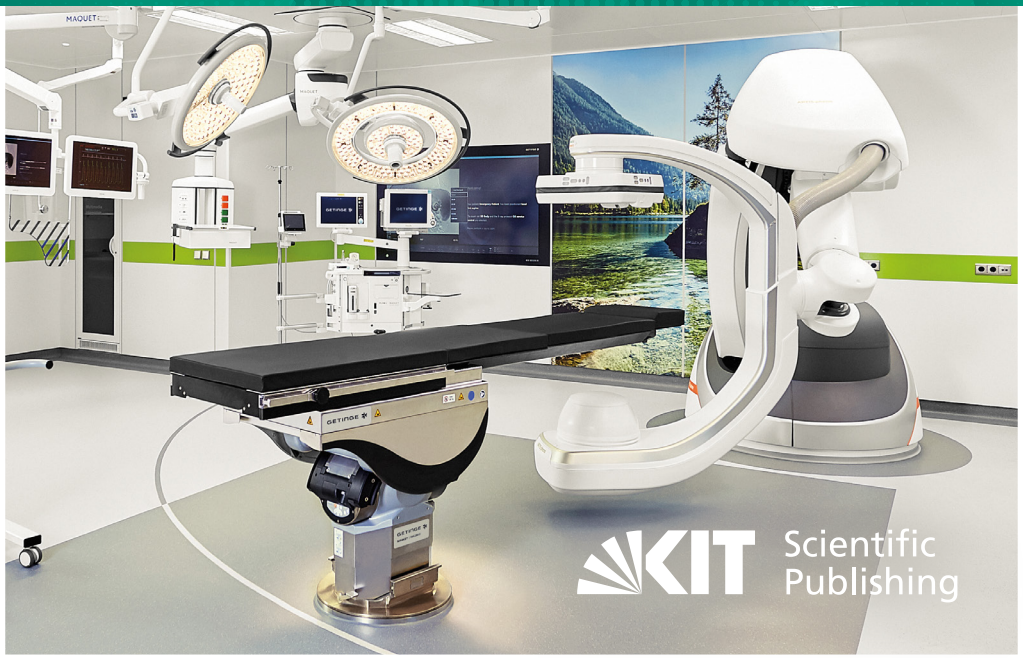
Andreas Puder

ANOMALY **DETECTION** FOR **INTEROPERABLE** AND **MODULAR** **OPERATING ROOM** TABLES



KIT Scientific Publishing

Andreas Puder

Anomaly detection for interoperable and modular operating room tables

Anomaly detection for interoperable and modular operating room tables

by
Andreas Puder

KIT Scientific Publishing

Karlsruher Institut für Technologie
Institut für Technik der Informationsverarbeitung

Anomaly detection for interoperable and modular operating room tables

Zur Erlangung des akademischen Grades eines Doktors der Ingenieurwissenschaften von der KIT-Fakultät für Elektrotechnik und Informationstechnik des Karlsruher Instituts für Technologie (KIT) genehmigte Dissertation

von Andreas Puder, M.Sc.

Tag der mündlichen Prüfung: 18. Dezember 2024
Hauptreferent: Prof. Dr.-Ing. Eric Sax
Korreferent: Prof. Dr.-Ing. Kåre Synnes

Impressum

Scientific Publishing

Karlsruher Institut für Technologie (KIT)
KIT Scientific Publishing
Straße am Forum 2
D-76131 Karlsruhe

KIT Scientific Publishing is a registered trademark
of Karlsruhe Institute of Technology.
Reprint using the book cover is not allowed.

www.bibliothek.kit.edu/ksp.php | E-Mail: info@ksp.kit.edu | Shop: www.ksp.kit.edu

Print on Demand 2025 – Gedruckt auf FSC-zertifiziertem Papier

ISBN 978-3-7315-1419-0
DOI 10.5445/KSP/1000179373

Kurzfassung

Während Konnektivität auf dem Verbraucherelektronikmarkt bereits allgegenwärtig ist, halten die entsprechenden Technologien langsam Einzug in den medizinischen Bereich. Das Risiko von Cyberangriffen, die zu Datenschutzverletzungen oder gar Patientenschäden führen könnten, wurde in der Vergangenheit als inakzeptabel angesehen. Darüber hinaus hat sich die Umsetzung in diesem streng regulierten Bereich als zunehmend schwierig erwiesen, was zum Teil daran liegt, dass die Interessen aller Beteiligten im Gesundheitswesen in Einklang gebracht werden müssen. Technologien, die Konnektivität, Interoperabilität und Automatisierung ermöglichen, haben jedoch das Potenzial, die Qualität und Effizienz der Patientenversorgung zu verbessern. In dieser Dissertation werden daher am Beispiel von Operationstischen (OP-Tischen) die notwendigen Anpassungen der Software und der elektrischen/elektronischen Architekturen medizinischer Geräte im Operationssaal (OP) untersucht, um dies zu erreichen, ohne die Sicherheit (Safety & Security) zu beeinträchtigen. Insbesondere die Erkennung von anomalem Verhalten, z. B. während der Bewegung des OP-Tisches durch Überwachung seiner Positionen, wird als wesentliches Element zur Erreichung dieses Ziels untersucht.

Medizinische Geräte müssen ihre ständige Verfügbarkeit sicherstellen, indem sie Anomalien erkennen, um Assistenzsysteme und automatisierte Funktionen im OP bereitzustellen. Das Erreichen dieser Ziele wird erschwert, wenn die Systeme nicht in sich geschlossen sind, sondern vom Benutzer durch die Kombination verschiedener Systemmodule konfiguriert werden können. Außerdem müssen Altgeräte und Zubehör aus verschiedenen Produktgenerationen über den gesamten Lebenszyklus hinweg kompatibel sein. Daher müssen die Architektur des Systems und die Gestaltung der Sicherheitsmaßnahmen flexibel sein, um sich dieser

Vielfalt anzupassen. Gleichzeitig müssen geeignete Safety- und Security-Prozesse in den Lebenszyklus des Geräts integriert werden, was in dieser Dissertation anhand von Lösungen aus der Automobilbranche behandelt wird.

Weitere Herausforderungen entstehen, wenn medizinische Systeme nicht isoliert betrachtet werden, sondern Daten von verbundenen Geräten konsumieren und diesen zur Verfügung stellen. Die langen Lebenszyklen von Systemen in Verbindung mit zunehmender Konnektivität, einschließlich Over-the-Air-Kommunikation, machen schnellere Update-Strategien erforderlich, die flexiblere Softwarearchitekturen erfordern. Sobald die Systemgrenzen durch die Abhängigkeit von anderen angeschlossenen medizinischen Geräten oder Backend-Systemen verschwimmen, steigt auch der Grad der Verteilung. Ein kritischer Faktor für die erfolgreiche Vernetzung medizinischer Systeme ist die Standardisierung der herstellerübergreifenden Kommunikation, wobei Forschungsprojekte bereits zu vielversprechenden Lösungen geführt haben. Dennoch müssen zuverlässige Daten für die beteiligten Geräte zur Verfügung stehen und Anomalien in den ausgetauschten Daten erkannt werden, um entsprechende Maßnahmen einzuleiten.

