernet, two technologies are applied to address the requirements for interoperable communication on a technical, syntactic, and semantic level between safety-related devices, namely OPC UA and OPC UA Safety.

OPC UA serves as a communication platform that addresses the requirement for interoperable, vendor-independent exchange of information. Additionally, it provides the required discovery features and modeling capabilities. For the exchange of safety-related information using the black channel principle according to IEC 61508 and IEC 61784-3, a safety protocol is specified in OPC UA Part 15: Safety. This specification defines safety-related data exchange between a SafetyProvider and a SafetyConsumer using an OPC UA Mapper as an interface to the underlying OPC UA communication platform. Three types of communication links, unidirectional, bidirectional, and multicast communication, are available to implement various scenarios. Initially, unidirectional communication will be used. If useful and applicable, other types could be applied as well.

The connection between SafetyProvider and SafetyConsumer can be established and terminated during runtime, allowing the reconfiguration of safety functions during operation. Figure 6.8 illustrates the communication relations on several layers that are necessary to transmit safety-related data between two safety applications using OPC UA Safety with unidirectional communication.

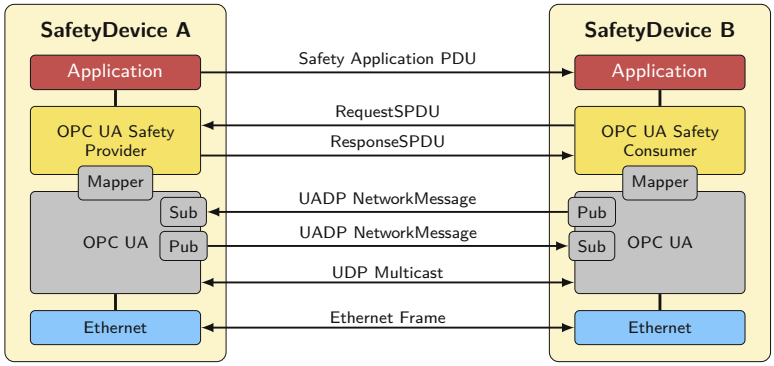


Figure 6.8: Safety device communication

The connection establishment starts with creating a bidirectional OPC UA PubSub connection using OPC UA UDP, a simple UDP-based protocol to transport UADP NetworkMessages. To accomplish bidirectional data transfer using PubSub, the same UDP multicast group has to be configured on both devices. Then, on both devices, a publisher and a subscriber are instantiated and configured for bidirectional data transfer accordingly. Additionally, the frequency at which the messages are sent has to be configured on the publishers. That frequency has to be adjusted to the required Safety Function Response Time (SFRT) of the safety function. Once the OPC UA multicast connection using UDP is established, a SafetyProvider on one device and a SafetyConsumer on the other device is instantiated and configured using an OPC UA Mapper as an interface to the underlying OPC UA communication platform. Here, the SafetyConsumer needs to know the frequency at which it requests the SPDU from the SafetyProvider. Again, that frequency must be adjusted to the safety function's required SFRT.

Now, after activating the publishers on both devices, the safety application PDU written by the provider application is transferred to the consumer application. If a safety-related message does not arrive in time, which depends on the required SFRT, the consumer application is notified by the SafetyConsumer and consequently can take action. On a safety actuator, this action could be to bring the attached drive into its safe state. At this stage, a safe communication link has been established for two applications to exchange safety-related messages. However, this communication is based on standard Ethernet, which does not guarantee correct data transmission on the physical layer. Due to data traffic congestion on the bridges or insufficient bandwidth, this circumstance can lead to long transmission delays or packet loss. Therefore, the next segment introduces deterministic data transport to ensure the reliable operation of the safety system.

6.3 Deterministic Transport of Data

Reliability is one of the essential characteristics of an industrial production facility, which, of course, also applies to its safety system. If the safety communication is based on a not reliable or non-deterministic data transport, the production facility has to go in its safe state whenever the safety-related communication is disturbed. Of course, this is not acceptable for machinery which should operate continuously. Therefore, deterministic data transport is a prerequisite for a safe and stable operation of safety communication based on Ethernet. It guarantees low latency, minimal jitter, and minimal packet loss within defined boundaries. The vendor-neutral technology TSN fulfills this requirement and also addresses interoperability on a lower level between devices of various manufacturers. The TSN profile IEC/IEEE 60802 selects features, options, configurations, defaults, protocols, and procedures of bridges and end stations to build industrial automation networks, which typically include control loop operations that are based on three types of application: sensor, controller, and actuator. In order to enable the traffic types defined in IEC/IEEE 60802 and required by an FSS, a minimum set of TSN standards is necessary. These standards, which can be divided into standards used for data transport and standards used for management of TSN streams, are listed in Table 6.1.

	Standard	Description
Transport	IEEE 802.1AS	Is a constrained subset of the Precision Time Protocol (PTP) that provides time synchronization with sub-microsecond precision
	IEEE 802.1Qbv	Defines a Time-Aware Shaper (TAS) providing fixed cycle times needed for control applications
Management	IEEE 802.1AB	Defines the Link Layer Discovery Protocol (LLDP) which is a vendor-neutral link layer protocol used by network devices for advertising their identity, capabilities, and neighbors on a local area network
	IEEE 802.1Qcc	Defines enhancements and performance improvements for the Stream Reservation Protocol (SRP), which provides protocols, procedures, and managed objects for bridges and end stations and is used for remote configuration management based on the Network Configuration Protocol (NETCONF)
	IEEE 802.1Qcp	Implements the YANG data model to provide a framework for status reporting and configuration of equipment

Table 6.1: TSN standards required for FSS

The standards listed in Table 6.1 constitute an absolute minimum required to simultaneously realize time-triggered and best-effort data transfer in one network infrastructure, ensuring the timely transmission of safety-relevant messages and the utilization of the remaining bandwidth for system configuration, network and device monitoring, and other services. Several additional standards can also be applied to improve performance, efficient bandwidth utilization, or increased reliability.

6.3.1 Scheduled Data Transmission

The concurrent transmission of several traffic types on one network requires high-precision time synchronization and traffic scheduling across the whole network. In doing so, each egress port on all network devices is equipped with a Time-Aware Shaper (TAS), allowing the synchronized scheduling of data streams with various traffic types controlled by a Gate Control List (GCL), as depicted in Figure 5.4. For this reason, the network-wide and coordinated preparation of the GCLs is of central importance. The creation of the GCLs requires knowledge about the desired TSN streams and the current network topology. Figure 6.9 illustrates the GCL creation process using a scheduler.

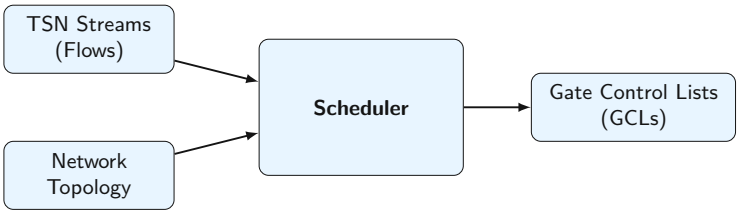


Figure 6.9: GCL creation process

Fundamentally, the scheduler has two inputs. First, a list of all desired TSN streams, including their parameters such as talker, listener, bandwidth, period, and deadline. Second, a graph of the topology of the network containing the nodes with their ports and the links in between. Thereby, the links also have a set of attributes such as type, transmission rate, length, propagation delay, and priority. The scheduler delivers as output the GCLs for all TASs in the network.

The problem of finding a valid schedule for a set of time-triggered streams in an TSN network for a given instance is known to be NP-complete [134]. Furthermore, there may be multiple valid schedules or no schedule at all. Therefore, Stüber et al. summarize in [135] various existing strategies to find an optimal schedule for a given problem, which are divided into exact and heuristic approaches and listed in Table 6.2.

Approach	Strategy
Exact	▪ Integer Linear Programming (ILP) ▪ Satisfiability Modulo Theories (SMT) ▪ Constraint Programming (CP) ▪ Pseudo-Boolean Optimization (PBO)
Heuristic	▪ Greedy Randomized Adaptive Search Procedure (GRASP) ▪ Tabu Search ▪ Simulated Annealing (SA) ▪ Genetic Algorithms (GA) ▪ List Scheduling (LS) ▪ Machine Learning (ML)

Table 6.2: TSN scheduling approaches accoring to Stüber et al. in [135]

The exact approaches compute a schedule or an optimal schedule if one exists or proves the problem instance is infeasible. Heuristic approaches can not guarantee finding an existing schedule. Instead, they try to find reasonably good solutions within a short time, meaning they cannot deduce whether a problem instance is infeasible or if a solution exists. The advantage of the heuristic approach is the required time to find a solution [135].

Calculating the schedules for all TASs of an FSS is a time-critical task due to the possibility of configuration changes during the operation of the production system. For this reason, the heuristic strategy, developed by Raagaard et al. in [136], serves as a basis. Some enhancements for advanced bookkeeping of the scheduled flows keep track of the time slots assigned to the frames of flows, improving the calculation performance significantly. Additionally, mechanisms to efficiently remove flows from the schedule could speed up the recalculation of the schedules in the case of network topology changes. The reconfiguration of TSN schedules at runtime is also addressed by Raagaard et al. in [137], where a configuration agent reacts to changes in the network traffic and reconfigures new flows, allowing to schedule thousands of frames in a matter of seconds. This possibility is essential for the reliability of an FSS and, therefore, for the whole production system, as communication interruptions due to device or link failure can be repaired automatically with short or even no outage.

Another aspect of the scheduled transmission of Ethernet frames using an TAS defined in IEEE 802.1Qbv is that it does not optimally use the available bandwidth due to the use of guard bands, which are necessary to ensure that the Ethernet interface is ready for transmission when a time-critical frame has to be sent. To optimize the transmission of Ethernet frames and mitigate the negative effects from the guard bands, IEEE 802.1Qbu introduces frame preemption, which splits long messages into parts and transfers the high-priority messages in the available time between used time slots. However, this technology will not be used for FSS due to its high complexity.

6.3.2 TSN Network Management

As already stated in the analysis, only the fully centralized model can be used from the three available TSN network management models defined in IEEE 802.1Qcc. Therefore, a Centralized Network Configuration (CNC) and a Centralized User Configuration (CUC) are required to handle all configuration tasks within the TSN network.

The CUC is a logical entity responsible for gathering the application-level requirements from the end stations, the talker, and the listener. From the end station requirements, the requirements for the TSN stream are derived and forwarded to the CNC, another logical entity responsible for stream reservation and deployment across the whole network for all bridges. In an FSS, the functionality of CUC and CNC is part of the safety network management, which will be discussed in the following section.

6.4 Safety Network Management

All safety-related connections needed for a safety configuration are organized and managed in the Safety Network Controller (SNC) using a southbound interface to the transport and safety communication layer and with northbound interfaces to the control layer for human control and Machine-to-Machine (M2M) control.

The structure of an FSS is inspired by a second-order cybernetic system introduced by Heinz von Foerster, which is the cybernetics of observing systems, also called the cybernetics of cybernetics [138] [139]. Second-order cybernetics addresses the concepts of observation, cognition, self-reference, and autonomy and includes the observer in a larger circularity [50]. Figure 6.10 illustrates a generic second-order cybernetic safety system, including an observed system representing the safety control loop and an observing system representing the SNC that controls the safety system and gets feedback from it.

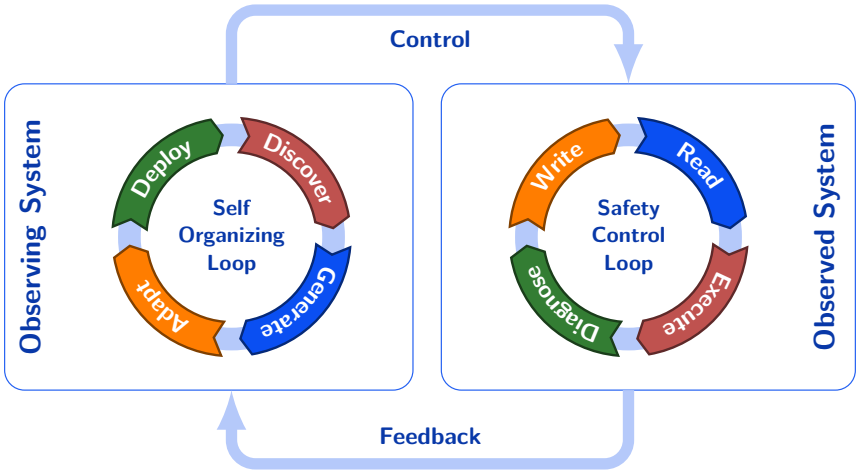


Figure 6.10: Second-order cybernetic safety system

A safety system that ensures the required level of safety within a machine or production line operates circularly and is considered the observed system, as shown in Figure 6.10. It reads the inputs from sensors, executes the safety logic, performs diagnostic tasks, and writes the outputs to the actuators. These four phases build the safety control loop and are executed in a deterministic, timely manner to guarantee safety conditions. Additionally, deterministic communication is a crucial element of a functional safety system that ensures reliable operation and guarantees real-time responses to safety-critical conditions. The configuration, which includes the safety logic and the settings for the deterministic transport of data, is applied once at start-up. Changes can be deployed during run-time or might require a re-start.

In an FSS, the tasks of the observing system, as depicted in Figure 6.10, are realized by the SNC. Thereby, self-organization is defined as the appearance of a structure or pattern without an external agent imposing it. The observing system aims to generate an optimal safety configuration whenever the discovery detects changes in the system or

the operator requests a rearrangement of the production facilities. In order to accomplish that goal, the observing system has to construct its own view of reality. This task is achieved by combining previous knowledge, the data obtained from the discovery, the feedback received from human control, and the actual process configuration. Particularly important in this respect is human interaction for two reasons. First, the generated configuration has to be validated by a safety engineer for legal reasons. Second, human adaptation opens up the possibility to adjust the configuration and align the model of the observing system with the one of the human auditor [50].

In the following, the SNC, which is considered the core element of an FSS, with its assigned tasks and functions, will be discussed in detail. Therefore, the control and feedback loop will be transformed into a configuration procedure and system services.

6.4.1 Configuration Procedure

Besides the administration of safety system configurations as depicted in Figure 6.5, the management of safety-related links entails some additional functionalities. These are the discovery of network devices and topology, including their parameters and services. Furthermore, the generation of safety configurations, the adaptation and verification of automatically generated configurations, a plausibility check for configurations, and the deployment of configurations, including validating the deployed configuration. At the core of the SNC, an information and knowledge-based system provides essential services for data storage, data processing, and modeling capabilities. Figure 6.11 depicts this basic structure and the configuration procedure.

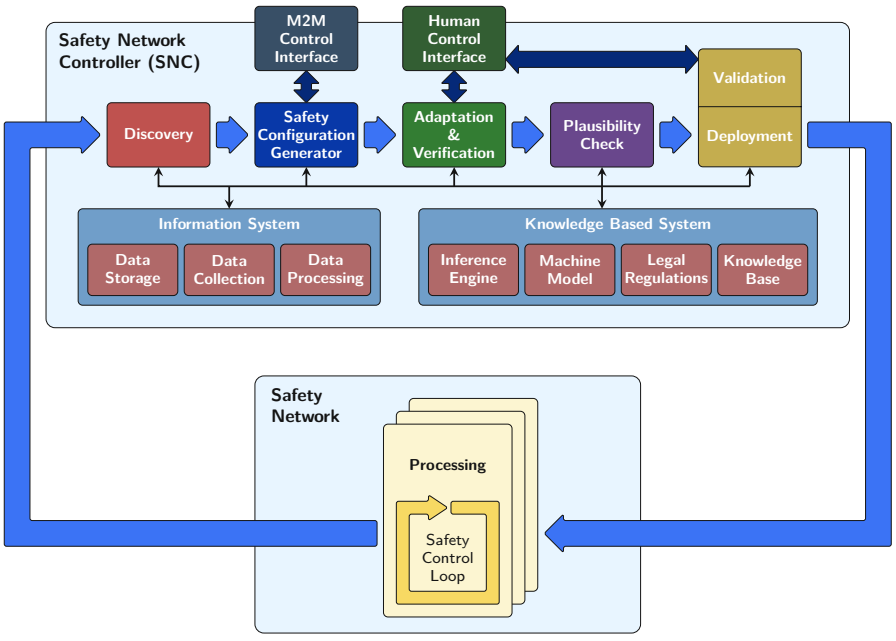


Figure 6.11: FSS configuration procedure

The safety configuration procedure, which is performed by the SNC, can be divided into five consecutive steps. First, the network devices and the network topology are automatically discovered. Second, the information gathered by the discovery, together with the existing knowledge in the KBS, is used in the safety configuration generator to automatically generate a suitable configuration. Third, the generated configuration can be adapted and has to be verified by the safety engineer. Fourth, the verified configuration has to go through the plausibility check to find potential conflicts or inconsistencies before, fifth, being deployed automatically to the system. Once the configuration is successfully deployed, the safety engineer has to validate the running configuration via the Human-Machine Interface (HMI).

Generally, the configuration procedure is designed with regard to the continuous operation of production facilities. Operating mode changes can be applied seamlessly, which is highly relevant for adding or removing mobile safety devices without stopping production.

Device and Network Discovery

The FSS control loop starts with discovering all relevant components, which are the network bridges, network links, and safety devices, to create a current representation of the physical components with their attributes and services. The discovery process can be divided into network topology, semantic information, and safety-relevant attributes. The network discovery uses the Link Layer Discovery Protocol (LLDP) in combination with NETCONF to create a network topology model, including endpoints, bridges, and links. This model will serve as a basis for calculating the required schedules for bridges and endpoints. Semantic information on the safety devices is obtained through discovery mechanisms provided by OPC UA, essentially Local Discovery Server (LDS) and Global Discovery Server (GDS). Attributes relevant to safety devices are gathered by analyzing the information model provided by the endpoint. All the information produced in this step is stored in a knowledge base, and the next step is initiated.

Safety Configuration Generator

When the discovery is completed, the safety configuration generator comes into action and assembles a comprehensive configuration, which includes configuration artifacts for TSN, OPC UA, and OPC UA Safety. The aim is to quickly find a suitable configuration for an existing situation to keep machine downtime as short as possible. This task is accomplished with the aid of a KBS, which provides information about the current composition of the system from the discovery and previous knowledge. This information can contain a machine model, legal regulations, or pre-certified safety configurations. The created configuration template is stored in a knowledge base, and the next step is initiated. The operator can also trigger this step via the M2MCI in case of manual changes in the production process.

Adaptation and Verification

In this step, the operator can view and adapt the generated configuration template from the previous step via a human control interface. Any adjustments or changes to the configuration will again be stored in a knowledge base. It is essential to verify the configuration in order to acknowledge the correctness of it. A configuration must be marked as correct in the knowledge base before proceeding to the next step.

Plausibility Check

Before starting the deployment of a verified configuration, several checks are carried out in order to test the plausibility of the new configuration. These checks aim to detect conflicts or inconsistencies and ensure that all devices in the configuration are available in the physical system. This step prevents foreseeable errors during deployment and thus prevents machine downtimes. After successfully passing the plausibility check, the configuration is marked in the knowledge base, and the procedure will proceed to the final step, the deployment.

Deployment and Validation

Once a configuration is verified and checked, the deployment to the physical devices can start. The reliable and smooth execution of this task relies highly on the correct sequence in which the subtasks are performed. The deployment of an FSS configuration, which consists of many artifacts, involves configuration activities on three levels: deterministic data transport using TSN, semantic information exchange using OPC UA, and safety connections using OPC UA Safety. These activities must be aligned with each other in order to enable seamless configuration changes.

If a configuration is new and has never been used before, it needs validation by the operator. This task is performed after the new configuration has been deployed by physically testing all safety measures. Therefore, an additional step is required using an HMI to validate the configuration and mark it as validated in the knowledge base.

Knowledge-Based System (KBS)

A KBS is located at the core of the SNC. A KBS is an entity that reasons and uses a knowledge base to solve complex problems and draw conclusions or make decisions. It has two characteristic features: a knowledge base, a technology used to store complex structured and unstructured information, and an inference engine, a component that applies logical rules to the knowledge base to derive new information.

The purpose of the KBS within an FSS is to provide a uniform platform for storing complex structured and unstructured information and to apply defined logical rules to this information to derive conclusions.

6.4.2 System Services

The SNC has to provide several services to fulfill the required tasks defined by the configuration procedure. These services can be grouped into discovery, deployment, knowledge representation, visualization & modification, and configuration. Figure 6.12 illustrates the service groups with their services, which must be made available by the SNC to allow the operator to carry out safety configuration changes quickly and efficiently.

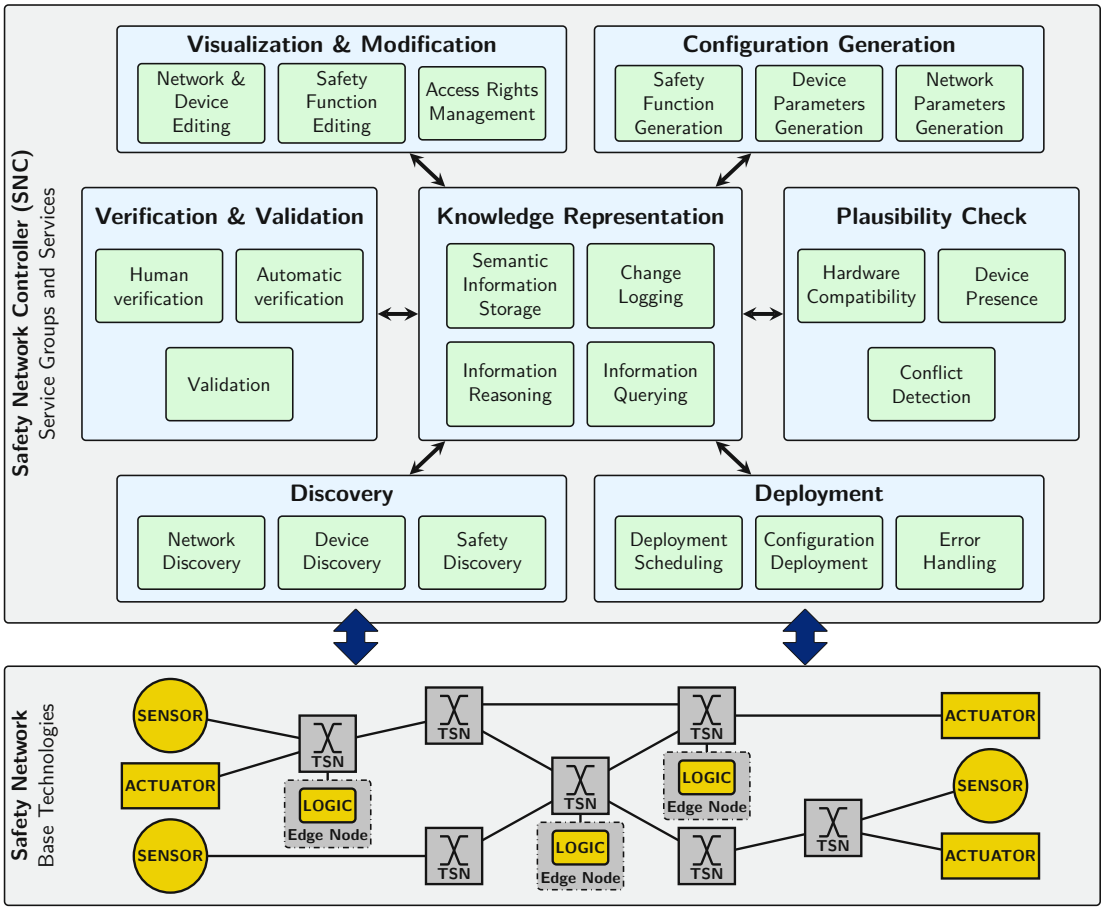


Figure 6.12: FSS service groups and services

The physical components of an FSS consisting of network bridges, network links, safety sensors, safety logic, and safety actuators build the safety network as depicted in Figure 6.12. The safety network is controlled by the SNC with its services and applications, which can be hosted in any virtual environment with physical access to the safety network. Furthermore, all services communicate via network interfaces, which allows the services to be distributed and hosted on different devices. In the following, the individual services will be described in detail.

Discovery

The discovery service group comprises three services that jointly create an actual and up-to-date image of the physical system. These services are network discovery, device discovery, and safety attributes discovery.

The first step in the discovery process is concerned with the network topology and is accomplished within the **network discovery service**. The service discovers the network topology consisting of network endpoints, bridges, and links utilizing LLDP and NETCONF. LLDP is an open and vendor-independent Layer 2 (Data Link Layer) Protocol, which is defined in IEEE 802.1AB. Its purpose is to advertise the identity and capabilities to the connected network neighbors. Each device needs to run an LLDP Agent, which collects the information of the neighbors and publishes its information periodically. The LLDP agent does not provide any functionality to provide its information via network service. Therefore, the collected neighbor information can be accessed via NETCONF to identify neighbor devices. The information elements used in an LLDP Protocol Data Unit (PDU) consist of sequential type, length, and value fields (TLV). Some basic TLVs are defined in IEEE 802.1AB, which can be mandatory or optional. The most common types are listed in Table 6.3.

Type	Name	Description	Usage
1	Chassis ID	An administratively assigned name that identifies the particular chassis within the context of an administrative domain that comprises one or more networks	Mandatory
2	Port ID	An administratively assigned name that identifies the particular port within the context of a system	Mandatory
3	Time To Live	The Time To Live TLV indicates the number of seconds that the recipient LLDP agent is to regard the information associated with this MAC service access point (MSAP) identifier to be valid	Mandatory
4	Port Description	The Port Description TLV allows network management to advertise the IEEE 802 LAN station's port description	Optional
5	System Name	The System Name TLV allows network management to advertise the system's assigned name	Optional
6	System Description	The System Description TLV allows network management to advertise the system's description	Optional
7	System Capabilities	The System Capabilities TLV is an optional TLV that identifies the primary function(s) of the system and whether or not these primary functions are enabled	Optional
8	Management Address	The Management Address TLV identifies an address associated with the local LLDP agent that may be used to reach higher layer entities to assist discovery by network management	Optional

Table 6.3: Selected TLV types according to IEEE 802.1AB

With the mandatory information elements, it is possible to create a unique model of the network topology. The optional information elements serve as additional information

for the network operator. An LLDP Agent does not provide a network interface for clients to collect information. Therefore, the information elements are encoded in XML format using specified YANG models and made accessible via NETCONF, which provides mechanisms to read, write, change, and delete the configuration of network devices.

The next subsequent discovery process step is the **device discovery**, where the end devices are identified and aligned with the network topology. The service uses the mechanisms from LDS and GDS defined in OPC UA Part 12: Discovery and Global Services to gather semantic information from the relevant devices, which are safety sensors, safety logic, and safety actuators. This part of the discovery targets generic device information essential to establish communication relations between safety devices using the OPC UA PubSub communication pattern, including secure channels with their certificates. Therefore, knowing the endpoints that an OPC UA server provides is necessary. Usually, a minimum of two endpoints are supported by an OPC UA server: one endpoint to support the discovery services and another endpoint to support server operation with a specific transport, encoding, and security profile. The OPC UA local discovery process is illustrated in Figure 6.13.

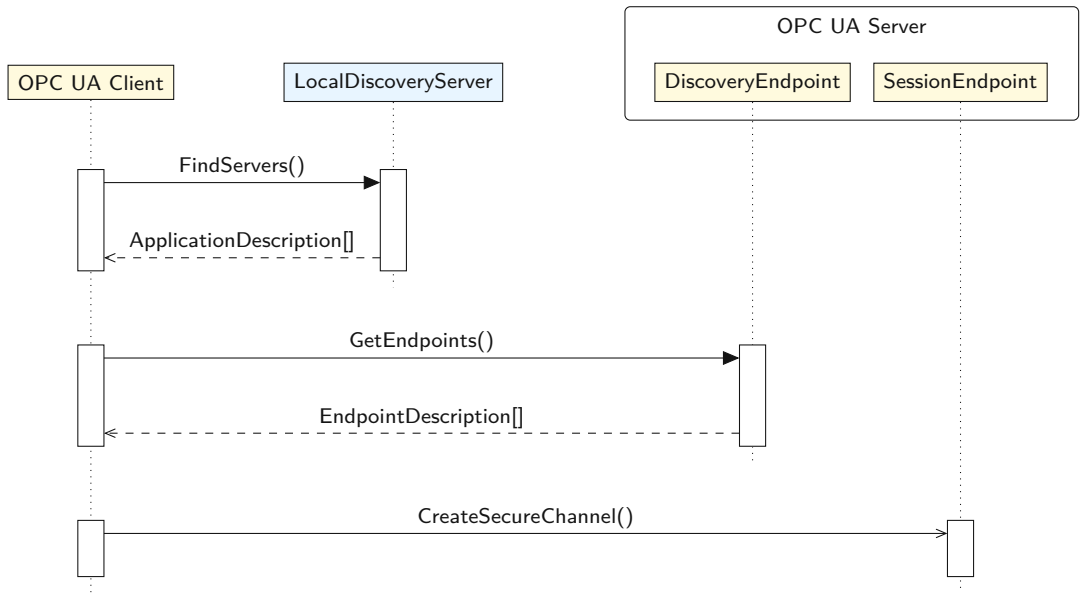


Figure 6.13: OPC UA local discovery process

Similar to the local discovery process using an LDS, there is also the possibility of using a global discovery process using an GDS. Instead of the method FindServers() at the LDS, at the GDS, the client uses the call service to invoke the QueryApplications() method to retrieve a list of devices with OPC UA servers that meet the filter criteria provided. Comparable to the FindServers() method of the LDS, the QueryApplications() method returns a list of servers and their DiscoveryUrls.

Once an OPC UA communication channel is established, the last step in the discovery process can start, the **discovery of safety attributes**. In doing so, the OPC UA Safety information model of the safety devices is scanned, reading the relevant parameters that are required to configure that device as part of a safety function. The respective parameters for safety provider and safety consumer are summarized in Table 6.4.

Safety Provider Parameter Name
SafetyProviderIDConfigured
SafetyProviderIDActive
SafetyBaseIDConfigured
SafetyBaseIDActive
SafetyProviderLevel
SafetyStructureSignature
SafetyStructureSignatureVersion
SafetyStructureIdentifier
SafetyProviderDelay
SafetyServerImplemented
SafetyPubSubImplemented

(a) Provider parameters

Safety Consumer Parameter Name
SafetyProviderIDConfigured
SafetyProviderIDActive
SafetyBaseIDConfigured
SafetyBaseIDActive
SafetyConsumerIDConfigured
SafetyConsumerIDActive
SafetyProviderLevel
SafetyStructureSignature
SafetyConsumerTimeout
SafetyOperatorAckNecessary
SafetyErrorIntervalLimit
SafetyClientImplemented
SafetyPubSubImplemented

(b) Consumer parameters

Table 6.4: OPC UA Safety device parameters as defined in [131]

When the final step of the discovery is completed, the gathered information is stored in a semantic information storage using an update and query language.

Deployment

Besides the discovery services, only the deployment services have direct access to the physical system components of an FSS. The aim of these services is to ensure the quick and reliable transfer and activation of dependent configurations on many devices and several layers in the correct order. In order to accomplish this complex task, the services deployment scheduling, configuration deployment, and error handling are needed.

A comprehensive system configuration of an FSS comprises configurations of many devices and several functionalities on these devices. Given the importance of keeping the configuration procedure as short as possible to avoid unnecessary machine downtime, finding the best combination of sequential and concurrent dispatching of the individual configurations is of utmost importance. Therefore, the purpose of the **deployment scheduling service** is to find the most suitable schedule for deployment for a given system configuration. The computed schedule is stored in the knowledge base, and the next step, the deployment, can be triggered.

When a valid system configuration and the matching deployment schedule are available in the semantic information storage, and the SNC triggers its deployment, the **configuration deployment service** comes into action. Via the northbound interface, utilizing SPARQL Protocol and RDF Query Language (SPARQL), the deployment logic obtains the system configuration with its deployment schedule from the semantic information storage. After processing the input data according to its target, the individual configurations are deployed according to the schedule via two southbound interfaces to the devices. For the TSN bridges, NETCONF is used, and for the safety devices, OPC UA. The structure of the configuration deployment service with its interfaces is depicted in Figure 6.14 [132].

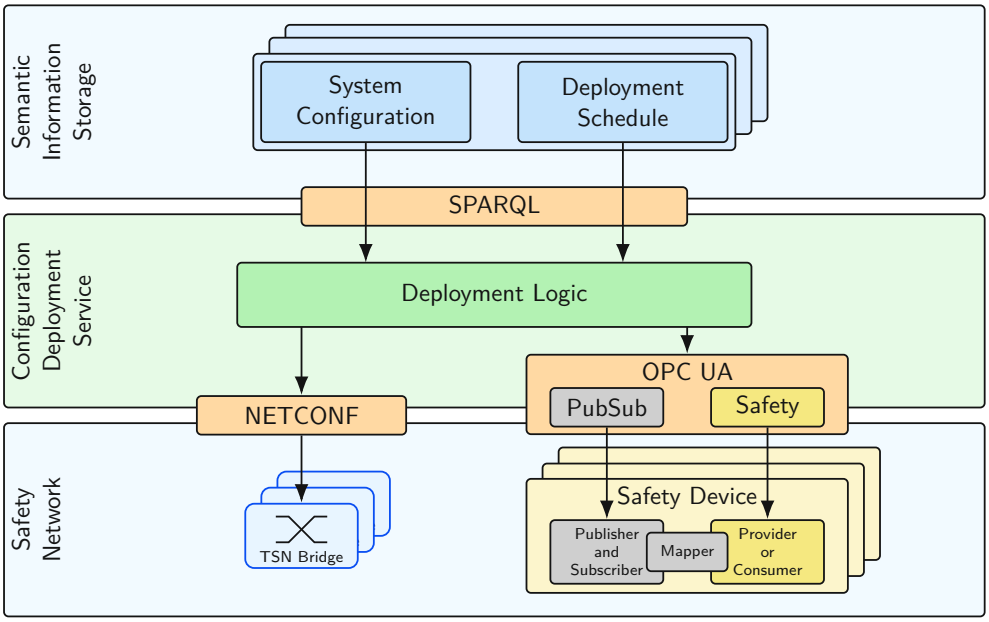


Figure 6.14: Configuration deployment service structure

The main task of the deployment logic is to establish communication channels for writing and activating the individual configurations to all devices using the appropriate interface and protocol. In doing so, particular attention is paid to concurrent and sequential tasks according to the deployment schedule.

An essential element before and during configuration deployment is the proper handling of potential errors that can occur in the course of the deployment process. Therefore, the **error handling service** acts in case of an error to ensure that the system is always in a defined state and informs the operator about the current state and actions. Its tasks include, in particular, analyzing the reasons for a failed deployment and finding a solution to return to a defined state from where the operation can continue. Possible approaches to react to unforeseen situations, such as non-responsive devices or interrupted communication, are rollback to a consistent configuration state or using a different system configuration from the semantic information storage.

Knowledge Representation

The central point of all services within the SNC is the knowledge representation, which includes services for semantic information storage, information reasoning, information querying, and change logging. These services build the heart of the SNC. All other services rely on the features provided by the knowledge representation.

All knowledge within an FSS is stored in a repository using a **semantic information storage** that allows storing, querying, and managing complex structured and unstructured information. Thereby, ontologies are applied using a semantic data schema paradigm where the ontologies are stored and managed independently from the data. This approach allows changing the data schema without interfering with the data, automatically discovering new facts, creating new knowledge based on semantic rules, and seamlessly integrating data from distributed data sets and data sources. Other than a relational database, an ontology can be seen as a flexible, interconnected, and interlinked graph data model. An advantage of this concept is the possibility of reusing already existing ontologies. For the use within an FSS, several ontologies from the domains of system architecture, asset properties, network, management, capabilities, and information model were examined and benchmarked in [140]. The combination of existing ontologies for the use as a knowledge base for FSS is illustrated in Figure 6.15.

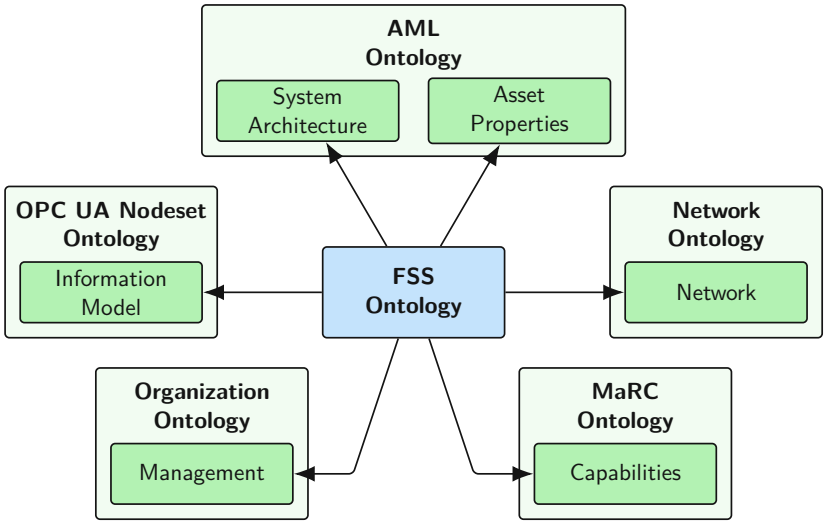


Figure 6.15: Ontology structure for FSS

The FSS ontology depicted in Figure 6.15 defines the data structure used in the FSS semantic information storage and also imports parts from existing ontologies. The AutomationML Ontology (AMLO) covers the Computer Aided Engineering Exchange (CAEX) part of the AutomationML standard, which enables the modeling of complete systems, starting from single automation components to entire complex production systems and supports the representation of different aspects of these systems, such as

geometry, kinematics, and topology. The Network Ontology for computer network management aims to cover the aspects of the network domain necessary for monitoring and controlling purposes. This ontology defines, among many others, objects for representing the communication infrastructure of a network (NetEntity class), objects that describe the traffic within a network (TrafficEntity class), objects that describe a misfunctioning state of the network (Abnormality class), and objects for modeling generic and specific events within a network (Event class). The Organization Ontology for organizational structures covers the main concepts that describe organizations and their relations. It intends to describe a collection of people organized together in some structure (Organization class), an organization recognized in legal jurisdiction and has rights and responsibilities (FormalOrganization class), and a collaboration between organizations (OrganizationalCollaboration class). Additionally, a specific organizational role can be assigned to a person or an agent (Role class) to indicate a person's affiliation to an organization and can include a time interval to establish the membership duration. The Manufacturing Resource Capability Ontology (MaRCO) provides objects for describing the capabilities of manufacturing resources, and its initial aim was to automatically support matchmaking between resource capabilities and product requirements. The ontology describes models for defining the hierarchical categorization of different manufacturing processes (Process Taxonomy Model), for product characteristics and manufacturing requirements (Product Model), for specifying the capability names, parameters, and relations between them (Capability Model), and for defining the resources and the system composed of them (Resource Model). The OPC UA NodeSet Ontology covers eight classes of the OPC UA base model UAReferenceType, UADataType, UAObjectType, UAObject, UAVariableType, UAVariable, UAMethod, and UAView representing the OPC UA base nodes. The initial goal of this ontology was to use the information models used by OPC UA in combination with the geometry and kinematic model of a resource to create a digital twin. In OPC UA, each node has a set of mandatory and optional attributes that provide further information about the node's properties. The most important attribute is the nodeId, unique for each node and used to identify it within the address space. Such properties are represented in the ontology via data properties [140].

Using a semantic information storage as a knowledge base for FSS opens up the possibility of **information reasoning**. In this context, reasoning is a process of performing inference to derive facts that are not explicitly expressed in an ontology or the knowledge base. Thereby, inference relies on two tools: ontologies and rules. It aims to obtain new results from a combination of rules and a set of activation conditions of these rules by reasoning on contextual parameters. This additional knowledge can support the FSS in generating new safety configurations but also provide valuable insights for the operators.

Besides the information provided by the semantic information storage, access from other services and its interface is vital. Therefore, an **information querying service** creates the preconditions for generic access for all FSS services. These services shall be able to access and update information within the knowledge base via a semantic interface without the risk of creating an inconsistent state.

An essential part of a safety system is its ability to document and track all changes made to its configuration. Therefore, the **change logging service** records all changes made to the knowledge base with timestamp and origin in a not modifiable manner.

Visualization & Modification

Data preparation and processing for human user interaction is accomplished within the **visualization & modification services**. These services include network and safety device editing, safety function editing, and access rights management. All these services are based on web technologies that allow the presentation of the GUI on different displays of various sizes and offer a good user experience. The basic structure for the FSS GUI, which is divided into frontend and backend applications, is depicted in Figure 6.16.

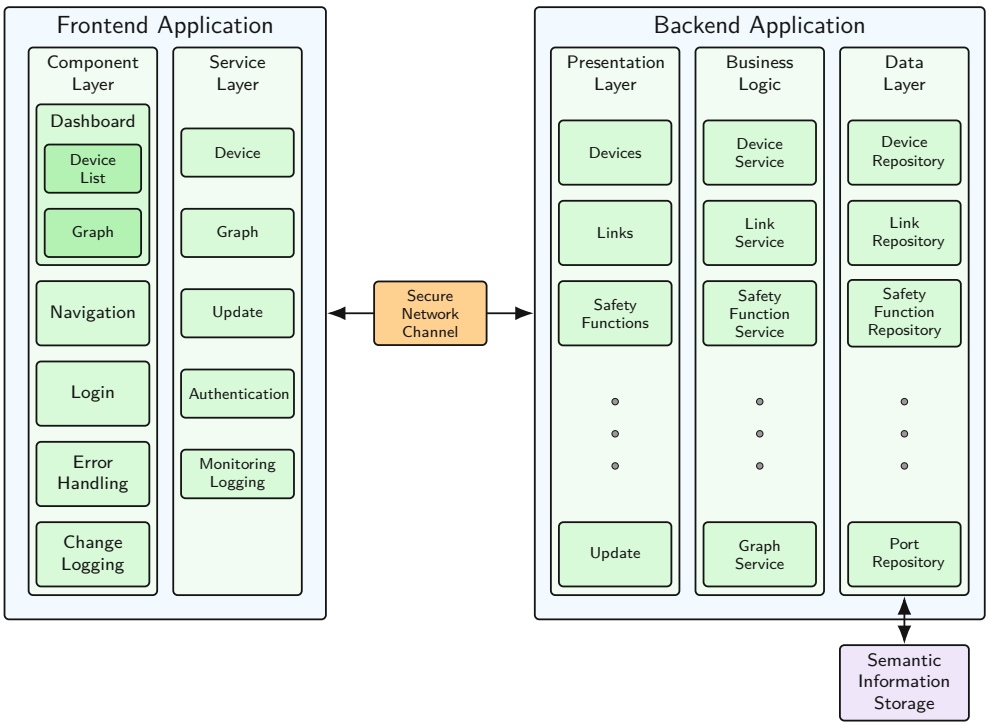


Figure 6.16: Visualization structure for FSS

To view and edit network devices, network topology, safety devices, and safety connections, the **network and device editing service** uses data from the semantic information storage and the underlying mechanisms for network, device, and safety discovery to create a graphical representation of the system and to provide a GUI for the operator and safety engineer. Additionally, the GUI allows adding objects and modifying the presented devices, links, and their parameters. Furthermore, the user can initiate a configuration deployment and review the change logging.