Weiterführend wird ein neuartiger Ansatz vorgestellt, bei dem hybride Plausibilitätsprüfungen eingesetzt werden, die datenbasierte und modellbasierte Algorithmen zur Anomalieerkennung kombinieren, um Herausforderungen im Bereich der Sicherheit zu bewältigen. Außerdem wird dieser Ansatz in verschiedenen Varianten für die von einem OP-Tisch an angeschlossene medizinische Geräte übermittelten Positionen angewendet und bewertet. Das Haupaugenmerk liegt dabei auf der Verringerung der Falsch-Positiv-Rate der detektierten Anomalien. Darüber hinaus wird eine gemischte elektrische/elektronische Architektur mit signalbasierter und service-orientierter Kommunikation vorgestellt, die die Flexibilitätsanforderungen für modulare und interoperable Systeme erfüllt und auch ältere Module berücksichtigt. Zusätzlich werden der neuartige Ansatz zur Anomalieerkennung und die gemischte Architektur aufeinander abgestimmt, da ihre Kombination entscheidend für die Wirksamkeit der Anomalieerkennung ist.

Abstract

While connectivity is already pervasive in the consumer electronics market, the corresponding technologies are slowly entering the medical field. The risk of cyberattacks leading to data breaches or even patient harm was considered unacceptable in the past. In addition, the implementation has proven increasingly challenging in this strictly regulated field, partly because of the necessary reconciliation of all stakeholders' interests in healthcare. However, technologies that enable connectivity, interoperability, and automation have the potential to improve the quality and efficiency of patient care. Therefore, this dissertation uses operating room (OR) tables as an example to investigate necessary adaptations to the software and electric/electronic architectures of medical devices in the OR to achieve this without compromising safety and security. In particular, detecting anomalous behavior, e.g., during the movement of the OR table by monitoring its positions, is being studied as an essential element in achieving this goal.

Medical devices must ensure their constant availability by detecting anomalies to provide assistance and automated functions in the OR. Achieving these goals is intricate if the systems are not self-contained but configurable by the user through combining different system modules. Additionally, legacy devices and accessories from different product generations must be compatible over the whole life cycle. Hence, the architecture of the system and the design of safety and security measures must be flexible to adapt to this diversity. At the same time, appropriate safety and security processes must be integrated into the device's life cycle, which this dissertation addresses with solutions from the automotive industry.

Further challenges arise when medical systems are not considered in isolation but provide data to, and consume data from connected devices. The long life cycles of systems combined with increasing connectivity, including over-the-air communication, necessitate faster update strategies requiring more flexible software architectures as well. Once the system boundaries become blurred through dependency on other connected medical devices or backend systems, the degree of distribution also increases. A critical factor for successful networking of medical systems is the standardization of cross-vendor communication, where research projects have already resulted in promising solutions. Nevertheless, reliable data must be available for the devices involved, which is why anomalies in the exchanged data must be recognized to initiate appropriate measures.

A novel approach using hybrid plausibility checks combining data-based and model-based algorithms for anomaly detection is presented to address safety and security challenges. Moreover, it is applied and evaluated in different variations for the positions communicated by an OR table to connected medical devices. The focus here is particularly on reducing the false positive rate of detected anomalies. In addition, a mixed electric/electronic architecture with signal-based and service-oriented communication that meets the flexibility requirements for modular and interoperable systems, also considering legacy modules, is presented. Furthermore, the novel anomaly detection approach and the mixed architecture are aligned, as their combination is considered crucial for the effectiveness of anomaly detection.

Foreword & Acknowledgements

My PhD journey has been an experience filled with challenges and discoveries. This dissertation represents not only my endeavors, but also the unwavering support and encouragement from many people. First and foremost, I would like to express my deepest gratitude to my family and friends who supported me. To my wife and son, your patience, understanding, and love have been my anchor throughout this journey. Your sacrifices and unwavering belief in me have been the driving force behind my perseverance. At the same time, you have always reminded me of the most important things in life.

I owe a great debt of gratitude to my doctoral advisor, Prof. Dr.-Ing. Eric Sax. Your guidance and insightful advice have been pivotal in shaping my research. I would also like to sincerely thank Prof. Dr.-Ing. Kåre Synnes for acting as my co-referent and for traveling all the way to Germany to personally attend my PhD defense. Furthermore, I would like to thank the chairman of the examination committee, Prof. Dr.-Ing. Georg Müller, and the examiners, Prof. rer. nat. Sebastian Kempf and Prof. rer. nat. Ulrich Lemmer.

Moreover, I am profoundly grateful to Getinge for their invaluable support. Special thanks go to Michael Früh and Matthieu Hirschel, whose contributions and encouragement have made this endeavor possible. I would also like to express my gratitude to my colleagues at ITIV, Getinge and the entire doctoral group of Prof. Sax. Your collaboration, insights, and camaraderie have been invaluable. Your support has not only contributed to the success of this research but has also made this experience memorable. To all those who have contributed to this journey, whether through direct assistance or moral support, I extend my sincere thanks.

Contents

1 Challenges in the Medical Device Industry

While connectivity and interoperability are commonplace in the consumer market, they are not yet widespread in the medical device industry [KSA+18] [Kas20]. This is primarily due to reasonable security and safety concerns as health data is increasingly becoming the focus of hackers. Even if no direct patient harm caused by cyberattack incidents has been reported [DSL+21], several hospital attacks have occurred. The first botnet attack, which took place in 1996, used compromised Unix machines in hospitals for a SYN flood attack [And20] [CAL96]. Thus, cybercriminals have targeted medical infrastructure already for decades. Another example of risks introduced by connectivity is the *Wannacry* infection of various British hospitals in May 2017 [And20], where the network connection had to be shut down to stop the infection of other hospitals and devices. Yet, the recent incident in which Crowdstrike's Falcon antivirus software shut down hospitals reveals that availability also plays an essential role in security measures [Cox24].

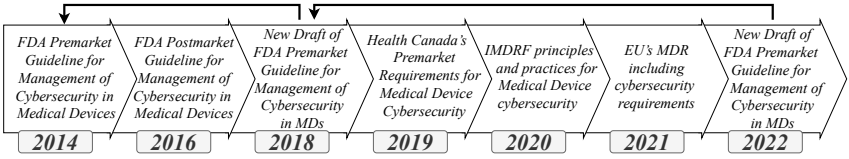


Figure 1.1: Updated evolution of medical device cybersecurity regulations based on [PHS23] [SYAY22] [Mad20] from the European Union (EU), Food and Drug Administration (FDA), Canada and International Medical Device Regulators Forum (IMDRF)

Prior risk management for medical devices addressed functional safety and did not include cybersecurity [JK17]. Thus, cybersecurity in the healthcare industry, including hospitals, is a new topic due to lacking prioritization [ASN22], although regulation started in 2014 (Fig. 1.1). Thus, manufacturers must catch up with other industries in protecting their systems and patient data and allocate significant resources toward improving their cybersecurity defenses [JK18]. A key cause for that development is the need to clarify the responsibilities for hospital security, which are shared among device manufacturers, healthcare providers, security experts, patients, and governing bodies [Del22].

According to [Wal17], cybercriminals captured 707 million records worldwide in 2015, and the healthcare sector accounted for 23% of attacks, with nearly one-fifth of all data captured. Moreover, the number of data breaches in the United States alone increased from 270 in 2015 to 519 in 2021, while there was a peak of 661 in 2020 [DRF+23]. From 2011 to 2021, data breaches affected over *303 million* people in the United States. In their study, Tin et al. point out alarming consequences of cyberattacks in the healthcare sector [DRF+23]. These include the correlation between cyberattacks and patient morbidity as well as mortality rates [Ald22].

Furthermore, the market for health records on the dark web is growing, where the average value of a health record in 2022 [Pon22] was 164$. Additionally, governments have increasingly targeted healthcare companies with cyberattacks for disruptive and political reasons [Dre17]. Lastly, IBM's *Cost of a Data Breach Report 2023* reveals that healthcare had the largest breach-related financial damages of all industries for 13 consecutive years, while the average costs of a data breach increased by 53.3% in the last three years to $10.93 million [IBM22]. Nowadays, hospitals are already equipped with a growing number of Internet of Things (IoT) devices. Still, they are unprepared to face the challenges posed by a shared network of IoT and other medical devices [AAC+20]. It applies even if a hospital is certified for the Health Insurance Portability and Accountability Act (HIPAA), a United States law that governs the security and privacy of Protected Health Information (PHI) and grants patients access to their medical records [US 08].

Still, surgical assistance systems facilitated by connecting medical devices and automation can improve surgical results [HAMPGCRM23]. The IoT has the potential to save up to 50,000 lives per year in the United States alone, which is estimated to be caused by preventable errors in hospitals [Sch14]. Improved device communication could enhance the distribution of information generally unavailable to physicians [Jam13][Sch14], allowing for improved treatment to avoid these preventable errors. Therefore, a focus on algorithms considering a set of interoperable devices - a System of Systems (SoS) - is necessary to create related solutions for medical devices. Investment programs such as the German *"Krankenhauszukunftsgesetz"* [Bun21], which targets digitization and Information Technology (IT) security, show that governments already see a need for action. A primary driver to reveal this demand was the COVID-19 pandemic, which put the hospitals - also economically - under pressure. Hence, the healthcare industry is speeding up the process of digitization [NKP+23].

Modern Cyber-Physical Systems (CPSs) (Chapter 2.7.4) go beyond traditional computing and communication functions of information technology and have the potential to fundamentally support the provision of efficient and affordable healthcare [SVDP12]. Statistics show that around 40% of surgery-related errors occur in the Operating Room (OR) [CWH+19]. According to [WCR+13], approximately 15.5 errors occurred during each surgery, of which around 23.5% were due to the device or equipment used. As ca. 43.4% of these errors were due to equipment combination and configuration errors, ca. 1.5 errors (10%) could be prevented by integrating medical devices in a SoS context [OMIM18].

In recent years, efforts have been made toward interoperable medical devices in the OR (Chapter 2.7), but many challenges remain. Most OR integration systems nowadays are proprietary and do not allow a manufacturer-independent combination of medical devices (Chapter 2.7). Still, more than the technology of connecting individual medical devices is essential to examine. New requirements resulting from interoperability and support of backend systems will impact future medical devices. Conventional software architectures and safety and security measures for self-contained systems will not be able to withstand future challenges and must be adapted to be future-proof.

1.1 Motivation

The increase in life expectancy during the 20th century [CDRV09] and the rise in chronic diseases [PWG+15] putting significant pressure on healthcare systems worldwide reinforce the necessity of aforementioned technological advances. Both effects are strongly correlated, as statistics show that 80% of people over 60 will suffer from a chronic disease [QGW+23]. This trend will continue for decades, as the "world's population aged over 60 will double from around 11% (∼810 million) in 2012 to around 22% (∼2 billion) in 2050" [Sel13]. Thus, demographics, in particular, are an essential growth factor for the healthcare industry, as the aging population increases the incidence and mortality of chronic diseases like cardiovascular diseases, tumors, and diabetes [QGW+23].

Therefore, hospitals and medical device manufacturers must become more cost-efficient while encouraging innovations and decreasing time to market. This is a challenging task due to the diversity of medical devices and the typically small development teams compared to industries such as automotive [PVR+22]. To make matters more complicated, the regulations for medical devices become stricter over time. Demands resulting from regulations, such as the Medical Device Regulation (MDR), which applies to the clinical testing and distribution of medical devices for human use in the European Union (Chapter 2.1), can be a threat to the existence of smaller companies in particular [Eng19]. Moreover, in the medical device sector, achieving therapeutic breakthroughs and cost efficiencies is mandatory to succeed in the marketplace [Sel13]. New products with similar outcomes will unlikely succeed in a market dominated by managed care and governments [Sel13].

Medical devices need to cover the quality attributes of safety, security, and reliability [GBMS18] (Appendix A.2.2). In the past, safety and security for connected medical devices were ensured by limiting the invocable functionality on their network interfaces [LSC+12]. This has the disadvantage of limited interoperability functionalities, such as positioning an Operating Room Table (OR table) remotely by a connected medical device (Chapter 2.7). By limiting these capabilities, the

potential for efficiency increases due to interoperability is reduced. In addition, the anesthesiology workplace focuses more and more on vital patient monitoring and support during surgery [CVK+18], which leads to a demand for exchanging the data of medical devices.

As digitization and interoperability are crucial to efficiency increases within healthcare, using modern IT will improve the quality of treatment in the OR in particular [CVK+18]. Furthermore, Service Oriented Architecture (SOA) has been proven for at least a decade to improve the interoperability and heterogeneity of networks in the IT domain and meet the unique characteristics of embedded networks [Käb13], leading to service-oriented communication standards in hospitals (Appendix A.1.5) and the OR in particular (Chapter 2.7).

In recent years, efforts have been made to improve the interoperability of medical devices in ORs, but it is still hardly existent [Kas20]. Economic concerns are also involved in integrating different medical devices. The OR management efforts increase with the introduction of equipment that is not integrated as a system. This leads to an intensification of the burden on the staff and the risks during surgery [OMIM18].

At first glance, the need for increased connectivity in healthcare seems indisputable. Cheng et al. emphasize this in [CWH+19] by outlining the benefits of an OR of the future-based CPS. Such systems that are part of a surgical process are also referred to as Medical Cyber-Physical Systems (MCPS) [LSC+12] (Chapter 2.7.4). For example, unintentional misuse of infusion pumps alone is estimated to kill about as many people as traffic accidents [And20], which could be prevented by automated data exchange between devices. Blaming the hospital staff for those incidents is not expedient because the sheer number of different devices makes it impossible for doctors and staff to be familiar with all the equipment. The Tokyo Women's Medical University Hospital surgery department, for example, owns 296 types and 746 units of medical equipment [OMIM18]. Nevertheless, the improvements are acquired by the partial loss of safety and security if not managed adequately. Especially hospitals have protection requirements regarding patient safety and privacy [And20].

Traditional medical devices are primarily self-contained systems with proprietary interfaces (Chapter 2.7). Introducing wireless communication technologies leads to new security challenges that are not completely solvable with standard security measures. For example, anonymizing patient data, typically done for research purposes, usually does not suffice. It is possible to tell which patient had an operation if the inquiry is specific enough [And20]. Also, it is essential to clarify how to handle cybersecurity issues in companies and hospitals once they appear. Thus, in the first place, it must be detected if an incident occurred. Therefore, anomaly detection-based Intrusion Detection Systems (IDSs) (Chapter 2.2.4) have proven to be effective in industries such as automotive and IT. Furthermore, recent activities in the automotive field show that cyberattacks are also rising due to increased networking of the formerly unconnected, closed systems, as physical access is no longer required [Web19].