One of the essential elements of an FSS is the safety functions that define the safety system's operation. Therefore, the design and realization of the safety functions is an important feature implemented in the **safety function editing service**. This service aims to present safety functions from the semantic information storage to the user and allows adding, deleting, and modifying these.

A fundamental feature in a safety-related system is user authentication. For that reason, a **access rights management service** controls the access to features and data of the FSS. This service includes, besides user authentication, role management, and administration of certificates.

Configuration Generation

Many tasks in an FSS are very complex, very time-consuming, or both. For those tasks, it is imperative to assist the operator in the form of configuration generation tools to allow quick configuration changes. Essentially, these tools are needed for safety function generation, safety device parameters generation, and network parameters generation.

The core element of an FSS configuration is a safety function consisting of safety sensors, safety logic, safety actuators, and communication between those. The **safety function generation service** creates template configurations for safety functions for a given system situation. The system state is derived from the discovery services and the data stored in the semantic information storage. These templates can be viewed and edited by the operator and stored in the semantic information storage.

Another difficult task is the creation of configurations for all safety devices within the network. This task is accomplished by a **device parameters generation service**, which generates a configuration file for each safety device with the information in the knowledge base. These configurations can be viewed and edited by the operator and stored in the semantic information storage. Special attention must be paid to the safety device type, the safety provider or the safety consumer, and the specific safety variables needed to establish a safety connection between two safety devices. Besides generating the parameters for the safety communication, this task also includes generating the parameters for the OPC UA PubSub communication underneath.

Preparing the configuration files for the TSN bridges is one of the most complex tasks. The challenge here is to create configurations for each network device that must be consistent throughout the whole network. Therefore, the **network parameters generation service** produces consistent network device configurations considering network synchronization, the TAS for all egress ports of the bridges, and the network links. The main task is to create the Gate Control Lists (GCLs) for each egress port of all TSN bridges as illustrated in Figure 6.9. The outputs of this service are configuration files stored in the semantic information storage, where the operator can access them via the GUI and by the deployment services.

Verification & Validation

A new or modified safety configuration requires verification before deployment to ensure that risk assessment and all applicable safety regulations have been appropriately considered. Furthermore, the system has to provide functionalities for the safety engineer to validate a deployed configuration before releasing it for operation.

An essential precondition for a safety configuration to be deployed is its verification. The **human verification service** provides the integral parts for tagging a configuration as verified by the safety engineer and implements measures to protect a verified configuration against subsequent changes.

Additionally, an **automatic verification service** supports the safety engineer during the safety engineering process by checking a configuration in progress against relevant safety regulations and standards, which allows drawing attention to errors at an early stage and, therefore, saving engineering time.

A mandatory feature of a safety system is the possibility of validation. Therefore, the **validation service** provides all the necessary components for conducting the validation procedure after deployment at the production machinery and for tagging a configuration as validated by the safety engineer. Furthermore, it implements measures to protect a validated configuration against subsequent changes.

Plausibility Check

Immediately before deployment, each safety configuration must pass a series of plausibility checks to prevent foreseeable problems leading to production problems or a standstill of the production system. In this respect, a configuration will only be deployed and activated if the configuration has passed the plausibility check.

The first and simplest check is carried out with the **device presence service** and examines the availability and completeness of all required devices. This check includes network devices, network links, and safety devices.

On successfully passing the first service, the **hardware compatibility service** checks if all devices fulfill the hardware requirements, such as the bandwidth of a network link, port capabilities and status of network bridges, capabilities of safety-relevant inputs and outputs, and SIL of the safety-relevant devices.

Finally, the **conflict detection service** aims to find potential conflicts within the individual components of the configuration, such as conflicting bandwidth reservation in different TSN bridges, which would lead to packet loss for a safety-related communication link, consequently bringing the machine into a safe state.

6.5 Control Interfaces

Controlling of an FSS can take place in two different ways, from a GUI via Human-Control Interface (HCI) and from an MES or ERP system via M2MCI as illustrated in segment 4 in Figure 6.1. In providing these interfaces, the FSS also has to consider the security of the communication, user authentication, user authorization, and concurrency of requests from various clients either via HCI or M2MCI.

6.5.1 Human-Control Interface (HCI)

The primary purpose of the HCI is to allow access from various HMI clients to the visualization and modification services of the FSS within the SNC.

For reasons of confidentiality, interoperability, and low latency, the interface is based on the Hypertext Transfer Protocol Secure (HTTPS), where the communication protocol is encrypted using Transport Layer Security (TLS) and WebSocket Secure (WSS) defined by the IETF as RFC 6455, which allows achieving communication with soft real-time behavior allowing fast updating of the GUI in case of changes in the system. Both protocols use the TCP port 443, which makes it simple for the administrators to configure a firewall rule.

6.5.2 Machine-to-Machine-Control Interface (M2MCI)

The main objective of the M2MCI is to enable safety configuration changes originating from a manufacturing orchestration system such as an MES or an ERP system.

Similar to the HCI, the M2MCI has to support confidentiality and interoperability. Therefore, OPC UA is used to expose information to other systems and allow to apply modifications. Besides secure communication based on HTTPS or WSS, OPC UA also supports user authentication and authorization, which makes it an ideal candidate also from the aspect that OPC UA is already used inside the FSS as a communication platform. Using HTTPS or WSS as a secure transport channel limits the ports that must be exposed to TCP port 443, which can be shared with the HCI and, therefore, simplifies firewall configuration.

The information model of the OPC UA server has to cover several aspects of an FSS. Essentially, these are the system status, configuration, and change logging. The system status objects indicate the current value of the operational and administrative state of the safety system. The system configuration objects consist of a folder for configurations and methods related to safety configurations, such as reading predefined configurations and activating a selected configuration. The change logging objects contain methods for reading the safety change log file of the system. Figure 6.17 illustrates a simplified information model of the FSS OPC UA server. The OPC UA diagram shows on the very top the FlexibleSafetySystemType, which is a subtype of the OPC UA BaseObjectType.

The FlexibleSafetySystemType object type includes three components FssStatus, FssConfigs, and FssLogging. The FssStatus object is a component with the type definition

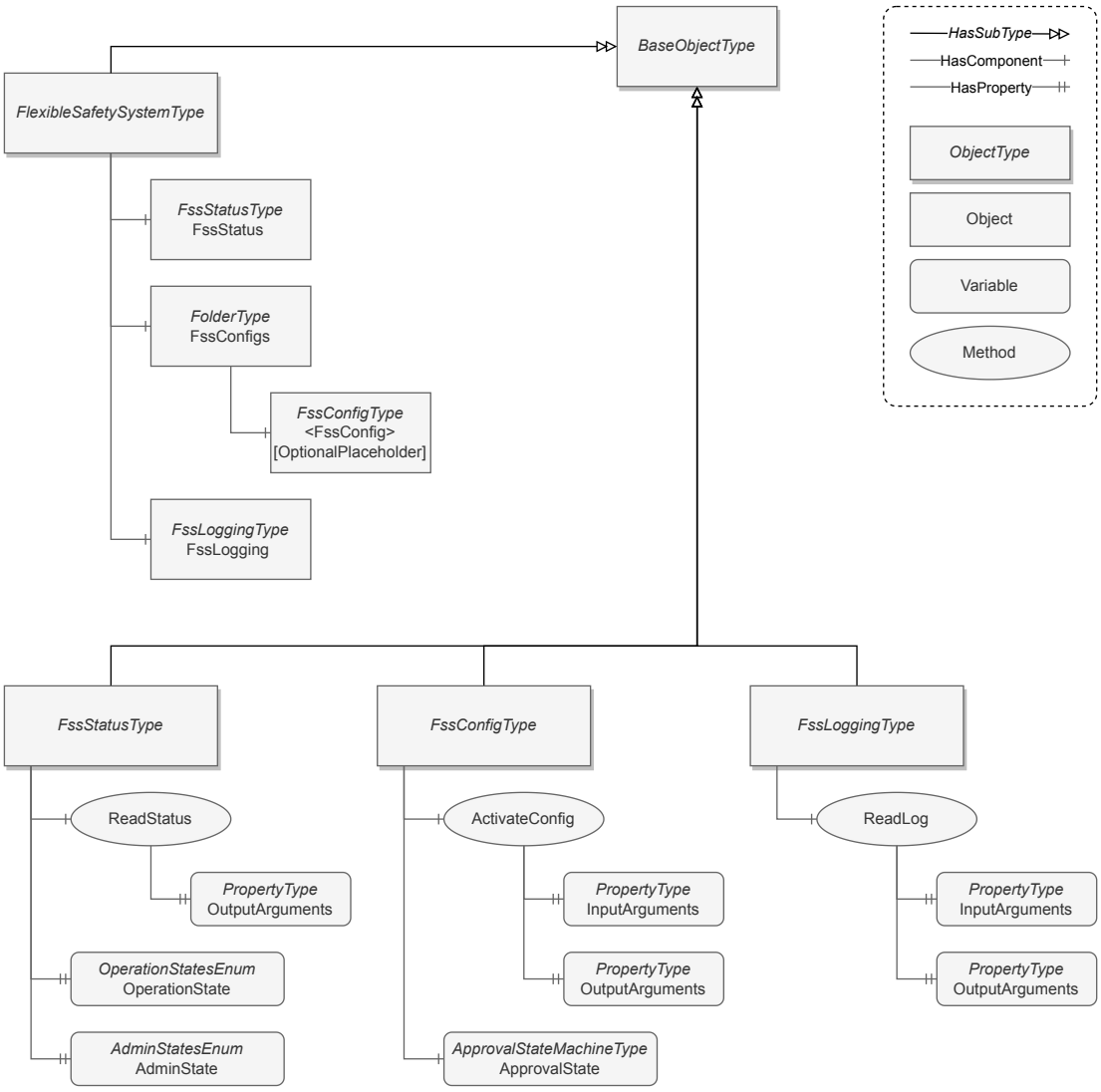


Figure 6.17: OPC UA diagram for M2MCI server objects

FssStatusType, which defines the operation and admin state of the system and provides a method for reading the status. The FssConfigs object is a folder for FssConfig objects of the type FssConfigType, which defines a component ApprovalState and provides a method for activating the configuration. The FssLogging object is a component with the type definition FssLoggingType, which defines a method for reading the change logging entries in the semantic information storage. It takes filter arguments such as time period or line count as input and returns requested content.

CHAPTER 7

Software Architecture

> ALTHOUGH DEFINING ARCHITECTURE IN THE CONTEXT OF COMPUTING MIGHT SEEM LIKE A SIMPLE TASK, IT BECAME ONE OF THE MOST CONTENTIOUS ISSUES IN DEVELOPING THE IEEE STANDARD 1471. THIS IS NOT SURPRISING CONSIDERING THAT, DESPITE APPROXIMATELY 5,000 YEARS OF PRACTICE, THE CIVIL ARCHITECTURE COMMUNITY HAS HAD LITTLE MORE SUCCESS IN PRECISELY DEFINING A BUILDING'S ARCHITECTURE. [141]
>
> – Mark W. Maier, 2001

Following the path of the Flexible Safety System Development Life Cycle (FSS-DLC) depicted in Figure 3.2, the system design phase is followed by the software architecture phase. This chapter derives a software architecture model from the results of the previous phases, the modular system design, and the base technologies. Furthermore, software architecture artifacts are the outcomes created in this phase.

The central component for managing all hardware devices of an Flexible Safety System (FSS) is the Safety Network Controller (SNC) as illustrated in Figure 6.1. The aim of the SNC is to manage all tasks to configure and monitor the TSN network, OPC UA information models, and OPC UA Safety communication. Furthermore, the SNC provides interfaces for human control and Machine-to-Machine (M2M) control.

The modular design of the SNC, proposed in the previous chapter, divides the functionalities into service groups and services, as depicted in Figure 6.12. In the following, a software architecture model is presented using the C4 (context, containers, components, and code) model for visualizing software architecture[1]. In doing so, the service groups will be modeled as containers and the services as individual components that encapsulate functionality behind a well-defined interface.

[1] https://c4model.com

7.1 Context

To help to fully understand the purpose and environment of the SNC, which is the central core part of the system, the context diagram depicted in Figure 7.1 illustrates the users and the other external systems it interacts with.

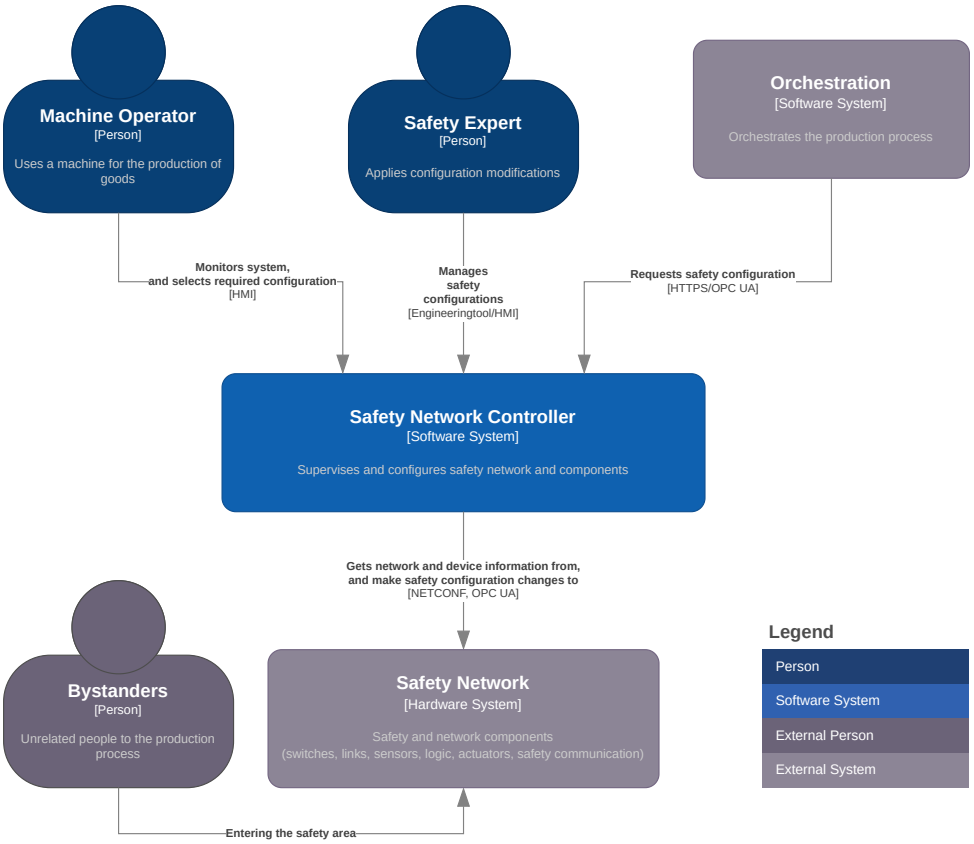


Figure 7.1: FSS software stucture context

To put the SNC into context, Figure 7.1 shows the controlling and controlled persons and external systems. Northbound, the controlling entities are depicted: machine operators, safety experts, and an orchestration system. The controlled safety network is placed southbound of the SNC, including bystanders who are unrelated people to the machine's safety system or the production process but must be taken into account for safety reasons.

The following sections take a detailed look at the SNC with its services and functions. In particular, following the C4 methodology, zooming in from a high-level system view to concrete functions and their code artifacts.

7.2 Containers

The container diagram in Figure 7.2 illustrates a high-level view of the SNC and its service groups mapped into containers that are essentially separately runnable and deployable units that execute code or store data. It also shows how responsibilities are distributed, the major technology choices, and how the containers communicate.

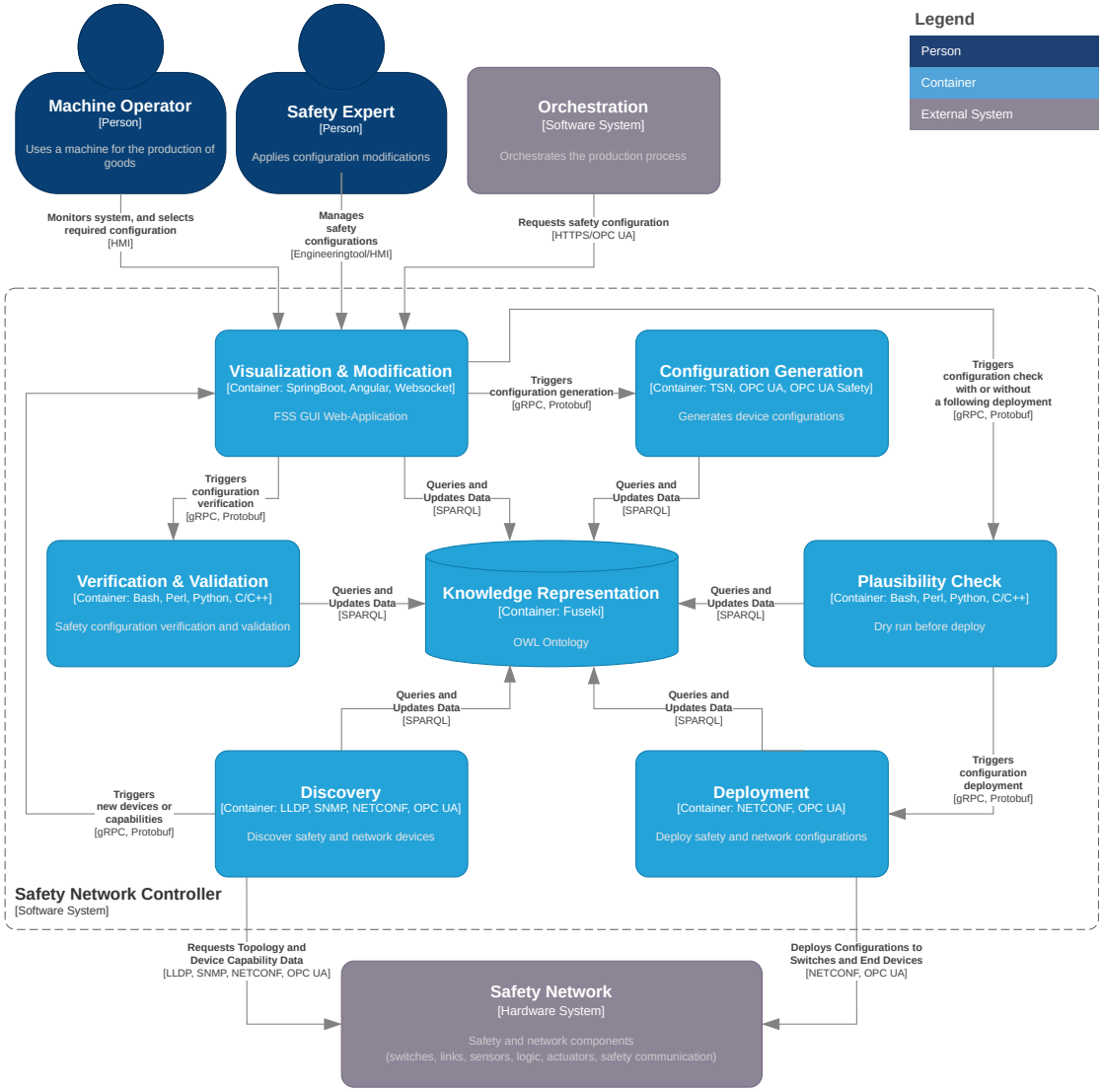


Figure 7.2: FSS software stucture containers

Essentially, the SNC consists of six application containers (discovery, deployment, visualization & modification, configuration generation, verification & validation, and plausibility check) and a data store container (knowledge representation). Data exchange between

the application and data store containers is accomplished using SPARQL queries. Communication among application containers is realized with gRPC and Protocol Buffers. The southbound interfaces connect the discovery and deployment containers to the safety network. The visualization & modification container provides the northbound control interfaces for machine operators, safety engineers, and orchestration systems.

7.3 Components

In the next step, each container is decomposed into components to identify the major structural building blocks and their interactions.

7.3.1 Discovery

Figure 7.3 illustrates the discovery container with its components network discovery, device discovery, and safety discovery connected southbound to the safety network and northbound to knowledge representation and visualization & modification containers.