Current safety and security measures must be reconsidered to overcome the above-mentioned challenges. Thus, a mandatory step is to detect anomalies in medical devices or patient states to prevent or reduce patient harm. At the same time, triggered alarms must be reliable, as false alarms may be harmful or lead to alarm fatigue (Chapter 2.1.3). Additionally, all development phases are vital for a medical device's security. Errors during a medical device's architecture and design phases usually cannot be corrected in later development [Wal17]. Hence, these phases must be carefully considered (Chapter 2.2.1).

Fortunately, the medical device industry is not alone in facing these challenges, as other industries also deal with increasingly connected devices. In the automotive industry, for example, Original Equipment Manufacturers (OEMs) have joined forces to work on a shared software architecture known as AUTomotive Open System ARchitecture (AUTOSAR). This benefits the entire industry, as the software architecture does not represent a significant competitive advantage because it cannot be directly experienced by customers [ST12]. Furthermore, it decreases the challenges and efforts for suppliers who do not have to support different operating systems or software architectures for the individual OEMs. AUTOSAR Adaptive [Vec19] (Appendix A.9.1) addresses future challenges resulting from automotive driving and increased connectivity. At the same time one of the most significant

changes here is the support for service-oriented communication. Also, anomaly detection approaches are transferable from the automotive industry, such as the *Automotive Observer* of Weber [WKSZ18] (Chapter 2.3.3). Hence, there is potential to learn from other industries, but the medical field has unique requirements and challenges that have not yet been solved in any other industry (Chapter 3.2.3).

Defining an appropriate architecture to address future challenges for any medical device is daunting due to the numerous systems classified as medical devices (Chapter 2.1.1) and their specific architecture and design. This diversity is also reflected in the medical device industry, as "the industry has a relatively small number of large, diversified companies and many smaller companies that are mainly engaged in research and development of new devices for specific therapeutic areas" [Med17]. Yet, this might be a consequence of Conway's Law [Con68], which states that "any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure." Thus, transferred to the whole healthcare sector as an organization, the wild growth of architectural approaches for medical devices might only result from the established communication structure.

1.2 Contribution

A generic system architecture for medical devices can be accomplished for a group of similar devices, as the automotive industry has proven. Hence, this dissertation contributes to the development of safe and secure software and Electric/Electronic architecture (E/E architecture) (Chapter 2.5) for future medical devices by focusing on OR tables (Chapter 2.6), which can be extrapolated to similar robotic medical devices. Additionally, OR tables are in direct contact with the patient and have demanding safety, security, and reliability requirements. Because it is a central component in surgery, it has potential to interoperate with other medical devices in the OR. Today, there already are interoperable systems like the Hybrid Operating Room (HOR) (Appendix A.1.2), which is a successful example since 2001 [Kul16] of how the combination of systems in the OR can improve surgeries.

Furthermore, the proposed software and E/E architecture are mandatory as a basis for the dissertation's core contribution, which is a novel approach for monitoring and detecting anomalous behavior in OR tables. This is done by combining a physical model, which is mathematically described based on expert knowledge, and a data-based model (Chapters 3 & 4). Thus, the suitability of SOA and its mixtures with signal-based communication, called hybrid or mixed architecture, is examined along with its value to distributed anomaly detection systems in this dissertation (Chapter 3).

Anomaly detection, which can be based on machine learning models or more classical approaches such as Kalman Filters (KFs) (Chapter 2.3), can significantly contribute to a system's safety, security, and reliability. In addition, novel approaches combining both led to promising improvements in state estimation and are therefore examined as potential improvements to anomaly detection. Due to the product-specific physical features of CPSs, the approaches of other systems cannot be applied without further ado. Since ORs are not easily trespassed by unauthorized personnel, physical features for anomaly detection are promising, especially because sensors are less effortless to manipulate (Chapter 4.1.4).

1.3 Research Questions

Motivated by the challenges of automation, connectivity, and interoperability, the following research questions (RQ) represent the focus of this dissertation:

RQ1 *Which requirements arise from the new connectivity, interoperability, and automation challenges for a new generation of OR tables?*

RQ2 *How can security and safety be ensured for OR tables while flexibility and connectivity increase?*

RQ3 *How must software- and E/E architectures of future OR tables be designed to meet these requirements and enable new safety and security measures?*

1.4 Structure of the Dissertation

The dissertation is divided into seven chapters. Chapter 1 covers the motivation, objectives, and research questions. The foundations and state-of-the-art in science and technology (Chapter 2) cover methods and technologies that focus on the development of medical devices, anomaly detection, software architecture, and E/E architecture. In addition, similarities in other domains, such as the automotive industry, are investigated to derive existing solutions for the medical field. Chapter 3 presents the novel approach for anomaly detection, combining data-based and model-based methods in an interoperable and modular OR table, which is the main contribution of this dissertation. In addition, the corresponding necessary changes in the software and E/E architecture, including the legacy modules of OR tables, are presented. The anomaly detection is then applied to the example of OR table positions including architectural changes (Chapter 4). This is followed by a prototypical implementation (Chapter 5) and an evaluation (Chapter 6) of the anomaly detection approach, considering the necessary architectural changes associated with connectivity, interoperability, and automation challenges. Chapter 7 then gives a conclusion on the scientific contribution and an outlook for future work.

2 Foundations and State of the Art in Science/Technology

2.1 Development of Medical Devices

The term medical device encompasses various products and ranges from surgical instruments such as scalpels to hearing aids to surgical robots. According to the Federal Food, Drug, and Cosmetic Act of 1938, a *medical device* is defined as "an instrument, apparatus, implement, machine, contrivance, implant, in vitro reagent or other similar article that is intended for use in the diagnosis of disease or other conditions, or in the cure, mitigation, treatment, or prevention of disease" [Sel13]. Therefore, this term has encompassed various types of devices for several decades. Today, this term is even broader as the MDR (EU 2017/745) defines it as follows [Eur17]:

Definition 1 - Medical Device: "Any instrument, apparatus, appliance, software, implant, reagent, material or other article intended by the manufacturer to be used, alone or in combination, for human beings for one or more of the following specific medical purposes: 1. diagnosis, prevention, monitoring, prediction, prognosis, treatment or alleviation of disease; 2. diagnosis, monitoring, treatment, alleviation of, or compensation for, an injury or disability; 3. investigation, replacement or modification of the anatomy or of a physiological or pathological process or state; 4. providing information by means of in vitro examination of specimens derived from the human body, including organ, blood and tissue donations [...]".

2.1.1 Differentiation of Medical Devices

The MDR has been active since the 26th of May 2021 in the European Union and has replaced the former Medical Device Directive (MDD). Furthermore, it unified the regulations for active implantable and other medical devices in a single legal act. Only active medical devices are relevant here [Eur17]:

Definition 2 - Active (medical) device: "*Active device* means any device, the operation of which depends on a source of energy other than that generated by the human body for that purpose, or by gravity, and which acts by changing the density of or converting that energy. Devices intended to transmit energy, substances or other elements between an active device and the patient, without any significant change, shall not be deemed to be active devices[...]. Software shall also be deemed to be an active device."