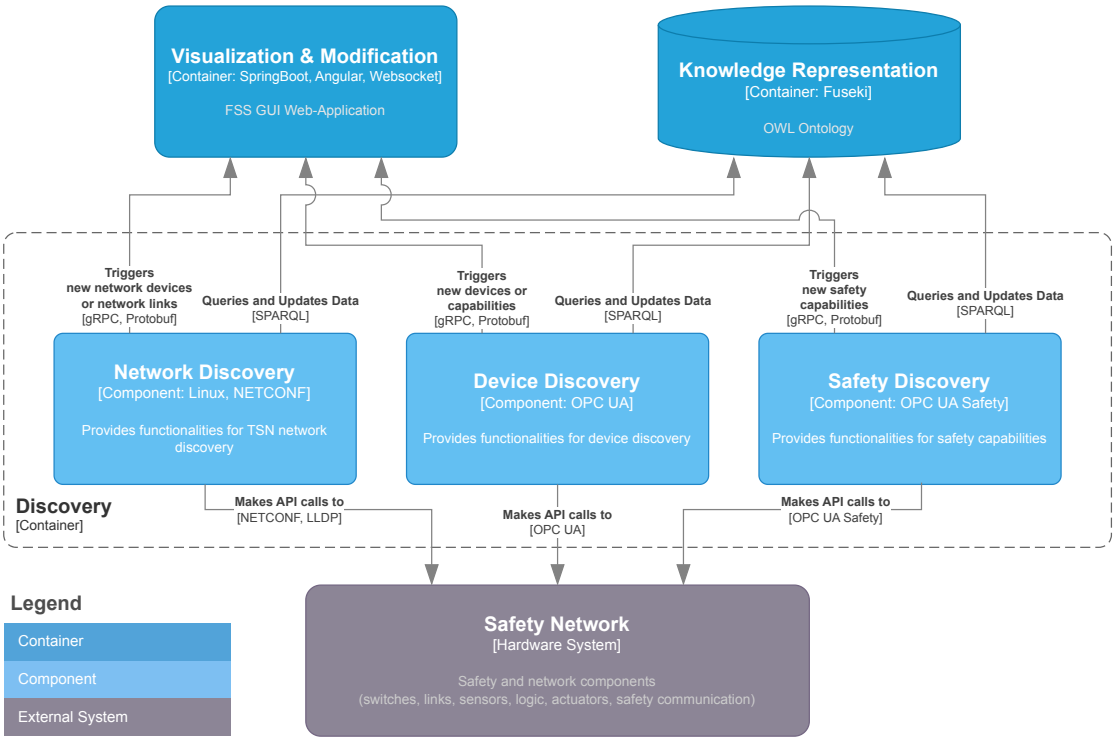


Figure 7.3: Discovery components

All three services in the discovery container discover their part of the system independently. When a change is recognized, it is written in the knowledge representation, followed by the respective trigger to the visualization & modification container.

7.3.2 Deployment

Figure 7.4 illustrates the deployment container with its components deployment scheduling, configuration deployment, and error handling. The components are connected southbound to the safety network and northbound to knowledge representation, visualization & modification, and plausibility check containers.

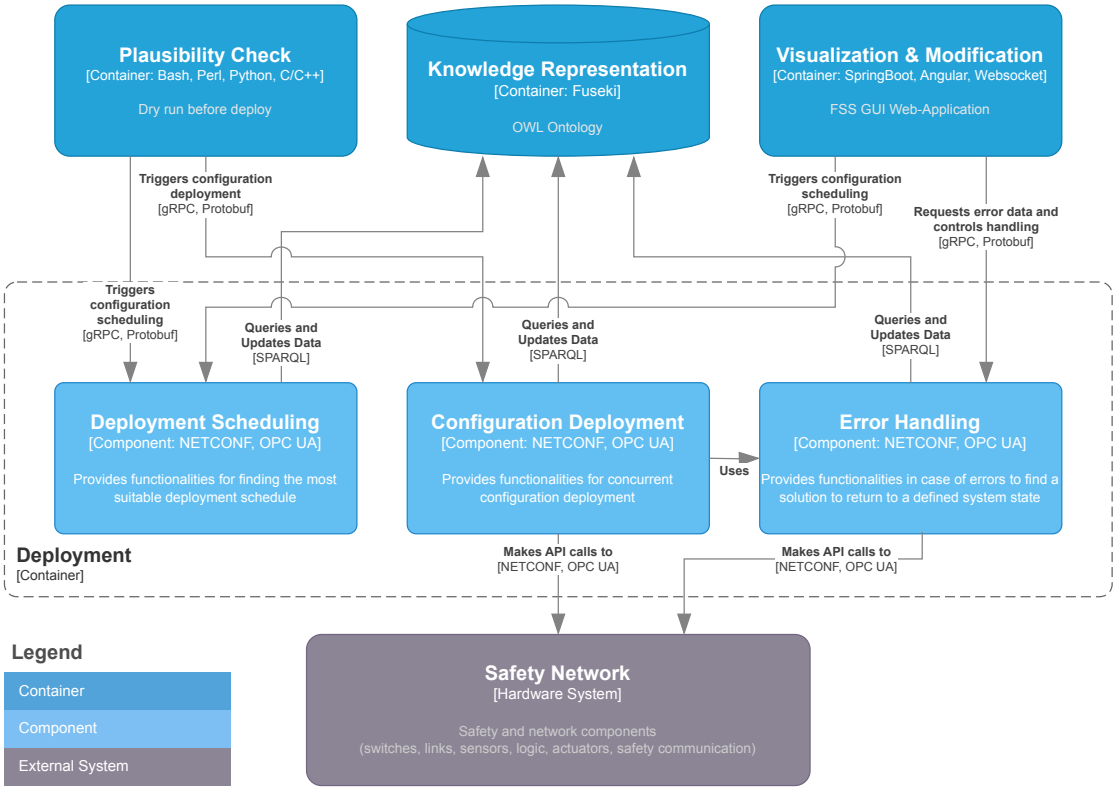


Figure 7.4: Deployment components

The deployment scheduling component calculates the required schedule for configuration deployment as depicted in Figure 6.14. It can be triggered from the plausibility check or visualization & modification containers. The resulting schedule is written into the knowledge representation. The configuration deployment component, triggered by the plausibility check, reads the configuration together with its schedule from the knowledge representation and deploys it into the safety network utilizing the error handling component. In case of errors, the error handling tries to find a solution to return to a defined state from where the operation can continue. Error handling can be monitored and controlled from the visualization & modification container.

7.3.3 Knowledge Representation

The knowledge representation is the central container within the SNC software system. All other containers are arranged around it and connected via a well-defined SPARQL interface. Figure 7.5 shows the components semantic information storage, information reasoning, change logging, and information querying, the last component acting as an interface for all other FSS services to connect.

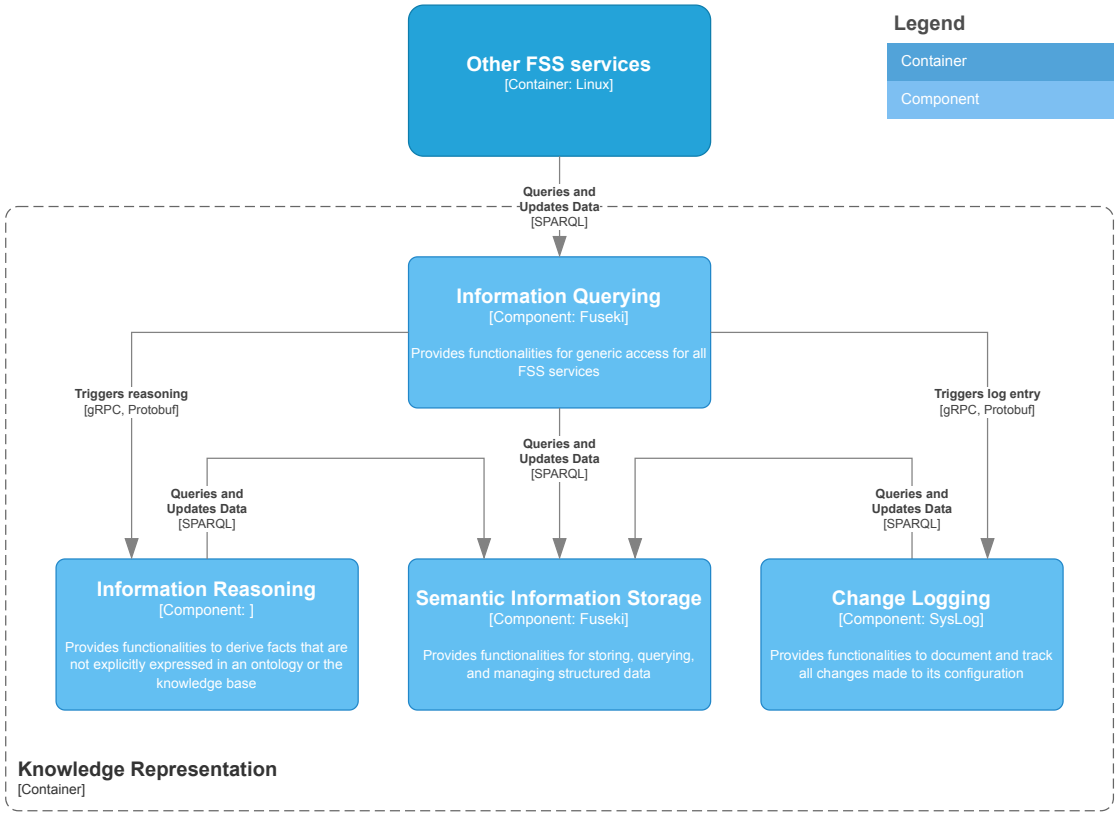


Figure 7.5: Knowledge representation components

The primary function of the knowledge representation container is located in the semantic information storage component. It stores all system knowledge and provides a SPARQL interface for the other components for querying and updating data. If required information is not explicitly available in the semantic information storage, the information reasoning component can be triggered to derive explicit knowledge from implicit knowledge. All transactions within the knowledge representation container are recorded and stored in an unalterable or subsequently changeable manner within the change logging component.

7.3.4 Visualization & Modification

All elements and functionalities required for external control of the FSS are combined within the visualization & modification container depicted in Figure 7.6. It contains the components of a sign-in controller, access rights management, configuration selection and deployment, safety function editing, and network and device editing.

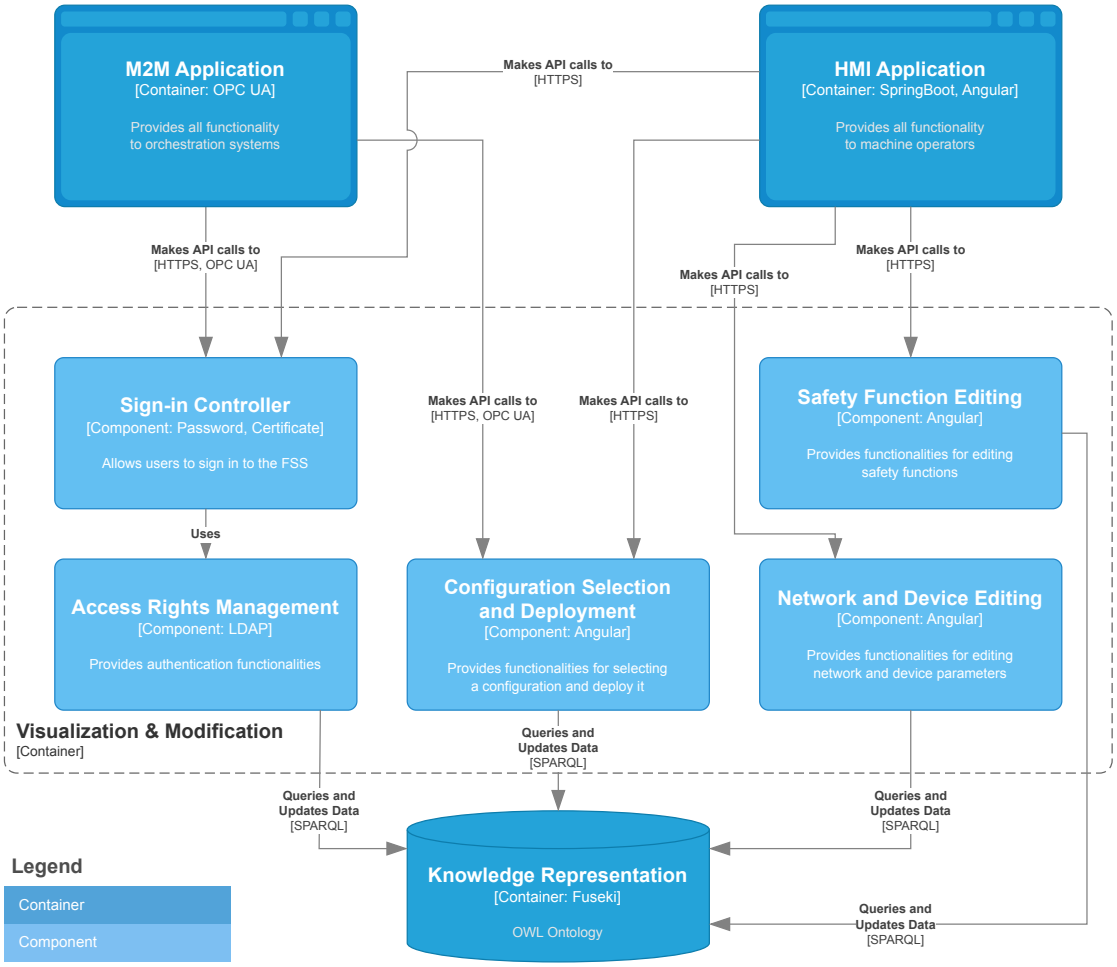


Figure 7.6: Visualization and modification components

External control is separated into human and M2M control. Both ways require the external application to authenticate in order to verify the identity of a user or service and to authorize their access rights. This task is accomplished by the sign-in controller and access rights management component. Depending on the granted access rights, the external applications can view and modify safety functions, network and safety devices, and select configurations for deployment. All data for those tasks are queried and stored in the knowledge representation container.

7.3.5 Configuration Generation

Initially, the configuration generation container contains the components for safety function generation, device parameters generation, and network parameters generation, as shown in Figure 7.7. These components represent the minimal functionality required for generating a configuration and can easily be extended due to the modular software architecture concept. All components work independently, are triggered by the visualization & modification container, query the required input from the knowledge representation container, and store the generated output in the knowledge representation container.

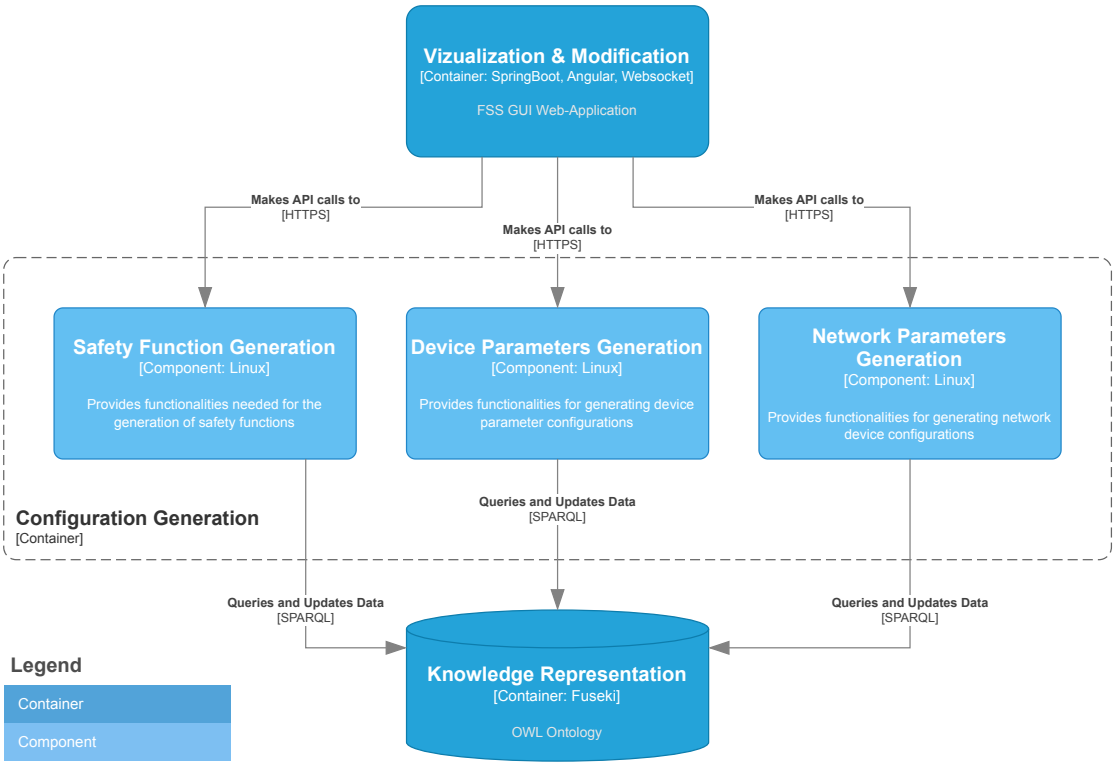


Figure 7.7: Configuration generation components

A safety system configuration consists of many artifacts. These artifacts are mostly configuration files intended for several devices on several layers. Therefore, services for automatically generating a vast number of files are paramount to ease the operation and reconfiguration of a highly complex system. As a basis, components for generating the parameters of the network and safety devices and for generating safety functions are foreseen. The network parameters generation component generates the schedules for the Time-Aware Shapers (TASs) in all Time-Sensitive Networking (TSN) bridges. The device parameters generation component generates the files to configure safety-related devices such as sensors, logic, and actuators. Finally, the safety function generation creates the files that describe all the safety functions within a system.

7.3.6 Verification & Validation

An essential feature for safety-related systems is the verification and validation of safety configurations. Therefore, the SNC software architecture includes a verification & validation container, illustrated in Figure 7.8, which provides the functionalities for human verification, automatic verification, and validation of safety configurations.

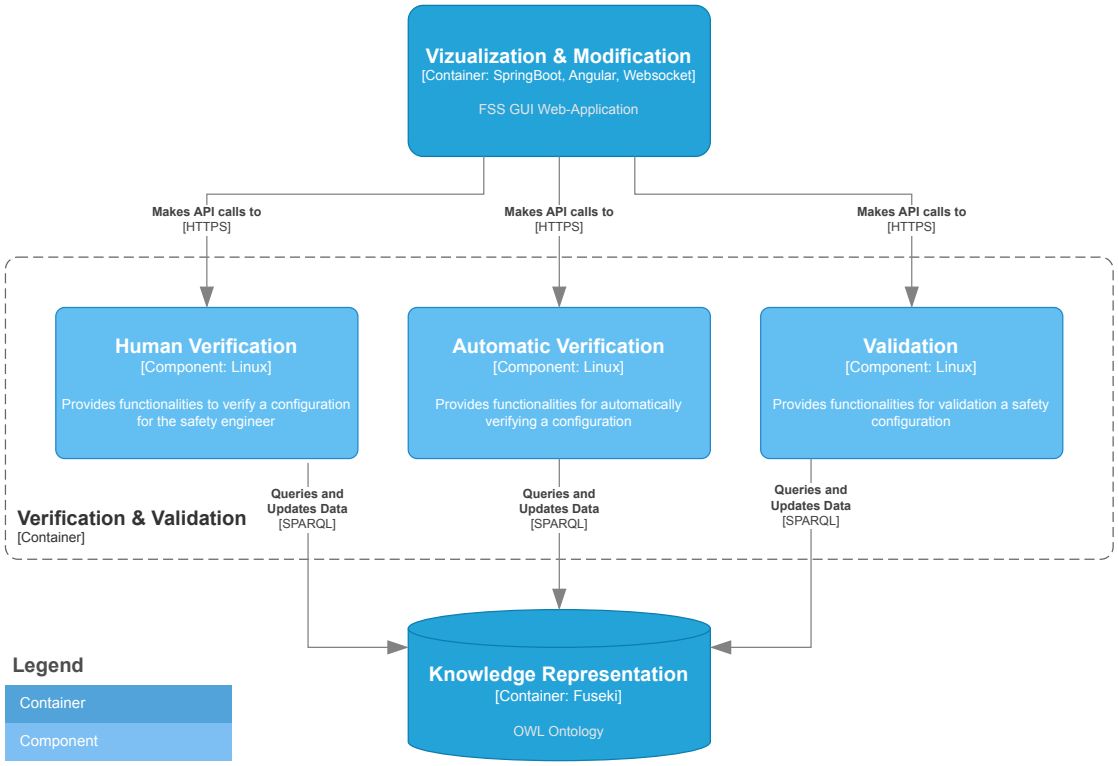


Figure 7.8: Verification components

The human verification component provides the functionalities for the safety engineer to verify a selected configuration and tag a configuration as verified. Furthermore, it implements measures to protect a verified configuration against subsequent changes. The automatic verification component is a tool to assist the safety engineer in the engineering process by checking a configuration in progress against relevant safety regulations and standards. In doing so, it queries the configuration under test, the machine model, legal regulations, and safety standards from the knowledge representation container. In the next step, it executes the automatic verification and informs the safety engineer via the visualization about possible faults. Once a verified safety configuration is deployed in the system, the safety engineer is required to check the correct operation in the real system. For that purpose, the validation component provides all functionalities to perform the validation procedure, including assistance via the Graphical User Interface (GUI) and implementing measures to protect a validated configuration against subsequent changes.

7.3.7 Plausibility Check

Each safety configuration must pass a series of plausibility checks immediately before deployment. This measure is foreseen to ensure that the deployment of a safety configuration does not lead to problems or a standstill of the production system due to missing or wrong devices or inconsistencies. These checks are carried out in the device presence, hardware compatibility, and conflict detection components depicted in Figure 7.9.