The IEC 60601-1 [IEC20] is a widely used standard for safe medical electrical systems and equipment and is likely to be generally accepted as a standard in the United States, Canada, the European Union, Japan, Brazil, Russia, and Australia [Tec22]. This standard does not cover all medical devices. Active implantable medical devices, for example, which fall under the definition of the MDR but not under the IEC60601-1, are handled within ISO14708-1 [ISO14]. The medical devices examined here fall under IEC60601-1 and, therefore, are represented as Programmable Electrical Medical Systems (PEMSs), which are "*Medical Electrical Equipment (ME Equipment) or Medical Electrical System (ME System)* containing one or more Programmable Electrical Subsystem (PESS)." According to IEC 60601-1, ME System, ME Equipment, and PESS are defined as follows [IEC20]:

Definition 3 - Medical Electrical System: "An ME System is a combination, as specified by the manufacturer, of items of equipment, at least one of which is ME Equipment to be interconnected by functional connection [...]."

Definition 4 - Medical Electrical Equipment: "An ME Equipment is an electrical equipment having an applied part or transferring energy to or from the patient or detecting such energy transfer to or from the patient [...]. ME Equipment includes those accessories as defined by the manufacturer that are necessary to enable the normal use of the ME Equipment [...]."

Definition 5 - Programmable Electrical Subsystem: "A PESS is a system based on one or more central processing units, including their software and interfaces."

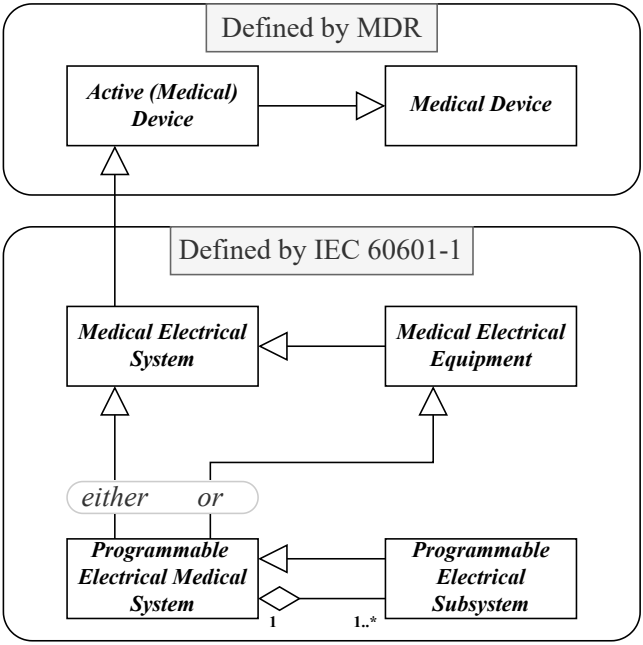


Figure 2.1: Classification, relationships, and composition of medical devices in the conceptual world of standards and laws according to [TSD+16] [Eur17]

Each set of (medical) systems is a subset or superset of other systems (Fig. 2.1). Since the PEMS are of primary concern here because OR tables (Chapter 2.6) are considered as PEMS and therefore need to fulfill the standards mentioned above, the term *medical device* always refers to PEMS in the course of this dissertation.

In addition, accessories are essential for OR tables and play a central role in this modular system. The MDR defines accessories for medical devices as follows [Eur17]:

Definition 6 - Accessory for a medical device: "*Accessory for a medical device* means an article which, whilst not being itself a medical device, is intended by its manufacturer to be used together with one or several particular medical device(s) to specifically enable the medical device(s) to be used in accordance with its/their intended purpose(s) or to specifically and directly assist the medical functionality of the medical device(s) in terms of its/their intended purpose(s)."

Another differentiation considered here is the distinction between Personal Health Devices (PHDs) and Point of Care (POC) medical devices. This is also reflected in the IEEE 11073 standards family, which is divided into "Health informatics – Point-of-care medical device communication" and "Health informatics – Personal health device communication" [Kas20]. PHDs are used directly by a single person in a private or domestic environment and are attributable to that person [Kas20]. For example, ISO/IEEE 11073-10419:2019 [ISO19c] defines the PHD communication for insulin pumps, and ISO/IEEE 11073-10424:2016 [ISO16] the PHD communication of equipment for breathing therapy of sleep apnea. Both standards define PHDs as "a device used in personal health applications."

POC medical devices, which are the subject of this dissertation, are typically used by professional staff, such as physicians and nurses, to provide near-patient care or diagnosis. Moreover, they are used to treating multiple patients after appropriate preparation [Kas20]. Examples of these systems are OR tables (Chapter 2.6) or respirators. ISO/IEEE 11073-10207 [ISO19b] defines them as follows:

Definition 7 - POC medical device: "Medical device that directly interacts with, monitors, provides treatment to, or is in some way associated with a single patient. For IEEE Std. 11073-10207, the scope of POC medical devices is further limited to patient-connected medical devices that provide support for electronic communication."

2.1.2 Medical Device Software

Software for medical devices underlies strict regulations such as the IEC62304 [IEC16b], a standard for the software life cycle process of medical devices (Appendix A.1.1). It defines a *software system* as an organized group of *software items* designed to perform at least one specific function. Moreover, a software item is "any identifiable part of a computer program," such as source code or control data. Furthermore, *software units* are decompositions of software items (Fig. 2.2), which are assignable to PEMS. By definition, software units cannot be decomposed any further, and the granularity is at the manufacturers' discretion.

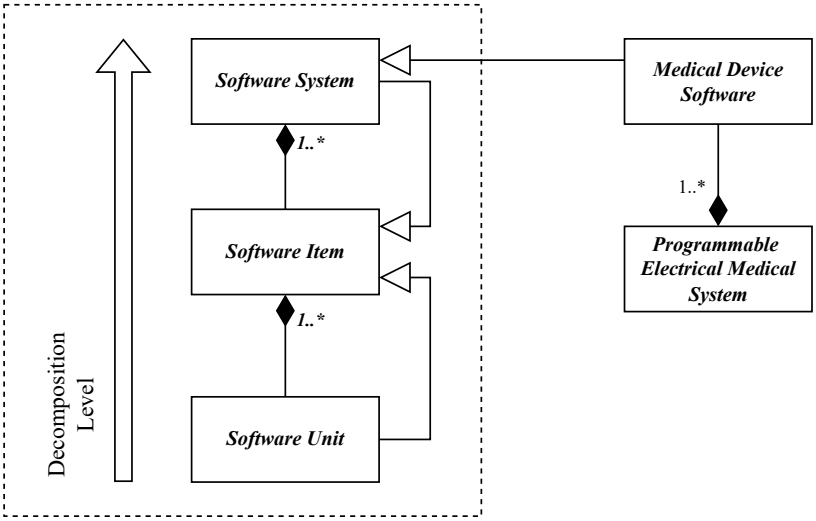


Figure 2.2: Software items and their relationships according to [IEC16b]

Decomposition into software items is vital for the Software Safety Classification (SSC) (Chapter 2.1.3), as the decomposed software items inherit the classification of the software they are a part of. The inheritance can be disrupted in exceptional cases, so the decomposed software item may have a lower class. Therefore, it must be argued how these software items are separated to be classified independently [IEC16b].

Medical device software in this dissertation's context is a "software system that has been developed for the purpose of being incorporated into the medical device being developed, or that is intended for use as a medical device" [IEC16b]. Furthermore, a pure software product is considered an individual medical device. In addition, legacy software for medical devices plays a crucial role primarily when medical devices communicate over an OR network (Chapter 2.7) since the diversity of different technology eras will increase over the device's life cycle. This is not only because of the increasing connectivity and interoperability: System-OR tables (Chapter 2.6.3) are already composed of different system components with partially legacy software. The IEC62304 also demands risk management activities when legacy software is used [IEC16b]. Therefore, among other things, legacy software must be included in the overall architecture of the medical device, and the hazardous situations that the software affects must be evaluated and addressed with risk control measures.

2.1.3 Medical Device Safety

Safety is one of the central quality aspects of medical devices (Chapter 1). Thus, regulatory agencies such as the FDA in the United States need to be convinced by the manufacturers that all safety hazards have been considered for and mitigated [AVSL11]. Safety in terms of systems and software engineering can be defined as follows [ISO17]:

Definition 8 - Safety: "The expectation that a system does not, under defined conditions, lead to a state in which human life, health, property, or the environment is endangered."

Furthermore, functional safety is a critical requirement for medical devices. It is defined according to IEC 61508 [IEC10] as "the ability of a safety-related system to carry out the actions necessary to achieve a safe state for the Equipment Under Control (EUC) or to maintain a safe state for the EUC" [SUD21].
Functional safety must be applied whenever there is an unacceptable risk connected with the function of a medical device that leads to death or severe injury

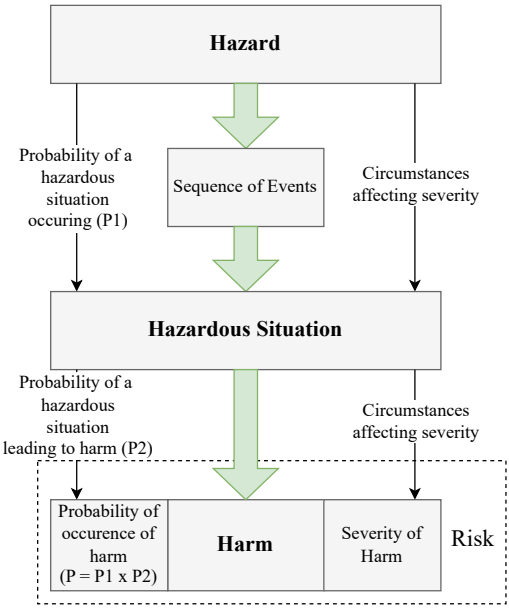


Figure 2.3: Relationship between hazard, sequence of events, hazardous situation, and harm based on [ISO19a]

[SUD21]. A risk in this context means the "combination of the probability of occurrence of harm and the severity of that harm" [ISO01][Eur17]. In comparison, *harm* is an "injury or damage to the health of people, or damage to property or the environment" [ISO01]. Several aspects determine the probability of a risk: First, exposure to a *hazardous situation*, which is a circumstance in which potential sources of harm (*hazards*) are present, and the associated events that may cause harm (*hazardous events*) must be identified. After that, the potential for harm prevention or mitigation must be evaluated, which is influenced by the safety measures of the device. This process is also referred to as *risk analysis* and leads to the detection of hazards and their corresponding risks [ISO01] (Fig. 2.3). A typical risk analysis and management methodology is the Failure Mode and Effects Analysis (FMEA) [Bij18], which focuses on identifying and evaluating potential failures and their impacts on a system.

The IEC60601-1 [IEC20] standard specifies the basic safety and essential performance requirements for medical electrical equipment. According to IEC60601-1, the first failure of a medical device must not cause such significant risks that it causes long-term damage to a patient. This can be achieved by either *minimizing the probability of the risk* by an appropriate measure or by *supplementing one measure with another* so that the risk of both measures failing is negligible [Gre14]. These activities are generally part of the process of *risk control*. In this process, the identified risks are reduced or held to a certain level through measures and decisions, resulting in the *residual risk* [ISO19a]. A typical safety measure to increase the availability of a technical device is redundancy [Bun18]. In this case, the safety-critical system is deployed at least twice, which has disadvantages such as increased costs (Chapter 3.2.4).

Risk control measures external to a software system, e.g., hardware, other independent software systems, or healthcare procedures, may degrade the SSC if its classification is B or C [IEC16b]. The SSC, which is comparable to the Automotive Safety Integrity Level (ASIL) [ISO18a] respectively, its base Safety Integrity Level (SIL) [IEC10], is made in three classes in ascending criticality according to SSC A, B, and C, and is tightly related to the risk analysis and the seriousness of the potential injury. A serious injury or illness is said to occur when it is life-threatening, results in irreversible impairment, or requires medical or surgical intervention to prevent irreversible impairment [IEC16b].

The current research on medical device safety generally leans towards process and systems engineering, while individual medical device manufacturers are responsible for implementing concrete technical measures. Although technical measures are crucial for ensuring medical device safety, they are specific to individual devices and fall within the responsibility of the manufacturer. At the same time, detecting errors/anomalies is one possible measure (Chapter 2.3.3). Therefore, research is focused on developing processes and systems that can integrate safety into the entire life cycle of medical devices, from the design phase to post-market surveillance.

Sango et al. propose a system, safety, and security co-engineering method for medical device design using model-based systems engineering methodology [SGGR19]. The functionality and connectivity of medical devices increase, leading to new safety and security challenges. They argue that traditional approaches to medical device design, which focus primarily on functionality and performance, are no longer sufficient to ensure the safety and security of these devices. The authors developed a co-engineering approach to address these challenges and integrate safety and security requirements throughout the design process, from initial concept development to final implementation of a medical device.

Miclăuș et al. [MVK+20] examine the impact of medical device design on safety, considering the concerning instances of data concealment and inadequate pre- and post-approval controls in the medical device industry. The authors observed that the design of medical devices can significantly impact their safety, and inadequate design or the absence of adequate safety features can pose significant risks to patients. Furthermore, they discuss how the current regulatory framework for medical devices may not be sufficient to ensure their safety, as demonstrated by several cases of medical device failures and recalls. They argue that more robust pre-approval testing and post-market surveillance of medical devices are needed to identify and address safety concerns before they can cause harm to patients.

2.1.4 Digital Twins and Simulation Tools

A *digital twin* is a virtual model of a physical object [MAPR19] created to simulate and analyze its behavior in a digital environment. It allows for generating data by simulating the actions and interactions of its real-world counterpart, producing datasets that mimic real-world scenarios. This concept is used in various areas, such as the automotive industry, production, or logistics, to gain system knowledge and optimize performance without directly affecting the physical system. Furthermore, digital twins are essential in leveraging simulation to generate data, enhancing understanding, efficiency, and decision-making.

Digital twins support proof of concepts and isolate phenomena in real system data and are, therefore, also beneficial in healthcare [Phi18]. They are advantageous when data collection from a real system requires unacceptable effort or is impossible, e.g., with the diversity of patients (Chapter 3.3). One of the advantages of using digital twins is the ability to generate and test new product variants within a simulated environment before physical prototypes are available. This facilitates the development and testing of system functionality in a virtual setting, thereby minimizing the risks associated with physical prototypes. In addition, using simulation models in the early stages of development helps identify and mitigate potential issues before they become costly. This enables the identification of potential design flaws and the implementation of appropriate mitigation strategies to ensure that the system is optimized for performance and safety. Various simulation tools exist for creating digital twins, whereby Gazebo [KH04], MATLAB [Mat24], and Blender [Fou24] are suitable tools [Lab22] (Table A.24) [Kin22] (Table A.25) [Käf17] [Hal23]:

MATLAB

MATLAB (MATrix LABoratory) [Mat24] is a tool for numerical computations derived initially from control theory, and it is widely used in engineering today. Simulink is a model-based extension for MATLAB that allows graphical modeling of control systems. A diagram in Simulink resembles a block diagram but includes proprietary syntactic and semantic extensions compared to the standardized IEC 60050-351 block diagram. Furthermore, Simscape is an extension of Simulink used to model and simulate physical systems. While Simulink is based on the representation of signal flows, Simscape blocks represent physical components, or relationships that exchange power or energy flows bidirectionally via their ports. This modeling approach is also known as acausal modeling [Kö13]. MATLAB and its extensions are available for Linux, MacOS and Windows.