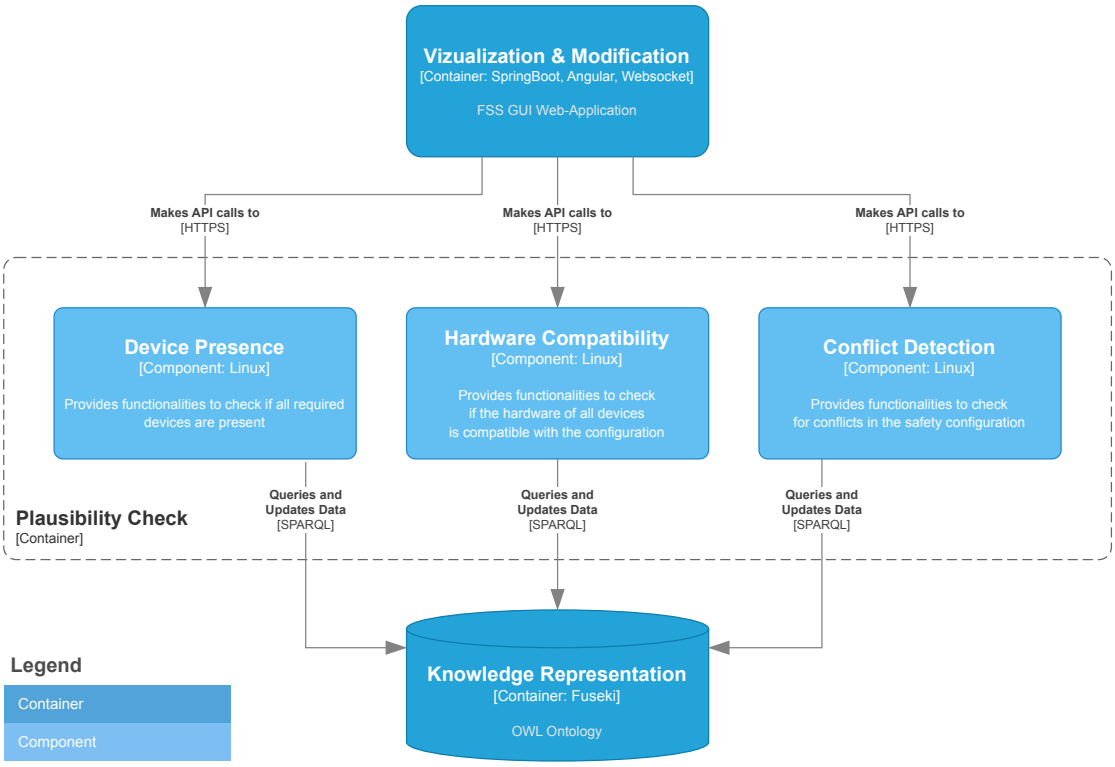


Figure 7.9: Plausibility check components

The device presence component checks if all devices within the safety configuration can be contacted and if a configuration is possible. If all required devices are available, the hardware compatibility component checks if all devices fulfill the hardware requirements. If all of this is successful, the conflict detection component aims to find potential conflicts within the individual components in the actual system and the new configuration.

All these checks aim to prevent unintended downtime of machinery due to mistakes or conflicts in configuration, such as wrong bandwidth reservation in different TSN bridges, which would lead to packet loss for a safety-related communication link and consequently bring the machine into a safe state.

7.4 Code

Several topics were identified and carried out as separate projects, such as bachelor or master thesis co-supervised by the author of this thesis, during the course on the work for a software architecture for a FSS. These projects covered specific aspects of the components defined in the overall software architecture. The assignment of those projects to the architecture is illustrated in Figure 7.10 and briefly presented in the following.

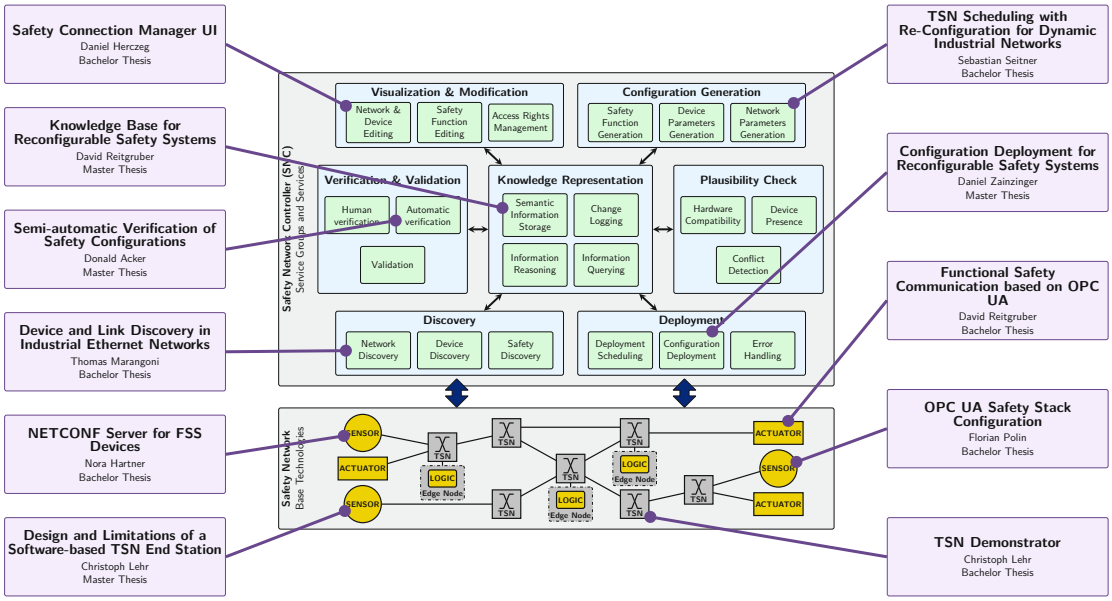


Figure 7.10: Mapping of artifacts to the overall architecture

7.4.1 Device and Link Discovery in Industrial Ethernet Networks

Bachelor Thesis [142] – Thomas Marangoni

This thesis implements a prototype of device and link discovery in industrial Ethernet networks. It detects devices and their ports with Link Layer Discovery Protocol (LLDP) and prepares a network graph with the collected data for the TSN standard. The prototype was tested in a laboratory environment.

7.4.2 Configuration Deployment for Reconfigurable Safety Systems

Master Thesis [132] – Daniel Zainzinger

This work shows how currently available technologies like Network Configuration Protocol (NETCONF) and OPC UA Methods can be used to perform a fast reconfiguration. Additionally, the limitations, especially for OPC UA Safety, are pointed out. The evaluation at the end of this work showed that 40 OPC UA Publishers can be enabled in 3.11 ms, which shows that this approach can be used for further works.

7.4.3 Safety Connection Manager UI

Bachelor Thesis [143] – Daniel Herczeg

Web-based solutions offer a range of advantages, such as good scalability, adaptability, and fast rollout of updates. This thesis discusses the development of a state-of-the-art web application that allows employees in an industrial environment to visualize safety-relevant network devices and manage them if necessary.

7.4.4 Knowledge Base for Reconfigurable Safety Systems

Master Thesis [140] – David Reitgruber

This thesis addresses some of the problems that arise using a dynamic approach to reconfigure safety systems. A knowledge base is developed, which stores safety-relevant information about the manufacturing system. The relevant domains are described, and existing ontologies covering these domains are researched and rated. New ontologies are created for domains that are not suitably covered by existing ontologies. The knowledge base is then evaluated based on before-defined competency questions and used in an application scenario.

7.4.5 TSN Scheduling with Re-Configuration for Dynamic Industrial Networks

Bachelor Thesis [144] – Sebastian Seitner

This work focuses on enhancing an existing and published heuristic approach to enable re-calculations in response to topology changes on existing network schedules to decrease the time it takes to reach a valid TSN network schedule. A proof-of-concept implementation and some evaluations are provided to judge the suitability of this approach.

7.4.6 Semi-automatic Verification of Safety Configurations

Master Thesis [145] – Donald Acker

The goal of this thesis is to investigate the possibilities of a semi-automatic verification of a safety configuration in progress against relevant safety regulations and standards to support the safety engineer during the safety engineering process. Such an assistive tool allows the detection of errors at an early stage and saves engineering time.

7.4.7 Functional Safety Communication based on OPC UA

Bachelor Thesis [146] – David Reitgruber

This thesis aims to connect two safe Programmable Logic Controller (PLC) systems via OPC UA and enable the exchange of safety data based on the ideas of the specification. For this, an example program gets created, which is then used to test and evaluate the functionality.

7.4.8 OPC UA Safety Stack Configuration

Bachelor Thesis [147] – Florian Polin

The aim of this thesis is to evaluate the practical usability of the OPC UA Safety Stack developed by the OPC UA Foundation. Furthermore, the configuration effort and the most suitable configuration procedure for an FSS shall be investigated. One of the base technologies of an FSS is OPC UA Safety. Work for developing a software stack has been started in 2021 by the OPC UA Foundation and resulted in a beta version available for academic partners in 2023.

7.4.9 Design and Limitations of a Software-based TSN End Station

Master Thesis [148] – Christoph Lehr

This thesis addresses several challenges that arise when implementing a TSN end station. First, a clock synchronization algorithm had to be implemented, allowing the end station to synchronize with the TSN network. Second, TSN organizes Ethernet packets in different queues according to their priority. Therefore, a real-time memory management system was implemented. Third, a network interface was developed to transmit messages precisely at pre-defined points in time. All these aspects were individually evaluated and then combined in a real-time capable TSN network stack.

7.4.10 TSN Demonstrator

Bachelor Thesis [149] – Christoph Lehr

The goal of this thesis is to evaluate the current state of Time-Sensitive Networking (TSN) related standards and to build a demonstration network. First, the required technical background, an introduction to general topics in TSN, and the contributing standards are provided. A survey on a set of manufacturers was conducted to show the current state of available products on the market.

7.4.11 NETCONF Server for FSS Devices

Bachelor Thesis [150] – Nora Hartner

The aim of this thesis is to create a proof-of-concept implementation of a NETCONF server needed for the configuration of FSS devices. NETCONF is used to discover, configure, and monitor the TSN network infrastructure of an FSS. Besides the NETCONF functionality of the TSN bridges, a NETCONF server is also required to discover and configure the end devices. This work focuses on implementing a NETCONF server for devices with small resources to read parameters from and write TSN configurations on the device.

CHAPTER 8

Architecture Evaluation

> NOW I HOLD THAT SCIENTIFIC THEORIES ARE NEVER FULLY JUSTIFIABLE OR VERIFIABLE, BUT THAT THEY ARE NEVERTHELESS TESTABLE. I SHALL THEREFORE SAY THAT THE OBJECTIVITY OF SCIENTIFIC STATEMENTS LIES IN THE FACT THAT THEY CAN BE INTER-SUBJECTIVELY TESTED. [151]
>
> – Karl Popper, 1935

Along the pathway of the Flexible Safety System Development Life Cycle (FSS-DLC) illustrated in Figure 3.2, the architecture evaluation phase constitutes an important milestone towards a proof-of-concept implementation of a Flexible Safety System (FSS). Therefore, this chapter evaluates the software architecture model from the previous phase using the use case definitions and requirements from the analysis phase. As an outcome, this phase produces overall evaluation results for the software architecture.

Before implementing the system design and software architecture of an Safety Network Controller (SNC), it is essential to evaluate if the architecture can meet all the requirements stated in Section 4.3 early in the system's life cycle. Therefore, a suitable methodology must be chosen to determine the quality attribute properties and trade-offs of a complex system [152].

The results of the architecture evaluation phase can serve as a basis for deciding whether the system design and software architecture can be implemented in the development phase or still need improvements.

8.1 Reasons for Architecture Evaluation

A software architecture relates to the organization of a system described in terms of its components, their externally visible properties, and the relationships among them. The architecture determines or precludes nearly all of the system's quality attributes, making it possible to evaluate architectural decisions concerning their impact on those attributes.

There are several reasons to perform an architecture evaluation before starting implementation, the most prominent being the time, cost, and effort spent to solve problems later in the FSS-DLC. The effect of the architecture on the system is profound. Therefore, an unsuitable architecture can precipitate failure to meet the performance, security, or economic goals. Changes that could have been anticipated and planned for in the later stages of the product life cycle will be rejected because they are too costly.

Architecture also determines the structure of a project, including schedules, budgets, performance goals, team structure, documentation organization, test plan, and maintenance activities, which all are organized around the architecture. Consequently, if the architecture is modified midstream because of some deficiency discovered late, it can significantly damage the entire project. It is much more efficient to adapt the architecture before it has been frozen by the establishment of downstream artifacts based on it [153].

Architecture evaluation is a relatively cheap way to avoid costly problems or even disasters. Performing an architecture evaluation can start even before the architecture is fully specified and can utilize different methods, which will be discussed next.

8.2 Evaluation Method Selection

Various ways exist to evaluate a software architecture depending on the situation and the existing artifacts. Generally, software architecture evaluation methods can be divided into the four main categories listed in Table 8.1.

Category	Description
Scenario-based	evaluate a particular quality attribute by creating a scenario profile that forces a very concrete description of the quality requirement
Mathematical model-based	uses mathematical proofs and methods for evaluating mainly operational quality requirements such as performance and reliability
Experience-based	evaluations are based on the previous experience and domain knowledge of developers or consultants
Simulation-based or Tool-based	rely on a high level implementation of some or all of the components in the software architecture. A simulation can then be used to evaluate quality requirements such as performance and correctness of the architecture

Table 8.1: Categories of software architecture evaluation methods

From the listed categories in Table 8.1, the scenario-based method is suitable for evaluating the SNC software architecture since the use cases with their scenarios from Section 4.2 can be transformed into scenarios for the evaluation. Table 8.2 lists the most common from several scenario-based methods, summarized from conducted surveys by Patidar et al. [154] and Shanmugapriya et al. [155].

Method	Acronym	Goal
Software Architecture Analysis Method	SAAM	Architectural suitability and risks analysis
Architecture Tradeoff Analysis Method	ATAM	Sensitivity and tradeoff analysis
Architecture-Level Modifiability Analysis	ALMA	Maintenance cost prediction, risk assessment, architectures comparison
Cost-Benefit Analysis Method	CBAM	Provide business measures for particular system changes. Make explicit the uncertainty associated with the estimates
Software Architecture Comparison Analysis Method	SACAM	Comparing software architectures from different domains

Table 8.2: Scenario-based software architecture evaluation methods

From these scenario-based methods, the Software Architecture Analysis Method (SAAM) seems suitable for FSS due to its simplicity. In practice, SAAM has proven useful for quickly assessing many quality attributes such as modifiability, portability, extensibility, integrability, as well as functional coverage. The method can also be used to assess quality aspects of software architectures such as performance or reliability [156].

Kazman et al. [157] first proposed SAAM in 1994 to evaluate user interface architectures with respect to modifiability and to compare competing software architectures. Three perspectives are defined for understanding and describing architectures: functionality, structure, and allocation. After describing the architecture, the SAAM is applied using the following main activities:

1. Characterize a canonical functional partitioning for the domain.
2. Map the functional partitioning onto the architecture's structural decomposition.
3. Choose a set of quality attributes with which to assess the architecture.
4. Choose a set of concrete tasks which test the desired quality attributes.
5. Evaluate the degree to which each architecture provides support for each task.

Originally, the SAAM was created to analyze an architecture for modifiability in its various forms and names. In practice, over the years, SAAM has proven useful for quickly assessing many quality attributes such as modifiability, portability, extensibility, integrability, and functional coverage. SAAM guides the inspection of the architecture, focusing on potential problem areas such as requirement conflicts or incomplete design specification from a particular stakeholder's perspective [158].

In 2002, Clements et al. [153] named three reasons why software architecture is important to large, complex, software-intensive systems. First, it is a vehicle for communication among stakeholders. Second, it is the manifestation of the earliest design decisions. Third, it is a reusable, transferable abstraction of a system. The authors present a refined version of SAAM defining inputs, outputs, and six steps for the SAAM evaluation. Figure 8.1 illustrates these steps of the SAAM evaluation process.

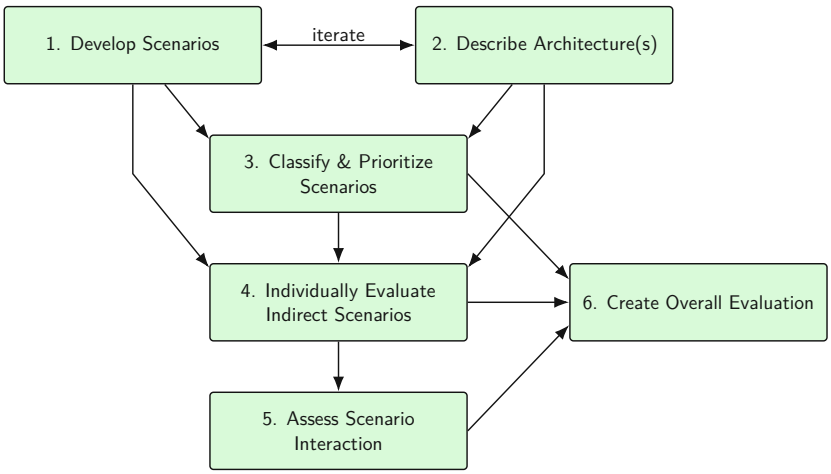


Figure 8.1: Activities and dependencies in a SAAM analysis

At the beginning of a SAAM analysis, some inputs in the form of descriptions of the architecture and a number of scenarios for understanding and evaluating the software architecture are required. Besides the enhanced communication among the participating stakeholders, the main outputs are a mapping of the architectures onto the scenarios that represent possible future changes to the system and an understanding of the functionality of the system or even a comparison of multiple architectures. Evaluating a single architecture using SAAM can indicate places where that architecture fails to meet its modifiability requirements or prominent alternative designs that would work better. If two or more candidate architectures should be compared or benchmarked, the SAAM can produce a relative ranking indicating which architecture satisfies its quality requirements more fully with less modification and with less resulting future complexity.

The activities in the SAAM process start with **developing scenarios** that represent the stakeholders' interests. In doing so, three kinds of scenarios can be differentiated: use case scenarios, growth scenarios, and exploratory scenarios. Use case scenarios represent how the stakeholders, in the role of an end user, expect the system to be used. Growth scenarios represent ways in which the architecture is expected to accommodate growth and change in the near-term, such as modifications, changes in performance, or integration with other software. Exploratory scenarios represent extreme forms of growth where the architecture might be stressed exceptionally by significant changes in the infrastructure or mission of the system or by dramatic new performance or availability requirements.

The next step is to **describe the candidate architecture** including the static and dynamic behavior. The static representation indicates the system's computation and data components as well as all relevant connections, whereas the dynamic representation describes how the system behaves over time. The development of scenarios and the architecture description drive each other and, therefore, are usually done in an interleaved way or in several iterations.

The third step is intended to **classify and prioritize the scenarios**. The scenarios are classified into direct and indirect scenarios, following a prioritization of those to allow the most important scenarios to be addressed within the limited amount of time available for the evaluation. Direct scenarios are those that are satisfied by the architecture through the execution of the system, while an indirect scenario is one that requires a modification to the architecture to be satisfied.

After a set of scenarios has been chosen for consideration, these scenarios are mapped onto the architectural description in the fourth step. For direct scenarios, it is demonstrated how the architecture would execute the scenario, while for indirect scenarios, it must be described how the architecture would need to be changed to accommodate that scenario. A modification to the architecture means that either a new component or connection is introduced or an existing component or connection requires a change in its specification. Therefore, it is necessary to **individually evaluate indirect scenarios**, which usually also includes an estimate of the cost of performing the change.

Scenarios interact in a component when two or more indirect scenarios require changes to a single component of an architecture. The purpose of step five is to **assess scenario interactions** of semantically unrelated scenarios explicitly for two reasons. First, to show which components are computing semantically unrelated functions. Second, it reveals whether the architecture is documented to the right level of structural decomposition.

Finally, to **create the overall evaluation,** each scenario is assigned a particular weight depending on its relative importance to the success of the system at hand. This weighting, which can be based on anticipated cost, risk, time to market, or some other agreed-upon criterion, is especially helpful to determine an overall ranking if multiple architectures are being compared and scrutinized or if different architectural alternatives are being proposed for a single architecture.

In summary, the SAAM is an architecture evaluation method that was created to operationalize the vague claims of modifiability, robustness, and portability that people typically make by replacing claims of quality attributes with scenarios that operationalize those claims. Following the SAAM approach facilitates an appropriate level of architectural documentation to be revealed and explained and to foster technical and social exchange among stakeholders.

8.3 SAAM Evaluation of the Software Architecture

Based on the results of the preceding chapters, the SNC software architecture is evaluated by applying the previously described SAAM procedure. Figure 8.2 illustrates the utilized evaluation process and the preceding activities.