Blender

Blender [Fou24] is a free and open-source 3D computer graphics software for creating animated films, visual effects, interactive 3D applications and virtual reality. It supports the entire 3D pipeline, including modeling, rigging, animation, simulation, rendering, compositing, motion tracking, and video editing. Blender

is available for Linux, MacOS, and Windows and offers extensibility through its Python Application Programming Interface (API). Moreover, it has a powerful physics engine, which is unique for a 3D modeling tool [Kin22]. In [PPE+23], Pottier et al. demonstrate the suitability of Blender to create a digital twin for the industrial application of a multi-camera metrology system.

Gazebo
Gazebo is an open-source 3D robotics simulator developed by the Open Source Robotics Foundation (OSRF). In Gazebo, scenarios can be created within an environment by integrating a physics engine, such as the Open Dynamic Engine (ODE), and enabling physically plausible interactions between objects through sensor simulation and actuator control. Simulated objects are represented with properties such as mass, velocity, and friction. This can be done using Unified Robot Description Format (URDF), an Extensible Markup Language (XML)-based format that describes a robot's structure, kinematics, dynamics, visual properties, and controller interfaces. Gazebo is used to train Artificial Intelligence (AI) systems and conduct regression tests due to its lifelike scenarios [OSR24]. It is available for Ubuntu (a Debian-based Linux distribution [Can24]), MacOS, and Windows, whereby the latter is only experimental. As the foundation behind Gazebo is developing Robot Operating System (ROS)/ROS2 (Appendix A.9.3), the support is directly integrated into the tool. ROS is an open-source middleware framework widely used to develop and control robotic systems and provide essential tools and libraries. For example, in a smart factory simulation context, Mattila et al. [MALA+22] implemented a proof-of-concept model using ROS and Gazebo to analyze different software architectures for device control within the smart factory.

2.2 Medical Device Security

Cybersecurity is becoming increasingly important for medical devices as they integrate more wireless, Internet- and network-connected capabilities, wearable media, and electronic health information exchange [FDA22]. With connectivity and interoperability being key value drivers of the future, security must be given the same priority as safety (Chapter 1) since security incidents can impact the

safety or effectiveness of the device [IEC21]. Although the medical device industry has special restrictions and requirements, there are also overlaps with the automotive sector [PRGS22].

According to Weber [Web19], a car passenger's safety depends on the faultless function of the vehicle, and this is only achievable by securing the electronic systems. Due to the similar trends in connectivity and assistance systems in the OR and the cybersecurity issues of cars revealed in the last decade [Rum22], the medical device industry will face similar challenges soon. Thus, safety and security strategies must be applied proactively, and industry practice has proven that security measures throughout the life cycle improve the security of products [IEC21]. Security in terms of system and software engineering can be defined as follows [ISO17]:

Definition 9 - Security: "The protection of system items from accidental or malicious access, use, modification, destruction, or disclosure."

Medical devices must protect sensitive personal data, and security measures must be implemented to prevent unauthorized access [Len20]. A patient's privacy can be compromised, especially when information is combined with the Electronic Health Record (EHR) (Chapter 1). These digital or physical entities are also known as *assets* and are valuable to individuals, organizations, or governments [ISO21a]. Moreover, assets are the primary target of *attacks*, defined as an "attempt to destroy, expose, alter, disable, steal, or gain unauthorized access to or make unauthorized use of an asset" [ISO18b].

Each accessible physical or functional interface of a system may expose assets to an attacker. Therefore, the sum of these accessible interfaces, which is called *attack surface* [IEC18], must be minimized by reducing vulnerabilities to avoid exploits. *Exploits* in this context mean a defined procedure to violate the system's security through a vulnerability [ISO15], while a *vulnerability* is an error or weakness, e.g., some deficiency, in the system that can be used to bypass the security strategy [ISO21a]. These *weaknesses* can be in the system's design, implementation, or operation and management [ISO21a].

A system's software security capabilities typically involve the "protection from, detection of, response to and recovery from incidents that can compromise the confidentiality, integrity or availability of the product's assets" [IEC21]. In the past, medical device manufacturers relied on countermeasures like *security through obscurity* through proprietary security solutions, which is considered ineffective [FSK10]. Another common security measure frequently applied is the restriction of callable functions via network interfaces [LSC+12], also known as the *principle of least privileges*. However, future medical devices will compete on their interface functionality [TEMH+20] (Chapter 2.7). Therefore, medical device providers must adopt countermeasures to maintain competitiveness, while these must be constantly improved as attackers discover new vulnerabilities.

2.2.1 Threat Analysis and Risk Assessment (TARA)

A Threat Analysis and Risk Assessment (TARA) must be executed early in the product development process (e.g., in V-Model[1], Fig. 2.4) to determine the vulnerabilities and threats of a system, followed by an FMEA (Chapter 2.1.3). A *threat* means there is a possibility of breaching the security of a system and compromising the confidentiality, integrity, or availability of an asset [ISO21a]. Systematic examination of these threats is part of a threat modeling process that identifies any circumstance or event that damages a system, such as a data breach or denial of service [ISO17]. The documented result of such a threat modeling process is also called a *threat model* [ISO17]. For a CPS, the related attack modeling must be appropriately performed to design security measures [SKR18] [RKG+19]. In addition, threat modeling for risk management processes (Chapter 2.1.3) will be required for medical devices in the future [IEC21]. The term *attack modeling* is similar to the term *threat modeling* and is used interchangeably, whereby attack models are based on the attacker's view of exploiting a vulnerability.

[1] The V-model is a process model for the life cycle of a system that assigns a test phase to each development phase [DW15]

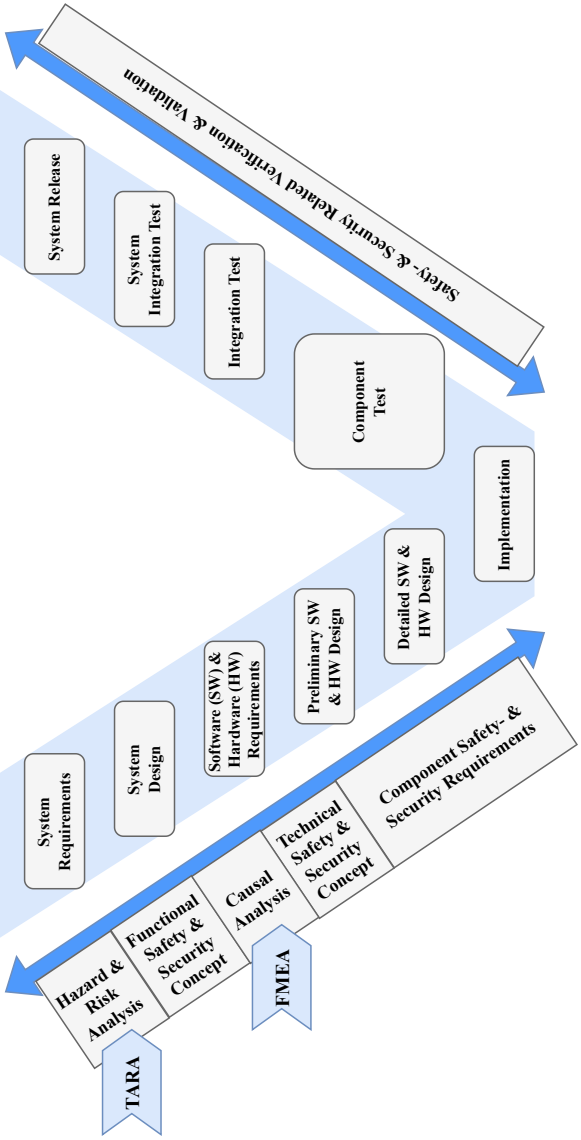


Figure 2.4: Safety and security analysis along the V-Model based on [PHS23]