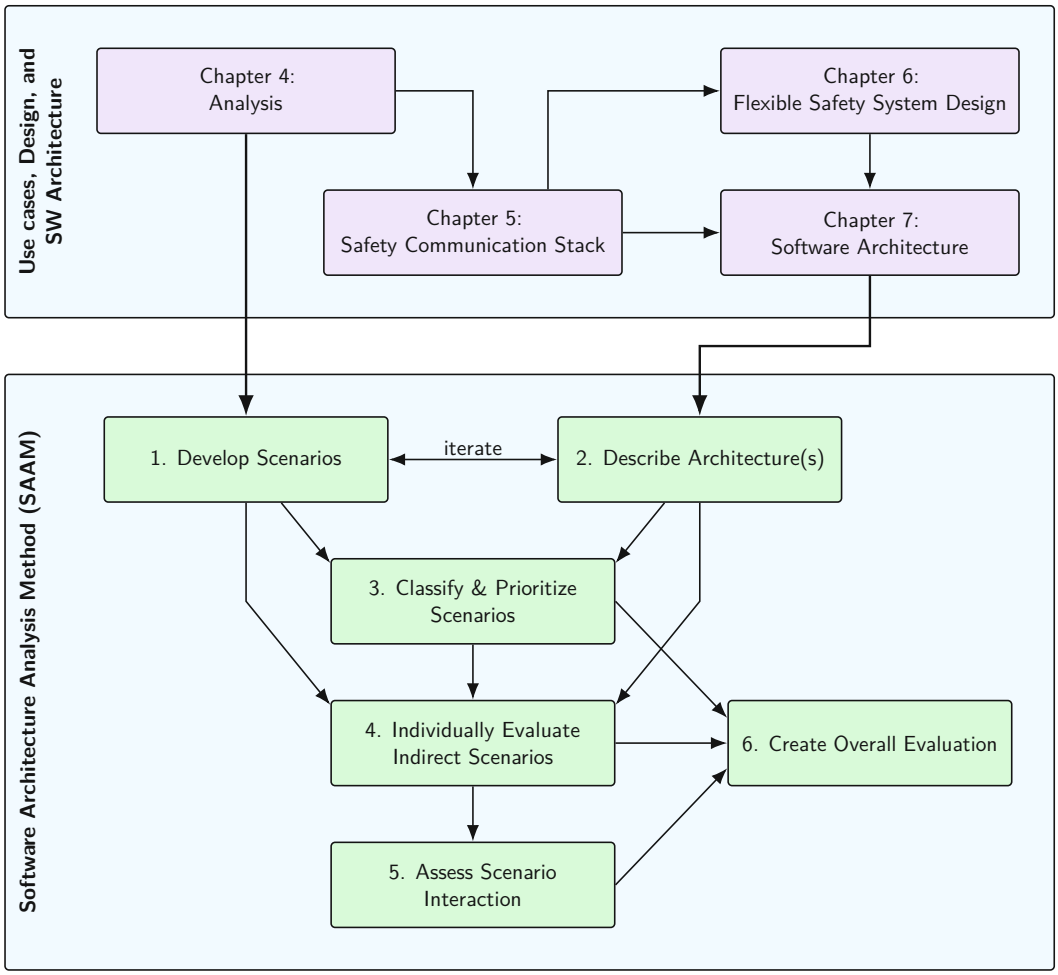


Figure 8.2: SNC software architecture evaluation process model

This evaluation aims to find out if the proposed FSS design in Chapter 6 and the derived software architecture in Chapter 7 can fulfill the requirements of the described use cases in Chapter 4. Figure 8.2 shows the workflow prior to the evaluation, which provides the inputs to the evaluation and the evaluation process. According to the SAAM procedure, the evaluation is carried out in six steps, which will be described in the following.

8.3.1 Step 1: Develop Scenarios

A scenario is a short statement describing an interaction of one of the stakeholders with the system. The scenarios are derived from the use cases and requirements, which were compiled in the use case analysis in Section 4.2 and the requirements determination in Section 4.3.

Table 8.3: FSS evaluation scenarios

	Scenario Number	Scenario Description	Requirement
Technology	01	Configure a safety link between two safety-relevant devices and transfer safety-relevant messages between those.	FR-06, FR-07, FR-08, FR-09, FR-10
	02	Configure a deterministic real-time data transmission stream between two network end devices.	FR-06
	03	Safety-relevant messages are exchanged between devices of different vendors.	FR-12
	04	Safety communication uses black channel principle and is certified.	FR-08, FR-09
	05	All data is transferred using one communication network.	NR-14
	06	Safety devices describe, advertise, and discover their capabilities.	FR-01, FR-02
	07	Use data encryption/decryption on all communications.	FR-05, NR-11
	08	Use the client/server communication model for configuration, monitoring, and logging.	FR-03, FR-07
	09	Use the publish/subscribe communication model for safety process data between safety devices.	FR-04
All use cases	10	Individual machines or manufacturing cells are operated by workers (non-experts) or an orchestration system.	OR-02, NR-06, NR-07
	11	Individual machines or manufacturing cells are configured by a safety engineer or domain experts.	OR-06, OR-10, NR-08
	12	User identification and authentication for elevated access.	FR-18
	13	Change full or partial safety configuration of the whole production system or sub-system when system is in safe state.	OR-09
	14	Seamless change of safety configuration when system is in operation.	FR-11
	15	Read and modify an existing safety configuration depending on the access level.	OR-10
	16	Add a new or remove an existing safety configuration.	OR-10
	17	A new network or safety device is discovered automatically and shown with its capabilities in the GUI to be used by the safety engineer.	FR-01, FR-02, FR-14, NR-03, NR-04
	18	Add a new and remove an existing network device or network link manually in an existing safety configuration.	OR-10, FR-15
	19	Add a new and remove an existing safety device or safety link manually in an existing safety configuration.	OR-10, FR-15

(Continued on next page)

Table 8.3 – continued from previous page

	Scenario Number	Scenario Description	Requirement
All use cases	20	Deploy predefined safety configurations on user request.	FR-21, NR-05
	21	On discovered changes in the production cell a new safety configuration is suggested which can be activated.	FR-16
	22	The system can be switched in maintenance mode by the safety engineer with proper indication for all other users.	OR-07, OR-10, FR-18, FR-28
	23	A configuration is modified using one engineering tool.	OR-10, FR-13, FR-18
	24	Create a safety function with existing devices and suggestions from the system.	FR-22
	25	Generate a set of TSN network configuration files needed for deployment.	FR-23
	26	Generate a set of safety device parameter configuration files needed for deployment.	FR-23
	27	Tagging of a safety configuration as verified in the GUI with the adequate access level of a safety engineer.	FR-24
	28	Verify a safety configuration in progress against relevant safety regulations.	FR-25
	29	Tagging of a safety configuration as validated in the GUI with the adequate access level of a safety engineer.	FR-26
	30	Perform a validation procedure of a deployed safety configuration guided by an assistive tool in the GUI.	FR-27
	31	Write an entry in the change log on each configuration change.	FR-17
	32	Display the entries of the change log with filter options on the GUI.	FR-17
	33	Export the change log with filter options.	FR-17
UC-01 Plug-and-Produce	34	A machine operator switches the machine to "manual mode" via the machine's HMI or any other suitable device that supports identification and authentication. Consequently, the safety configuration is adapted, and the user gets feedback about the outcome.	OR-01, OR-02, OR-03, OR-04, OR-09, FR-11, FR-18, FR-21, NR-05
	35	A machine operator switches the machine in "semi-automatic mode" via the machine's HMI, the control station, or any other suitable device that supports identification and authentication. Consequently, the safety configuration is adapted, and the user gets feedback about the outcome.	OR-01, OR-02, OR-03, OR-04, OR-09, FR-11, FR-18, FR-21, NR-05
	36	A machine operator switches the machine to "full-automatic mode" via the control station or any other suitable device that supports identification and authentication. Consequently, the safety configuration is adapted, and the user gets feedback about the outcome.	OR-01, OR-02, OR-03, OR-04, OR-09, FR-11, FR-18, FR-21, NR-05

(Continued on next page)

Table 8.3 – continued from previous page

	Scenario Number	Scenario Description	Requirement
UC-02 Flexible Manufacturing Cell	37	An operator of a manufacturing cell switches the machine to "manual operation" via the control station or any other suitable device that supports identification and authentication. Consequently, the safety configuration is adapted, and the operator gets feedback about the outcome. The machine operators or workers in the cell get information about the current configuration and state.	OR-01, OR-02, OR-03, OR-04, OR-09, FR-11, FR-18, FR-21, NR-05
	38	An operator of a manufacturing cell switches the machine to "semi-automatic operation" via the control station or any other suitable device that supports identification and authentication. Consequently, the safety configuration is adapted, and the operator gets feedback about the outcome. The cell indicates its configuration, safety areas, and state to the machine operators or workers in the cell.	OR-01, OR-02, OR-03, OR-04, OR-09, FR-11, FR-18, FR-21, NR-05
	39	An operator of a manufacturing cell switches the machine to "full-automatic operation" via the control station or any other suitable device that supports identification and authentication. Consequently, the safety configuration is adapted, and the operator gets feedback about the outcome. The cell indicates its configuration and state to bystanders. No machine operators or workers are in the cell.	OR-01, OR-02, OR-03, OR-04, OR-09, FR-11, FR-18, FR-21, NR-05
UC-03 Mobile Devices	40	The operator of a mobile device is notified about the possibility of connecting the device to the safety configuration of that cell.	FR-01, FR-02, FR-29, NR-03
	41	The operator of a mobile device acknowledges the involvement of the emergency stop button of that device in the safety configuration of a manufacturing cell. After the successful reconfiguration, the emergency stop button is illuminated to indicate the function to the user.	FR-12, FR-13, NR-05
	42	The operator of a mobile device is notified when leaving the cell. The emergency stop button is disconnected from the safety system of the manufacturing cell, and the illumination is disabled.	FR-29, NR-04
UC-04 Container-Based	43	The cell operator configures a container safety application on a host. The application is discovered and displayed on the GUI.	FR-01, FR-02, FR-29, FR-30, FR-32, NR-03
	44	The cell operator changes the configuration of a container safety application on a host. The application is rediscovered and updated on the GUI.	FR-01, FR-02, FR-29, NR-03
	45	The cell operator transfers a container safety application from one to another host. The application is discovered and updated on the GUI. All safety-related links and network connections are updated and transferred as well.	FR-01, FR-02, FR-29, FR-30, FR-31, FR-32, NR-03
	46	The cell operator removes a container safety application from a host. The application is removed from the GUI.	FR-29, FR-30, FR-32, NR-04

8.3.2 Step 2: Describe Architecture

The software architecture of the SNC is presented in detail in Chapter 7 using the C4 methodology. The four levels of this methodology address context, containers, components, and code. It, therefore, can be directly used for the SAAM evaluation.

8.3.3 Step 3: Classify and Prioritize Scenarios

In this step, the scenarios in Table 8.3 are classified into direct and indirect scenarios and prioritized according to their importance for the project. The result of this classification and prioritization is listed in Table 8.4.

Priority	Scenario Number	Class
01	01	direct
02	02	direct
03	03	direct
04	04	direct
05	05	direct
06	06	direct
07	08	direct
08	09	direct
09	10	direct
10	11	direct
11	12	direct
12	13	direct
13	15	direct
14	16	direct
15	17	direct
16	18	direct
17	19	direct
18	20	direct
19	25	direct
20	26	direct
21	27	direct
22	28	direct
23	29	direct

Priority	Scenario Number	Class
24	30	direct
25	31	direct
26	32	direct
27	24	direct
28	33	indirect
29	21	direct
30	14	direct
31	22	indirect
32	07	direct
33	23	direct
34	34	direct
35	35	direct
36	36	direct
37	37	direct
38	38	direct
39	39	direct
40	40	indirect
41	41	indirect
42	42	indirect
43	43	indirect
44	44	indirect
45	45	indirect
46	46	indirect

Table 8.4: Scenarios prioritization and classification

At this point, it can be determined that the majority of scenarios are direct, meaning that the proposed software architecture directly supports the scenario. Only nine scenarios are indirect, meaning that the software architecture needs modification in order to support those scenarios. The indirect scenarios are mainly those identified during the project work and, therefore, still need to be considered in the architecture.

8.3.4 Step 4: Individually Evaluate Indirect Scenarios

The indirect scenarios require modification or an extension of the software architecture, which could mean slight modification of services or components but could also lead to the extension of the architecture with additional services and components. Table 8.5 lists all indirect scenarios and the required change.

Scenario Number	Required Change	Number of Changed/Added Components	Effort Estimate
22	A maintenance mode has to be added to the "Semantic Information Storage" component in the "Knowledge Representation" container. In the "Visualization & Modification" container, a "Maintenance" component has to be added.	2	medium
33	To export the change log, the "Visualization & Modification" container has to be extended by an "Export" component. Existing interfaces and data structures can be used.	1	low
40, 41, 42	The handling of mobile devices requires an extension of the "Semantic Information Storage" component in the "Knowledge Representation" container, a new "Mobile Devices" component in the "Visualization & Modification" container, a change in the "Configuration Deployment" component in the "Deployment" container, and a change in the "Device Parameters Generation" component in the "Configuration Generation" container.	4	high
43, 44, 45, 46	For container-based applications, the architecture has to be extended by adding a "Container Applications" component in the "Visualization & Modification" container, extending the "Semantic Information Storage" component in the "Knowledge Representation" container, and adding a "Container Deployment" component in the "Deployment" container.	3	high

Table 8.5: Evaluation of indirect scenarios

8.3.5 Step 5: Assess Scenario Interaction

The aim of this step is to determine if two or more indirect scenarios require changes to a single component of an architecture. For this purpose, Table 8.5 from the previous step is analyzed to point out which architecture components are computing semantically unrelated functions. One central component that has to be modified in all indirect scenarios is the "Semantic Information Storage" component in the "Knowledge Representation" container. However, this does not indicate that this component computes semantically unrelated functions, but this is because the semantic information storage stores all systems knowledge. The "Visualization & Modification" container has to be extended in all indirect scenarios with additional components, which shows no interactions among these scenarios. Also, the modification of the "Deployment" and "Configuration Generation" containers does not reveal any interactions among indirect scenarios.

8.3.6 Step 6: Create Overall Evaluation

The overall evaluation summarizes the outcomes of the subsequent steps three to five. Depending on the purpose of the evaluation, the summary is either a comparison of architecture alternatives by weighting the candidates or a simple recap of the previous steps. As the goal of this evaluation was to scrutinize the proposed design in Chapter 6 and the derived software architecture in Chapter 7, no architecture alternatives are compared, but scenarios created to analyze if the proposed design and architecture meet the user's requirements and use cases.

The classification and prioritization of scenarios in step three revealed which scenarios are covered directly by the proposed architecture and which require additional changes to meet all requirements derived from the use cases. The prioritization of scenarios not only helps to identify the most important scenarios to be addressed within the limited amount of time available for the evaluation but also gives the development team some guidance to organize the software architecture development.

The evaluation of indirect scenarios has shown that the nine identified indirect scenarios can be aggregated into four groups. These groups require different changes or extensions of the software architecture, which were described and estimated.

Finally, the assessment of scenario interaction did not uncover any architecture components computing semantically unrelated functions. The finding of such components would reveal potentially poor separation of concerns in one component, indicating where the designer should focus subsequent attention.

8.4 Potential Weaknesses and Shortcomings

For the matter of completeness, it must be mentioned that the evaluation conducted in this chapter can contain some potential weaknesses and shortcomings. These can originate from the circumstance that use cases and scenarios might be incomplete, and therefore, several requirements still need to be discovered. Furthermore, an inherent bias in developing the scenarios could distort the analysis results and, therefore, affect the system design, the software architecture, or the evaluation.

In order to address these potential threads, which could lead to false evaluation results, the evaluation should be repeated with a broader group of stakeholders after the proof-of-concept implementation phase and before starting the development phase.

CHAPTER 9

Conclusion and Future Work

> Überhaupt hat der Fortschritt das an sich, dass er viel grösser ausschaut, als er wirklich ist. [159]
>
> – Johann Nepomuk Nestroy, Der Schützling, 1847

When work on this thesis began, there were many unanswered questions and uncertainties about the feasibility of a safety system that combines flexibility, interoperability, modularity, easy handling, and simple operation while at the same time complying with existing safety norms, standards, and regulations.

The starting point for the Flexible Safety System (FSS) project was the fact that the increasing complexity of machinery in smart factories has led and, in the future, will lead even more to very complex and time-consuming safety configurations. The current static approach of designing and configuring safety-relevant systems, including their certification, impedes the flexibility that machinery in the context of Industry 4.0 (I4.0) requires. In order to underline this, several use cases were presented to show the significance and relevance of flexibility, interoperability, modularity, and rapid reconfiguration in smart manufacturing production facilities.

An attempt has been made to show the challenges at hand and to lay the foundations for a new architecture to address these challenges. The following summarizes which goals have been accomplished so far in the endeavor to incorporate functional safety with flexibility, interoperability, and rapid reconfiguration. Furthermore, the next steps towards realizing an FSS are described, and several ideas and concepts for future work are presented to improve and facilitate the workflow for operators and safety engineers.

9.1 Achieved Outcomes

The primary objective of this work is related to the topics of flexibility, interoperability, and rapid reconfiguration of functional safety systems in I4.0 production facilities and the potential contradictions between those. In order to tackle these issues methodically, research questions were formulated, and an adapted systems development life cycle was defined. Subsequently, requirements of functional safety systems as part of I4.0 production facilities were identified and analyzed with the help of representative use cases. From the generic structure of a functional safety system, which is determined in principle by safety standards, a safety communication platform is derived that fulfills the previously determined requirements. Based on the findings in the analysis and the presented safety communication platform, the design and architecture for an FSS was proposed and evaluated.

9.1.1 Research Results

The research on the topic started with a literature review to substantiate the problem statement and to get answers to the first research question.

RQ1: Which properties are essential for a safety communication platform that enables flexibility and interoperability of Industry 4.0 production facilities?

The answers to this question were derived from the literature review in Section 1.2, the state-of-the-art analysis in Chapter 2, and the use case analysis in Section 4.2, which led to the following findings.

Operational principles and design of Reconfigurable Manufacturing Systems (RMSs), the foundation for production facilities in I4.0, were defined and formulated in 1995. These principles addressed the core characteristics of ideal RMSs. Since 2011, with the advent of the Internet of Things (IoT) and the introduction of I4.0, much literature has been published to define the characterizing features of I4.0. The three common characteristic features are network collaboration or interoperability, flexibility, and real-time capability. RMSs can be used to set up I4.0 production facilities that can quickly adjust their functionalities and production capacity in response to sudden market changes by allowing them to apply rapid changes in their structure and configuration. These properties help manufacturing companies, especially in Europe, to remain globally competitive by increasing the productivity of their production facilities.

However, all those measures to increase productivity do not take the mandatory safety system into account, which, by its static and unchangeable nature, impedes the flexibility that smart manufacturing production facilities require. In the area of reconfigurable functional safety systems that could enable comprehensive reconfiguration of production facilities, little research has been done so far. Several approaches were proposed, such as modular certification, continuous safety performance status demonstration, or worker

assistance systems, but none of those approaches offer a comprehensive solution for a flexible, reconfigurable, and interoperable safety system.

As an intermediate step on the way to designing a safety system that supports flexibility, interoperability, and rapid reconfiguration, existing technologies that fulfill the requirements had to be investigated and examined. The three base technologies Time-Sensitive Networking (TSN), OPC Unified Architecture (OPC UA), and OPC UA Safety emerged during this process. These technologies were combined into a safe communication platform supporting deterministic data transmission, interoperable information exchange, and safety-related communication in Chapter 5.

After justifying the problem statement, the requirements and prerequisites for future functional safety systems had to be worked out by first formulating and subsequently answering the second research question.

RQ2: What are the technical prerequisites for management and configuration that allow rapid reconfiguration of future functional safety systems?

An analysis was carried out to answer this question, starting with an abstract system model of a production system with its components and their relationships in Section 4.1. Then, representative use cases with their scenarios were defined in Section 4.2. Based on the model and the use cases, requirements were determined in Section 4.3. The generic system structure of functional safety systems complements the analysis in Section 4.4.

The abstract system model includes the structure and the relevant components of a generic production system and the safety system embedded into it. Additionally, characterizing properties of safety function, safety devices, safety communication, and safety configuration were specified to complete the model. To gain an understanding of the production processes and workflows, use cases and scenarios in the domain of discrete manufacturing were compiled with a focus on the convenience of use, rapid reconfiguration, and machinery safety. In this way, the range of applications covers plug-and-produce manufacturing cells, flexible manufacturing cells, mobile devices with safety features, and container-based safety applications. Finally, a requirements determination considers the system described in the model and the use cases and derives requirements that are categorized into operations, functions, and performance.

The four parts analyzed, abstract system model, use cases, requirements determination, and functional safety system structure, answer questions concerning the technical prerequisites for future functional safety systems and form the basis for the subsequent steps of defining a design and architecture for a system that meets all requirements. In the context of the proposed safety communication platform consisting of three base technologies, particular consideration was given to the need for a comprehensive safety configuration tool due to the complexity of the setup of a safety configuration. The establishment and removal of safety connections, which are the basic building blocks for a safety system configuration, require assistance to allow users to complete a configuration task quickly

and safely. The objectives of modularity, flexibility, and ease of use are addressed by suggesting essential services such as discovery, deployment, knowledge representation, configuration generation, and visualization interconnected with a well-defined Application Programming Interface (API) to serve as a starting point for the system design.

Grounded in the analysis of use cases and requirements of future production systems and based upon an proposed safety communication platform in Chapter 5, the design of an FSS was tackled in Chapter 6. Then, a software architecture was developed in Chapter 7, followed by an evaluation of the architecture in Chapter 8, which aimed at finding answers to the third research question.

RQ3: Which components and specific features are needed for an architecture that enables a functional safety system for future production facilities?

To answer this question, taking different views on the topic is necessary. The first view is on a physical system that provides the necessary capabilities to fulfill flexible, interoperable, and reconfigurable safety-related communication. The proposed safe communication platform combines three existing base technologies to enable a deterministic information exchange of safety-related messages. However, introducing three technologies also entails a very complex configuration effort. Therefore, a second view, the user perspective, is needed. For good usability and simple operation, the software architecture has been designed with the needs of operators and workers in mind. For this reason, many of the services aim to assist the user and simplify the safety configuration process. The assistance aspect applies, for example, to the discovery and deployment services, which automatically generate a physical system model and transfer configurations in the correct order with the proper timing to the affected devices. Configuration generation and plausibility check services also serve the purpose of assistance. Besides the goal of rapid reconfiguration, all these services aim to reduce engineering and operating efforts and decrease machine downtime.

9.1.2 Design and Architecture

The tangible outcome of this work is a system design and a software architecture for an FSS that can serve for a prototype implementation and the subsequent product development of such a system. The system design and, consequently, the software architecture was created based on Tim Berners-Lee's design principles as described and set out in Section 3.2.

Simplicity is addressed by assistive services whose task is to hide details and complexity of the used base technologies and to allow users to interact smoothly and efficiently with the system. The system is broken into service groups and services with relatively closely bound features interconnected with a well-defined API to fulfill the modular design principle. Each part of the architecture (e.g., features, services, service groups, system segments, production system) is considered part of a bigger system and part of

a modular design, which is the third design principle. Safety standards leave scope for interpretation in order to enable well-functioning production systems. An FSS accepts not precisely defined commands and creates configurations that strictly conform to the defined specifications and, therefore, conform to the principle of tolerance. The FSS design considers that no single common point limits how the system scales or produces a single point of complete failure. Following the principle of decentralization, the Safety Network Controller (SNC) is designed so that all its components can be hosted as desired in arbitrary places within the system. One of the main objectives of the FSS design is interoperability, which directly relates to the design principle of test of independent invention. Interoperability and the use of existing technology pervade the entire system design, including the interfaces exposed to other systems, such as orchestration systems. Only the least powerful means or resources were chosen, including the base technologies. For instance, in the case of TSN, only a subset of all available standards and features are used to fulfill the functionalities required for the deterministic transfer of safety-related data. In the case of internal communication among services, a common language SPARQL Protocol and RDF Query Language for querying and updating data, and a unified and lean framework based on the remote procedure call framework gRPC with Protobuf for exchanging events between services.

9.1.3 Registered Trade Mark

In the course of the work on FSS, a research project with the same name was established in the context of the Common Research Program (CRP) within the Center for Digital Production (CDP) and with the involvement of industrial partners.

As part of CDP's Intellectual Property Rights (IPR) strategy, the trade mark "CONFISAFE FLEXIBLE SAFETY SYSTEMS" was filed in Austria (Österreichisches Patentamt)[1] and Europe (EUIPO)[2] as figurative mark containing word elements, depicted in Figure 9.1, with the nice classes[3] 9 and 42.

Figure 9.1: FSS Logo

ConfiSafe provides a structure and services for a functional safety system that can be quickly and easily reconfigured before and during machine operation. The design utilizes existing vendor-neutral base technologies for safe and reliable communication and

[1] https://see-ip.patentamt.at/MarkeSuche/Details/ba9f9bfb-95d7-4e24-84d2-586c08094354
[2] https://euipo.europa.eu/eSearch/#details/trademarks/018863867
[3] https://euipo.europa.eu/ohimportal/en/nice-classification

provides services for device discovery, configuration generation, and automatic deployment. Another objective is assisting the operator in applying changes simply, quickly, and safely.

9.2 Reflection

At the beginning of this endeavor, the predominant reactions when presenting the idea of an FSS, as very often in the past, were: "it cannot be changed due to regulation" and "it has to be done in a certain way". Fortunately, in the course of the activities within the framework of FSS, these common myths were put into perspective by proposing innovative ways based on existing legislation and regulation. Most of the misconceptions could be attributed to the circumstance that safety is considered by the machine operator very often as a necessary, annoying, and unpleasant thing. It is seen as something obscure that cannot be touched and is usually not understood by the operator. Sometimes, this even leads to the suspension or deactivation of crucial safety measures for the purpose of enabling a specific workflow. Furthermore, changes in the safety configuration of a machine are associated with high financial and time efforts as it involves new safety certification as well as many highly qualified experts and a tedious process that leads to long downtime of the machine.

As the activities within the framework of FSS developed, the concerns regarding the feasibility of an FSS gave way to acceptance and led to affirmation and considerable interest. Some of the reasons for the change of attitude towards FSS were most likely the release of the specification for OPC UA Safety as one of the base technologies and that the first TSN switches became available. As a result of increased interest and attention, FSS evolved into a lighthouse project at the Center for Digital Production (CDP) with the participation of industry partners and accredited safety certification bodies.

Several events, such as workshops and partner meetings, were organized as part of the project to address all sorts of topics related to FSS. One of these workshops dealt with the topic of Intellectual Property Rights (IPR) to identify the potential for new ideas or patents. Some ideas were the automatic and continuous assessment of risk, the automatic deriving of knowledge from safety regulations and standards, a conformity check for safety configurations, a tool for the semantics of safety functions, and a safety protocol gateway. Another workshop's content was reconfiguring in the conformity process and tool qualification and certification. Issues such as "modification vs. placing new on the market" and "maximum flexibility vs. within a defined framework" were discussed among participants from universities, industry partners, and safety certification bodies. It could be concluded that there are more technical challenges than legal obstacles to enable flexible safety reconfiguration.

Since the beginning of the project, several scientific publications have been created. Due to the large scope of the project, these publications originated from various authors affiliated with TU Wien or CDP. They included eight bachelor theses, two master theses, and fifteen conference papers.

9.3 Next Steps

It is planned to test and verify the present concept for an FSS in a lab environment with a technology demonstrator and in a real production environment at the Pilotfabrik of TU Wien as described in Section 4.2.

9.3.1 Technology Demonstrator

Until recently, hardware equipment for TSN was unavailable, and the OPC UA Safety software stack, developed by the OPC Foundation, was only accessible as a beta version to co-financing members. For this reason, only system fragments could be considered, and only some code artifacts were created, including some bachelor and master theses as presented in the code part of the software analysis phase in Section 7.4.

Not long ago, the first TSN switches became available that are now object to tests regarding Network Configuration Protocol (NETCONF) and discovery features using Link Layer Discovery Protocol (LLDP). Also, not long ago, the OPC Foundation granted an academic license for the current OPC UA Safety software stack, and it is now object to tests concerning its configuration and first test implementations. With the resources at hand, a technology demonstrator should be created that combines a TSN network, open62541[4] an open source and free implementation of OPC UA, and the OPC Foundation's OPC UA Safety software stack. This demonstrator will also include self-programmed sensors, logic, and actuators due to the fact that safety-related devices supporting OPC UA Safety are currently unavailable, and it will still take some years until certified devices become available.

The goal of creating a technology demonstrator before the planned proof-of-concept implementation is to achieve a working setup of the proposed safe communication platform with a minimal and predictable effort to find possible design flaws early to make any required modifications.

9.3.2 Proof-of-Concept in Pilotfabrik

Once the technology demonstrator has proven its function and capabilities, a proof-of-concept implementation of the FSS is supposed to be implemented within the flexible production cell in the Pilotfabrik of TU Wien.

The successful implementation of an FSS as proof-of-concept for the proposed design and software architecture could initiate the development of such a system together with industry partners.

[4] https://github.com/open62541/open62541

9.4 Future Work

The design and architecture proposed in this work represent only the basis for a FSS. Many extensions and enhancements are imaginable and possible, especially in the context of assistive features. Only three concepts for possible extensions shall be briefly sketched in the following.

9.4.1 Machine-readable Semantic Safety Regulation Model

In order to provide machine operators or safety engineers with the best possible support during the operation or engineering process, it would be of great value if the system could provide a semantic model of the applicable safety regulations or standards. Such a model could be the basis for functionalities to support a safety engineer during the engineering process by pointing out mistakes or suggesting solutions. Furthermore, it could draw attention to potential safety issues during the operation of a machine.

9.4.2 Semi-Automatic Verification and Validation

Verifying and validating safety configurations are necessary steps in the safety configuration process. Therefore, assistance in these steps could help save significant amounts of time until a new safety configuration can be activated. Specifically, the semi-automatic verification of a configuration could make use of the previously presented machine-readable semantic safety regulation model.

9.4.3 AI-Supported Configuration Generation

For some time and due to increasing processing power, AI-supported tools have become a promising option in many areas. These new technologies could dramatically simplify the generation of configuration files needed to reconfigure the safety system.

Potential future directions of the proposed system design and software architecture include automatic compliance and conformance checking of safety configuration candidates. Also, legal and regulatory issues and the integration of local and international safety standards might be touched on. Such activities could facilitate features such as fully automatic configuration generation and certification. Furthermore, it could enable the fully automatic deployment triggered by a system change without human interaction, which is unthinkable for various valid reasons today. This thesis is just the beginning.

List of Figures

List of Tables

Acronyms

AGV Automated Guided Vehicle 19, 74, 76, 78, 80

AMQP Advanced Message Queuing Protocol 95

ANSI American National Standards Institute 27

API Application Programming Interface 106, 164

AVB Audio Video Bridging 92

CAEX Computer Aided Engineering Exchange 128

CCA Cause Consequence Analysis 35

CL Check List 35

CNC Centralized Network Configuration 92, 104, 118

CPPS Cyber-Physical Production System 5, 48

CPS Cyber-Physical System 15, 48

CUC Centralized User Configuration 93, 104, 118

DI Diagnostics Interface 101

EAC Eurasian Conformity 29

EEA European Economic Area . . . . . 25 f.

ERP Enterprise Resource Planning 110, 114, 133

ESPE Electrosensitive Protective Equipment . . . . . 22 f.

ETA Event Tree Analysis 35

FMEA Failure Mode and Effects Analysis 35

FSS Flexible Safety System . . vii, 14, 17, 54, 58, 64, 69, 103, 107–110, 112, 114, 116, 118–123, 126, 128–131, 133, 135, 140 f., 145, 147, 149, 151, 154, 161 f., 164–168

FSS-DLC Flexible Safety System Development Life Cycle 51, 53 ff., 79, 107, 135, 149 f.

FTA Fault Tree Analysis 35

GCL Gate Control List 91, 117, 131

GDS Global Discovery Server 97, 121, 125

GUI Graphical User Interface . . . . . 110, 130 f., 133, 143

HAZOP Hazard and Operability 35

HCI Human-Control Interface 56, 110, 133

HMI Human-Machine Interface . . . . . 74, 76, 82 f., 121 f., 133

I4.0 Industry 4.0 . . . . . 3–8, 10 ff., 14, 103, 161 f.

IA Industrial Automation 89

ICT Information and Communication Technology 3

IIoT Industrial Internet of Things 4

IoT Internet of Things 3, 30, 162

IT Information Technology 4

KBS Knowledge-Based System . . . . . 109 f., 121 f.

LDS Local Discovery Server 97, 121, 125

LLDP Link Layer Discovery Protocol 121, 124, 167

LOPA Layer of Protection Analysis 35

M2M Machine-to-Machine 11, 30, 48, 94, 119, 135, 141

M2MCI Machine-to-Machine-Control Interface 56, 110, 121, 133

MES Manufacturing Execution System 110, 114, 133

MIB Management Information Base 94

MQTT Message Queue Telemetry Transport 95

NETCONF Network Configuration Protocol . 94, 104, 109, 111, 121, 124 f., 127, 167

OEE Overall Equipment Effectiveness 7

OPC UA PI OPC UA Platform Interface 101

SysML Systems Modeling Language 49

TAS Time-Aware Shaper . . . . . 91, 105, 117 f., 131, 142

TSN Time-Sensitive Networking 46, 49, 86, 88, 91, 94, 103 ff., 108 f., 111, 116 ff., 121 f., 127, 131, 142, 145, 147, 163, 165, 167

WIA What-If Analysis 35

Bibliography

[1] W. L. Chenery, "Safety in the Ford Factory," *The Survey September 1920-March 1921*, vol. 45, p. 64, Oct. 1920. http://archive.org/details/surveysepmar1921surv.

[2] A. H. Maslow, "A theory of human motivation," *Psychological Review*, vol. 50, p. 370–396, 1943.

[3] S. M. Andersen, S. Chen, and C. Carter, "Fundamental human needs: Making social cognition relevant," *Psychological Inquiry*, vol. 11, no. 4, p. 269–275, 2000.

[4] D. Goetsch, *Occupational Safety and Health for Technologists, Engineers, and Managers*. What's New in Trades and Technology Series, Pearson, 2019.

[5] A. Meiklejohn, "Industrial health: Meeting the challenge," *British journal of industrial medicine*, vol. 16, p. 1–10, Jan. 1959.

[6] J. F. Witt, "The transformation of work and the law of workplace accidents, 1842-1910," *The Yale Law Journal*, vol. 107, no. 5, p. 1467–1502, 1998.

[7] J. Mokyr and R. H. Strotz, "The second industrial revolution, 1870-1914," *Storia dell'economia Mondiale*, vol. 21945, 1998.

[8] US Department of Labor, "Occupational safety and health act of 1970," *Public Law*, vol. 91, p. 596, 1970.

[9] Council of the European Communities, "Council directive 89/392/EEC of 14 june 1989 on the approximation of the laws of the member states relating to personal protective equipment," *Official Journal L*, vol. 392, no. 06/12, 1989.

[10] IEC 61508, "Functional safety of electrical/electronic/programmable electronic safety-related systems," standard, International Electrotechnical Commission, Geneva, CH, Apr. 2010.

[11] IEC 62061, "Safety of machinery - functional safety of safety-related control systems," standard, International Electrotechnical Commission, Geneva, CH, Mar. 2021.

[12] ISO 13849-1, "Safety of machinery - safety-related parts of control systems - part 1: General principles for design," standard, International Organization for Standardization, Geneva, CH, Dec. 2015.

[13] H.-C. Moon, J.-E. Chung, and S.-B. Choi, "Korea's manufacturing innovation 3.0 initiative," *Journal of Information and Management*, vol. 38, no. 1, p. 26–34, 2018.

[14] J. Schuldt and S. Friedemann, "The challenges of gamification in the age of industry 4.0: Focusing on man in future machine-driven working environments," in *2017 IEEE Global Engineering Education Conference (EDUCON)*, p. 1622–1630, 2017.

[15] S. Weyer, M. Schmitt, M. Ohmer, and D. Gorecky, "Towards industry 4.0 - standardization as the crucial challenge for highly modular, multi-vendor production systems," *IFAC-PapersOnLine*, vol. 48, no. 3, p. 579–584, 2015. 15th IFAC Symposium onInformation Control Problems inManufacturing.

[16] N. Velásquez Villagrán, E. Estevez, P. Pesado, and J. De Juanes Marquez, "Standardization: A key factor of industry 4.0," in *2019 Sixth International Conference on eDemocracy eGovernment (ICEDEG)*, p. 350–354, 2019.

[17] C. von Arnim, M. Drăgan, F. Frick, A. Lechler, O. Riedel, and A. Verl, "Tsn-based converged industrial networks: Evolutionary steps and migration paths," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, p. 294–301, 2020.

[18] D. Etz, T. Fruhwirth, A. Ismail, and W. Kastner, "Simplifying functional safety communication in modular, heterogeneous production lines," in *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, p. 1–4, IEEE, jun 2018.

[19] Y. Koren, X. Gu, and W. Guo, "Reconfigurable manufacturing systems: Principles, design, and future trends," *Frontiers of Mechanical Engineering*, vol. 13, p. 121–136, Jun 2018.

[20] D. Etz, P. Denzler, T. Frühwirth, and W. Kastner, "Functional safety use cases in the context of reconfigurable manufacturing systems," in *2022 27th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, p. 1–8, 2022.

[21] P. Hans-Christian, Y. Burak, and K. Tamer, "The impact of industry 4.0 on the supply chain," in *Innovations and Strategies for Logistics and Supply Chains: Technologies, Business Models and Risk Management. Proceedings of the Hamburg International Conference of Logistics (HICL), Vol. 20*, (Berlin), p. 31–58, epubli GmbH, 2015. 10419/209189; https://econpapers.repec.org/RePEc:zbw:hiclpr:20.

[22] A. Gilchrist, *Introducing Industry 4.0*, p. 195–215. Berkeley, CA: Apress, 2016.

[23] B. Vogel-Heuser and D. Hess, "Guest editorial industry 4.0–prerequisites and visions," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, p. 411–413, 2016.

[24] D. P. Perales, F. A. Valero, and A. B. García, "Industry 4.0: A classification scheme," in *Closing the Gap Between Practice and Research in Industrial Engineering* (E. Viles, M. Ormazábal, and A. Lleó, eds.), (Cham), p. 343–350, Springer International Publishing, 2018.

[25] H.-C. Pfohl, B. Yahsi, and T. Kurnaz, "Concept and diffusion-factors of industry 4.0 in the supply chain," in *Dynamics in Logistics* (M. Freitag, H. Kotzab, and J. Pannek, eds.), (Cham), p. 381–390, Springer International Publishing, 2017.

[26] H. Kagermann, R. Anderl, J. Gausemeier, G. Schuh, W. Wahlster, and J. Winter, *Industrie 4.0 in a Global Context: Strategies for Cooperating with International Partners (acatech STUDY)*. utzverlag, 11 2016.

[27] R. M. Nachiappan and N. Anantharaman, "Evaluation of overall line effectiveness (ole) in a continuous product line manufacturing system," *Journal of Manufacturing Technology Management*, vol. 17, p. 987–1008, Jan 2006.

[28] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Business & information systems engineering*, vol. 6, no. 4, p. 239–242, 2014.

[29] T. Sauter, S. Soucek, W. Kastner, and D. Dietrich, "The evolution of factory and building automation," *IEEE Industrial Electronics Magazine*, vol. 5, p. 35–48, Sep. 2011.

[30] Bitkom, VDMA, and ZVEI, "Umsetzungsstrategie industrie 4.0," Ergebnisbericht der Plattform Industrie 4.0, Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e. V. (Bitkom), Verband Deutscher Maschinen- und Anlagenbau e. V. (VDMA), Zentralverband Elektrotechnik- und Elektronikindustrie e. V. (ZVEI), Berlin, Frankfurt am Main, April 2015.

[31] W. Banzhaf, "Self-organizing systems," *Encyclopedia of Complexity and Systems Science*, p. 589–598, January 2009.

[32] P. O. Antonino, F. Schnicke, Z. Zhang, and T. Kuhn, "Blueprints for architecture drivers and architecture solutions for industry 4.0 shopfloor applications," in *Proceedings of the 13th European Conference on Software Architecture - Volume 2*, ECSA '19, (New York, NY, USA), p. 261–268, Association for Computing Machinery, 2019.

[33] P. Liggesmeyer and M. Trapp, *Safety in der Industrie 4.0*, p. 107–123. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017.

[34] T. Meany, "Functional safety and industrie 4.0," in *2017 28th Irish Signals and Systems Conference (ISSC)*, p. 1–7, June 2017.

[35] O. Jaradat, I. Sljivo, I. Habli, and R. Hawkins, "Challenges of safety assurance for industry 4.0," in *2017 13th European Dependable Computing Conference (EDCC)*, p. 103–106, Sep. 2017.

[36] J. Henriksson, M. Borg, and C. Englund, "Automotive safety and machine learning: Initial results from a study on how to adapt the iso 26262 safety standard," in *2018 IEEE/ACM 1st International Workshop on Software Engineering for AI in Autonomous Systems (SEFAIAS)*, p. 47–49, May 2018.

[37] M. Felser, M. Rentschler, and O. Kleineberg, "Coexistence standardization of operation technology and information technology," *Proceedings of the IEEE*, vol. 107, p. 962–976, June 2019.

[38] F. Pelzer, A. Klose, J. Miesner, M. Schmauder, and L. Urbas, "Safety in modular process plants: demonstration of safety concepts," *e & i Elektrotechnik und Informationstechnik*, 09 2021.

[39] F. Pelzer, A. Klose, M. Barth, H. Manske, R. Oehlert, S. León, A. Horch, J. Knab, B. Gut, R. Drath, and L. Urbas, "Intermodulare funktionale sicherheit für flexible anlagen - teil 1: Grundlagen und anforderungen," *atp magazin*, vol. 62, p. 84–92, 06 2020.

[40] F. Pelzer, A. Klose, R. Drath, A. Horch, S. León, H. Manske, C. Kotsch, R. Oehlert, J. Knab, M. Barth, B. Gut, and L. Urbas, "Intermodulare funktionale sicherheit für flexible anlagen der prozessindustrie - teil 2: Architektur und engineering intermodularer sicherheit und safety-mtp," *atp magazin*, vol. 62, 10 2020.

[41] F. Pelzer, A. Klose, J. Helmert, L. Urbas, and S. Pannasch, "Conducive design for safety in modular plants," in *Safety Management and Human Factors*, vol. 64, p. 125–134, 07 2022.

[42] A. Klose, F. Pelzer, M. Barth, R. Drath, R. Oehlert, S. León, A. Horch, C. Kotsch, J. Knab, B. Gut, H. Manske, and L. Urbas, "Distributed functional safety for modular process plants," in *2020 IEEE Conference on Emerging Technologies and Factory Automation (EFTA)*, 09 2020.

[43] M. Ehrlich, A. Bröring, D. Harder, T. Auhagen-Meyer, P. Kleen, H. Trsek, L. Wisniewski, and J. Jasperneite, "Alignment of safety and security risk assessments for modular production systems," *e & i Elektrotechnik und Informationstechnik*, 09 2021.

[44] S. Schreiber, T. Schmidberger, A. Fay, J. May, J. Drewes, and E. Schnieder, "Uml-based safety analysis of distributed automation systems," in *2007 IEEE Conference on Emerging Technologies and Factory Automation (EFTA 2007)*, p. 1069–1075, 2007.

[45] B. G. Mark, E. Rauch, and D. T. Matt, "Worker assistance systems in manufacturing: A review of the state of the art and future directions," *Journal of Manufacturing Systems*, vol. 59, p. 228–250, 2021.

[46] D. Etz, T. Frühwirth, and W. Kastner, "Self-configuring safety networks," in *Technologien für die intelligente Automation*, p. 232–245, Springer Berlin Heidelberg, 2020.

[47] D. Etz, T. Fruhwirth, and W. Kastner, "Flexible safety systems for smart manufacturing," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, p. 1123–1126, IEEE, sep 2020.

[48] A. Klose, F. Pelzer, D. Etz, D. Strutzenberger, T. Frühwirth, W. Kastner, and L. Urbas, "Building blocks for flexible functional safety in discrete manufacturing and process industries," in *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, p. 1–8, 2021.

[49] D. Etz, H. Brantner, and W. Kastner, "Smart manufacturing retrofit for brownfield systems," in *International Conference on Industry 4.0 and Smart Manufacturing (ISM 2019)*, vol. 42, p. 327–332, Elsevier BV, 2020.

[50] D. Etz and W. Kastner, "Human control of self-organizing safety systems," in *2020 2nd International Conference on Societal Automation (SA)*, p. 1–7, IEEE, May 2021.

[51] R. P. Feynman, "Appendix f: Personal observations on the reliability of the shuttle," *Report of the Presidential Commission on the Space Shuttle Challenger Accident*, vol. 2, p. F1–F5, 1986.

[52] J. Ridley and D. Pearce, *Safety with Machinery*. Butterworth-Heinemann, 2006.

[53] ISO 12100, "Safety of machinery - general principles for design - risk assessment and risk reduction," standard, International Organization for Standardization, Geneva, CH, Nov. 2010.

[54] O. Goernemann and H.-J. Stubenrauch, "Electro-sensitive protective devices (ESPE) for safe machines," tech. rep., SICK AG, Waldkirch, Germany, Aug. 2017.

[55] European Agency for Safety and Health at Work, "Directive 2006/42/ec - new machinery directive," 2021. Available online at: https://osha.europa.eu/en/legislation/directives/directive-2006-42-ec-of-the-european-parliament-and-of-the-council, last accessed on 20.02.2023.

[56] Council of the EU, "Council and european parliament agree on new safety requirements for machinery products," 2022. Available online at: https://www.consilium.europa.eu/en/press/press-releases/2022/12/15/council-and-european-parliament-agree-on-safety-requirements-for-machinery-products, last accessed on 20.02.2023.

[57] European Agency for Safety and Health at Work, "Directive 2009/104/ec – use of work equipment," 2021. Available online at: https://osha.europa.eu/en/legislation/directives/3, last accessed on 20.02.2023.

[58] C. Bittner, H. Bode, A. Christ, R. Gaiser, A. Hahn, J. Hasel, T. Klindt, M. Moog, A. Neudörfer, A. Schott, E. Sieber-Fazakas, K. Stark, J. Vetter, G. Wemmer, H. Wessels, M. Wimmer, and M. Wustlich, "The safety compendium," tech. rep., Pilz GmbH & Co. KG, Ostfildern, Germany, Aug. 2017. Available online at: https://www.pilz.com/mam/pilz/content/editors_mm/safety_compendium_en_2017_12_low.pdf, last accessed on 20.02.2023.

[59] Japan Quality Assurance Organization, "Application guide for conformity assessment based on electrical appliance and material safety law," Tech. Rep. 01745, Japan Quality Assurance Organization, aug 2021. Available online at: https://www.jqa.jp/english/safety/file/guide_pse.pdf, last accessed on 22.02.2023.

[60] S. Antonsen, K. Skarholt, and A. J. Ringstad, "The role of standardization in safety management – a case study of a major oil & gas company," *Safety Science*, vol. 50, no. 10, p. 2001–2009, 2012. Papers selected from 5th Working on Safety International Conference (WOS 2010).

[61] S. Timmermans and M. Berg, *The gold standard: The Challenge of Evidence-Based Medicine and Standardization in Health Care.* Temple University Press, 2003.

[62] B. Chiao, J. Lerner, and J. Tirole, "The rules of standard-setting organizations: an empirical analysis," *The RAND Journal of Economics*, vol. 38, no. 4, p. 905–930, 2007.

[63] K. Blind, A. Lorenz, and J. Rauber, "Drivers for companies' entry into standard-setting organizations," *IEEE Transactions on Engineering Management*, vol. 68, no. 1, p. 33–44, 2021.

[64] ISO/IEC GUIDE 51:2014, "Safety aspects – guidelines for their inclusion in standards," standard, International Organization for Standardization, Geneva, CH, Apr. 2014.

[65] ANSI B11.0-2020, "Safety of machinery," standard, American National Standards Institute, Houston, USA, Dec. 2019.

[66] ISO 13850, "Safety of machinery - emergency stop function - principles for design," standard, International Organization for Standardization, Geneva, CH, Nov. 2015.

[67] R. Preiss, *Risk Analysis Techniques in Engineering.* Brunn am Gebirge, Austria: TÜV Austria Academy, 2020.

[68] V. de Vasconcelos, W. A. Soares, A. C. L. da Costa, and A. L. Raso, "Chapter 2 - deterministic and probabilistic safety analyses," in *Advances in System Reliability Engineering* (M. Ram and J. P. Davim, eds.), p. 43–75, Academic Press, 2019.

[69] R. Apfeld, M. Hauke, and S. Otto, "Definition of safety functions: what is important?," tech. rep., Institut für Arbeitsschutz der Deutschen Gesetzlichen Unfallversicherung (IFA), Berlin, Germany, June 2015.

[70] ISO 13855, "Safety of machinery – positioning of safeguards with respect to the approach speeds of parts of the human body," standard, International Organization for Standardization, Geneva, CH, May 2010.

[71] IEC, "Functional safety – essential to overall safety," tech. rep., International Electrotechnical Commission, mar 2015.

[72] S. Brown, "Overview of IEC 61508. Design of electrical/electronic/programmable electronic safety-related systems," *Computing and Control Engineering*, feb 2000.

[73] IEC 61511, "Functional safety – safety instrumented systems for the process industry sector," standard, International Electrotechnical Commission, Geneva, CH, Jan. 2023.

[74] IEC 60204-1, "Safety of machinery – electrical equipment of machines," standard, International Electrotechnical Commission, Geneva, CH, Sept. 2021.

[75] H. Gall, "Functional safety iec 61508 / iec 61511 the impact to certification and the user," in *2008 IEEE/ACS International Conference on Computer Systems and Applications*, p. 1027–1031, March 2008.

[76] J. Münch, O. Armbrust, M. Kowalczyk, and M. Soto, *Software Process Definition and Management*. Springer Publishing Company, Heidelberg, May 2012.

[77] J. Akerberg, M. Gidlund, T. Lennvall, J. Neander, and M. Björkman, "Efficient integration of secure and safety critical industrial wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, 12 2011.

[78] M. C. Magro, P. Pinceti, and L. Rocca, "Can we use IEC 61850 for safety related functions?," in *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)*, p. 1–6, 2016.

[79] A. Elia, L. Ferrarini, and C. Veber, "Analysis of ethernet-based safe automation networks according to iec 61508," in *2006 IEEE Conference on Emerging Technologies and Factory Automation*, p. 333–340, Sept 2006.

[80] B. Wilamowski and J. Irwin, *Industrial Communication Systems*. ENGnetBASE 2015, CRC Press, 2016.

[81] N. Da Dalt and A. Sheikholeslami, *Understanding Jitter and Phase Noise: A Circuits and Systems Perspective*. Cambridge University Press, 2018.

[82] L. Dürkop, J. Jasperneite, and A. Fay, "An analysis of real-time ethernets with regard to their automatic configuration," in *2015 IEEE World Conference on Factory Communication Systems (WFCS)*, p. 1–8, May 2015.

[83] R. Schlesinger, A. Springer, and T. Sauter, "Concept for the coexistence of standard and real-time ethernet," in *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, p. 1–10, June 2018.

[84] P. Sanz, Óscar Seijo, M. Canto, J. Montalban, P. Angueira, and I. Val, "On the feasibility of wireless communications for safety applications in industry," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 3, p. 2432–2441, 2023.

[85] I. Val, Óscar Seijo, R. Torrego, and A. Astarloa, "Ieee 802.1as clock synchronization performance evaluation of an integrated wired–wireless tsn architecture," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, p. 2986–2999, 2022.

[86] A. Larrañaga, M. C. Lucas-Estañ, I. Martinez, I. Val, and J. Gozalvez, "Analysis of 5g-tsn integration to support industry 4.0," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, p. 1111–1114, 2020.

[87] IEC 61784-3-3, "Industrial communication networks - profiles - part 3-3: Functional safety fieldbuses - additional specifications for cpf 3," standard, International Electrotechnical Commission, Geneva, CH, Aug. 2010.

[88] PROFIBUS Nutzerorganisation, Karlsruhe, Germany, *PROFIsafe System Description - Technology and Application*, Apr. 2016.

[89] G. Peserico, A. Morato, F. Tramarin, and S. Vitturi, "Functional safety networks and protocols in the industrial internet of things era," *Sensors*, vol. 21, Sep. 2021.

[90] OPC UA Part 15: Safety, "OPC Unified Architecture Part 15: Safety," specification, OPC Foundation, Scottsdale, USA, Oct. 2019.

[91] R. Rezaei, T. K. Chiew, S. P. Lee, and Z. S. Aliee, "Interoperability evaluation models: A systematic review," *Computers in Industry*, vol. 65, no. 1, p. 1–23, 2014.

[92] Y. Liao, L. F. P. Ramos, M. Saturno, F. Deschamps, E. F. R. de Loures, and A. L. Szejka, "The role of interoperability in the fourth industrial revolution era," *IFAC-PapersOnLine*, vol. 50, no. 1, p. 12434–12439, 2017. 20th IFAC World Congress.

[93] G. Weichhart, H. Panetto, and A. Molina, "Interoperability in the cyber-physical manufacturing enterprise," *Annual Reviews in Control*, vol. 51, p. 346–356, 2021.

[94] European Parliament, Council of European Union, "Directive 2006/42/ec of the european parliament and of the council," 2006. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32006L0042&from=EN.

[95] J.-L. Boulanger, "13 - tools qualification," in *Certifiable Software Applications 2* (J.-L. Boulanger, ed.), p. 171–197, Elsevier, 2017.

[96] P. B. Medawar, *The Limits of Science*. A Cornelia & Michael Bessie book, Harper & Row, 1984.

[97] G. Elliott, *Global Business Information Technology: An Integrated Systems Approach*. Pearson Addison Wesley, 2004.

[98] L. Wittgenstein, "Lecture on ethics," *The Philosophical Review*, vol. 74, p. 3–12, Jan. 1965.

[99] J. Furborg and A. Söderberg, "Introduction to hardware architecture and evaluation according to en iso 13849-1," tech. rep., RISE Research Institutes of Sweden, 2019.

[100] A. Dennis, B. Wixom, and R. M. Roth, *Systems Analysis and Design*. John Wiley & Sons, Inc., 8th ed., Dec 2021.

[101] C. S. Wasson, *System Analysis, Design, and Development: Concepts, Principles, and Practices*. John Wiley & Sons, Inc., Jan 2005.

[102] H. ElMaraghy, L. Monostori, G. Schuh, and W. ElMaraghy, "Evolution and future of manufacturing systems," *CIRP Annals*, vol. 70, no. 2, p. 635–658, 2021.

[103] J. Morgan, M. Halton, Y. Qiao, and J. G. Breslin, "Industry 4.0 smart reconfigurable manufacturing machines," *Journal of Manufacturing Systems*, vol. 59, p. 481–506, 2021.

[104] K. Fowler, "Chapter 1 - best practices in mission-assured, mission-critical, and safety-critical systems," in *Mission-Critical and Safety-Critical Systems Handbook* (K. Fowler, ed.), p. 1–82, Boston: Newnes, 2010.

[105] Departement of Defense, ed., *Systems Engineering Fundamentals*. Virginia: Defense Acquisition University Press, 2001.

[106] W. Leontief, "The choice of technology," *Scientific American*, vol. 252, no. 6, p. 37–45, 1985.

[107] N. Finn, "Introduction to time-sensitive networking," *IEEE Communications Standards Magazine*, vol. 2, no. 2, p. 22–28, 2018.

[108] D. Bruckner, M. Stanica, R. Blair, S. Schriegel, S. Kehrer, M. Seewald, and T. Sauter, "An introduction to OPC UA TSN for industrial communication systems," *Proceedings of the IEEE*, vol. 107, p. 1121–1131, June 2019.

[109] L. Lo Bello and W. Steiner, "A perspective on ieee time-sensitive networking for industrial communication and automation systems," *Proceedings of the IEEE*, vol. 107, p. 1094–1120, June 2019.

[110] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 11, p. 17–27, March 2017.

[111] S. Gent, P. Gutiérrez Peón, T. Frühwirth, and D. Etz, "Hosting functional safety applications in factory networks through time-sensitive networking," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, p. 230–237, 2020.

[112] H. Baniabdelghany, R. Obermaisser, and A. Khalifeh, "Extended synchronization protocol based on ieee802.1as for improved precision in dynamic and asymmetric tsn hybrid networks," in *2020 9th Mediterranean Conference on Embedded Computing (MECO)*, p. 1–8, 2020.

[113] L. Deng, G. Xie, H. Liu, Y. Han, R. Li, and K. Li, "A survey of real-time ethernet modeling and design methodologies: From avb to tsn," *ACM Comput. Surv.*, vol. 55, jan 2022.

[114] C. Simon, M. Maliosz, and M. Mate, "Design aspects of low-latency services with time-sensitive networking," *IEEE Communications Standards Magazine*, vol. 2, no. 2, p. 48–54, 2018.

[115] M. Gutiérrez, A. Ademaj, W. Steiner, R. Dobrin, and S. Punnekkat, "Self-configuration of ieee 802.1 tsn networks," in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, p. 1–8, 2017.

[116] J. L. Messenger, "Time-sensitive networking: An introduction," *IEEE Communications Standards Magazine*, vol. 2, no. 2, p. 29–33, 2018.

[117] W. Steiner, P. G. Peón, M. Gutiérrez, A. Mehmed, G. Rodriguez-Navas, E. Lisova, and F. Pozo, "Next generation real-time networks based on it technologies," in *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, p. 1–8, Sept 2016.

[118] J. Farkas, L. L. Bello, and C. Gunther, "Time-sensitive networking standards," *IEEE Communications Standards Magazine*, vol. 2, no. 2, p. 20–21, 2018.

[119] M. Pahlevan and R. Obermaisser, "Redundancy management for safety-critical applications with time sensitive networking," in *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, p. 1–7, Nov 2018.

[120] IEC/IEEE 60802, "Time-sensitive networking profile for industrial automation," standard, Institute of Electrical and Electronics Engineers, New York, USA, June 2022. https://1.ieee802.org/tsn/iec-ieee-60802/.

[121] V. Gavriluţ and P. Pop, "Scheduling in time sensitive networks (tsn) for mixed-criticality industrial applications," in *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, p. 1–4, 2018.

[122] W. Mahnke, S.-H. Leitner, and M. Damm, *OPC Unified Architecture*. Springer Publishing Company, Incorporated, 1st ed., 2009.

[123] J. Pfrommer, A. Ebner, S. Ravikumar, and B. Karunakaran, "Open source opc ua pubsub over tsn for realtime industrial communication," in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, p. 1087–1090, 2018.

[124] OPC 10000-1: UAPart 1: Overview and Concepts, "OPC Unified Architecture Part 1: Overview and Concepts," specification, OPC Foundation, Scottsdale, USA, Nov. 2022.

[125] OPC 10000-6: UA Part 6: Mappings, "OPC Unified Architecture Part 6: Mappings," specification, OPC Foundation, Scottsdale, USA, Nov. 2022.

[126] OPC 10000-14: UA Part 14: PubSub, "OPC Unified Architecture Part 14: PubSub," specification, OPC Foundation, Scottsdale, USA, Nov. 2022.

[127] OPC 10000-2: UA Part 2: Security, "OPC Unified Architecture Part 2: Security," specification, OPC Foundation, Scottsdale, USA, Nov. 2022.

[128] OPC 10000-3: UA Part 3: Address Space Model, "OPC Unified Architecture Part 3: Address Space Model," specification, OPC Foundation, Scottsdale, USA, Nov. 2022.

[129] OPC 10000-5: UA Part 5: Information Model, "OPC Unified Architecture Part 5: Information Model," specification, OPC Foundation, Scottsdale, USA, Nov. 2022.

[130] OPC 10000-12: UA Part 12: Discovery and Global Services, "OPC Unified Architecture Part 12: Discovery and Global Services," specification, OPC Foundation, Scottsdale, USA, Nov. 2022.

[131] OPC 10000-15: UA Part 15: Safety, "OPC Unified Architecture Part 15: Safety," specification, OPC Foundation, Scottsdale, USA, Nov. 2022.

[132] D. Zainzinger, "Configuration deployment for reconfigurable safety systems," Master's thesis, TU Wien, Wien, 2022. https://permalink.catalogplus.tuwien.at/AC16661416.

[133] UNIDO, "The role of government, regulations, standards and new technologies," in *International Conference on Ensuring Industrial Safety*, (Vienna, Austria), United Nations Industrial Development Organization, 2019. https://www.unido.org/sites/default/files/files/2020-01/International Conference on Ensuring Industrial Safety.pdf.

[134] W. Steiner, "An evaluation of smt-based schedule synthesis for time-triggered multi-hop networks," in *2010 31st IEEE Real-Time Systems Symposium*, p. 375–384, 2010.

[135] T. Stüber, L. Osswald, S. Lindner, and M. Menth, "A survey of scheduling algorithms for the time-aware shaper in time-sensitive networking (tsn)," 2023. Available online at: https://arxiv.org/abs/2211.10954, last accessed on 27.06.2023.

[136] M. L. Raagaard and P. Pop, "Optimization algorithms for the scheduling of ieee 802.1 time-sensitive networking (tsn)," tech. rep., Technical University of Denmark, January 2017.

[137] M. L. Raagaard, P. Pop, M. Gutiérrez, and W. Steiner, "Runtime reconfiguration of time-sensitive networking (tsn) schedules for fog computing," in *2017 IEEE Fog World Congress (FWC)*, p. 1–6, 2017.

[138] H. von Foerster, *Understanding Understanding: Essays on Cybernetics and Cognition*. New York: Springer Publishing Company, Incorporated, 1st ed., 2010.

[139] E. von Glasersfeld, *Cybernetics and the Theory of Knowledge*. UNESCO Encyclopedia, 2002.

[140] D. Reitgruber, "Knowledge base for reconfigurable safety systems," Master's thesis, TU Wien, Wien, 2022. https://permalink.catalogplus.tuwien.at/AC16662174.

[141] M. Maier, D. Emery, and R. Hilliard, "Software architecture: introducing IEEE Standard 1471," *Computer*, vol. 34, no. 4, p. 107–109, 2001.

[142] T. Marangoni, "Device and link discovery in industrial ethernet networks," Tech. Rep. 183/1-210, A-Lab @ Automation Systems Group, TU Vienna, February 2022. http://www.auto.tuwien.ac.at/bib/pdf_TR/TR0210.pdf.

[143] D. Herczeg, "Safety connection manager ui: A state-of-the-art approach," Tech. Rep. 183/1-209, A-Lab @ Automation Systems Group, TU Vienna, January 2022. http://www.auto.tuwien.ac.at/bib/pdf_TR/TR0209.pdf.

[144] S. Seitner, "Tsn scheduling with re-configuration for dynamic industrial networks," Tech. Rep. 183/1-218, A-Lab @ Automation Systems Group, TU Vienna, September 2023. http://www.auto.tuwien.ac.at/bib/pdf_TR/TR0218.pdf.

[145] D. Acker, "Semi-automatic verification of safety configurations," tech. rep., A-Lab @ Automation Systems Group, TU Vienna, 2024. Thesis in progress, not yet published.

[146] D. Reitgruber, "Functional safety communication based on opc ua," Tech. Rep. 183/1-203, A-Lab @ Automation Systems Group, TU Vienna, February 2020. http://www.auto.tuwien.ac.at/bib/pdf_TR/TR0203.pdf.

[147] F. Polin, "Opc ua safety stack configuration," tech. rep., A-Lab @ Automation Systems Group, TU Vienna, 2024. Thesis in progress, not yet published.

[148] C. Lehr, "Design and limitations of a software-based TSN end station," Master's thesis, TU Wien, Wien, 2023. https://permalink.catalogplus.tuwien.at/AC16761384.

[149] C. Lehr, "TSN demonstrator," Tech. Rep. 183/1-202, A-Lab @ Automation Systems Group, TU Vienna, January 2020. http://www.auto.tuwien.ac.at/bib/pdf_TR/TR0202.pdf.

[150] N. Hartner, "Netconf server for fss devices," tech. rep., A-Lab @ Automation Systems Group, TU Vienna, 2024. Thesis in progress, not yet published.

[151] K. Popper, *The Logic of Scientific Discovery*. London: Routledge, 2002.

[152] R. Kazman, L. Bass, M. Klein, T. Lattanze, and L. Northrop, "A basis for analyzing software architecture analysis methods," *Software Quality Journal*, vol. 13, p. 329–355, Dec 2005.

[153] P. Clements, R. Kazman, and M. Klein, *Evaluating Software Architectures: Methods and Case Studies*. Addison-Wesley Professional, 2001.

[154] A. Patidar and U. Suman, "A survey on software architecture evaluation methods," in *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, p. 967–972, 2015.

[155] P. Shanmugapriya and R. Suresh, "Software architecture evaluation methods-a survey," *International Journal of Computer Applications*, vol. 49, no. 16, 2012.

[156] M. T. Ionita, D. K. Hammer, and H. Obbink, "Scenario-based software architecture evaluation methods: An overview," in *Workshop on methods and techniques for software architecture review and assessment at the international conference on software engineering*, p. 1–12, Citeseer, 2002.

[157] R. Kazman, L. Bass, G. Abowd, and M. Webb, "Saam: a method for analyzing the properties of software architectures," in *Proceedings of 16th International Conference on Software Engineering*, p. 81–90, 1994.

[158] B. Roy and T. N. Graham, "Methods for evaluating software architecture: A survey," *School of Computing TR*, vol. 545, p. 82, 2008.

[159] J. Nestroy, *Der Schützling: Posse mit Gesang in vier Akten*. Burgtheater Wien: Programmbuch, Wien: Agnes Werk Geyer und Reisser, 1989.

Curriculum Vitae

Dipl.Ing.(FH) Dieter Etz, MBA