Khalid et al. provide four factors on which an attack model for CPSs is based [KRS20]. The *attacker factor* focuses on the intentional altering of the CPS behavior, while the factor of *attacker's access* is defined by its type, e.g., via physical, wireless, or other network interfaces. Furthermore, the *attacking mechanism* describes how an attack is performed, and the *payload factor* is defined as targeted functionality to fulfill the *attacker's motive*. Four types of attack targets for medical devices exist [AVSL11]:

1. Directly *harm* a patient's health
2. Gain access to a *patient's health data*
3. Set up a *Denial of Service (DoS)* on a medical device so that it can no longer perform its functions
4. *Access* to an institution, e.g., hospital, to obtain patient data on a large scale

Due to dependencies, communication behavior and other side-channel parameters, as used in traditional systems for security measures, are insufficient for CPSs. This is mainly because of uncertainties in physical behavior [SKR18] [WJL+17], which is also evident in the attacks on CPS presented in [KRS20]. Thus, CPSs need another approach to analyze security, which differs from traditional IT systems [CWA17]. In addition, these measures must be adaptable to new attacks, so-called zero-day attacks, technological progress, and environmental changes [RKG+19].

Especially for systems like MCPS, the relationship between security vulnerabilities and risks regarding the safety of a product must be considered. Therefore, as part of the medical device development process, a TARA must be performed following [IEC21] to identify the vulnerabilities in the device's essential functions. A commonly used security model for TARA in the automotive industry is HEAling Vulnerabilities to ENhance Software Security and Safety (HEAVENS), which is based upon the Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege (STRIDE) model and the E-safety vehicle intrusion protected applications (EVITA) Project [LAO21]. Microsoft introduced STRIDE in 2005 to analyze software systems for possible threats [Mic16] (Appendix A.7.2).

The result of the project EVITA [Fra18] was a standardized approach for analyzing safety and security risks for automotive networks, focusing on the security of the in-vehicle system to prevent or at least detect tampering [Hen12]. HEAVENS [ISB+16] identifies security requirements focused on a particular E/E architecture and provides a risk matrix with ASIL-related threat levels considering functional use cases [HH20]. Since HEAVENS does not comply with the new cybersecurity standard ISO/SAE 21434 for road vehicles [ISO21b], Lautenbach et al. covered the corresponding gaps in HEAVENS 2.0 [LAO21]. According to the authors, HEAVENS 1.0 and 2.0 can be adapted to industries with similar characteristics, such as the medical device industry [LAO21] (Chapter 3.1).

Morana et al. outline the Process for Attack Simulation and Threat Analysis (PASTA) framework [MU15], a threat modeling method that prioritizes risk and seeks to align business objectives with technical requirements. The framework involves key decision-makers and produces an asset-focused output by enumerating and scoring threats. It consists of seven stages, including defining objectives and technical scope, breaking down the application, conducting a vulnerability analysis, modeling attacks, and assessing risk and impact. PASTA employs design and elicitation tools to support these stages, such as architectural diagrams, Data Flow Diagrams (DFDs), attack trees, use cases and abuse cases.

The Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) methodology is a tool for organizations seeking to evaluate their cybersecurity risks and identify vulnerabilities in their information infrastructure [ADSW03]. As it comprises three phases, this approach involves identifying infrastructure vulnerabilities, building asset-based threat profiles, and developing a security strategy and plans. While initially designed for large organizations, a variation called OCTAVE-S has been developed specifically for small organizations. To address concerns regarding extensive and unclear documentation [Sta16], the OCTAVE approach is under review [SCO+18]. Further threat modeling approaches are examined by Shevchenko et al. [SCO+18] and Hao et al. [HH20].

2.2.2 Defense-In-Depth

A traditional way to improve security is by implementing security layers [NWL+15] rather than relying on a single measure designed to block all incoming attacks [Yos15]. An attacker must overcome multiple protective security measures to compromise a system entirely, minimizing the overall risk of a successful attack [RGKS20]. This is also known as *Defense-in-Depth*, which is defined as an "approach to defend the system against any particular attack using several independent methods" [IEC18]. Meanwhile, the IEC81001-5-1 [IEC21] standard also requires that a medical device manufacturer apply technical requirements at each level of defense and consider safety and performance requirements when determining safety risk controls. Thus, it must be considered that any layer of protection will probably be breached and that vulnerabilities in one layer can impact the security measures in another layer [IEC18].

[NWL+15] proposes a four-layer approach for vehicles: E/E architecture, connected vehicle, in-vehicle network, and the individual Electronic Control Units (ECUs). In addition, each layer should be designed to be self-sufficient and not rely on the same measures as the others [IEC18]. Weber [Web19], which has been inspired by [EM16] and [MV14], separates layer 3 (E/E architecture Chapter 2.5) into two separate layers *access management for in-vehicle networks* and *domain-separation by E/E architecture*, resulting in a five-layered security approach (Appendix A.7.1). Although the layers are specified for vehicles in the relevant literature, they can also be applied to other CPS.

2.2.3 Zero Trust Security Model

John Kindervag introduced the term *zero trust* [Kin10] in 2010, establishing an approach in which elements within a network must be verified before they can be granted access [GC21]. Today, it describes a set of paradigms that focus defense on users, assets, and resources. After that, assets or user accounts cannot be assumed to be implicitly trustworthy just because of their physical location, network, or ownership of the assets. [RBMC20] Thus, anything that tries to establish access

must be verified, which leads to continual verification instead of verifying once at the perimeter [Dep21]. Hence, several sources see the layered defense-in-depth (Chapter 2.2.2) as obsolete; some claim that a zero trust approach should be considered. The White House, for example, published a security guideline for all ministries and authorities in the United States to move towards the zero trust paradigm [May22]. At the same time, the German Federal Office for Information Security (BSI) followed with a position paper emphasizing the need for this approach [Bun23]. Gematik, as the operator of the German healthcare system's telematics infrastructure (TI), has created a foundation of zero trust architecture in this sector with the support of the BSI [gem23]. Three core principles (CP) based on [Kin10] can be defined as a foundation [GC21]:

CP1 **Ensure all resources are accessed securely, regardless of location:** This principle breaks with traditional layered approaches and requires that all access be subject to a strict policy model.

CP2 **Adopt the least privilege strategy and strictly enforce access control:** The ability to send network packets to a system is a privilege and, therefore, needs to be managed.

CP3 **Inspect and log all traffic:** Network traffic should be comprehensively analyzed, logged, and augmented with identity and device context.

2.2.4 Intrusion Detection Systems (IDS)

In comparison to, e.g., firewalls (Appendix A.7.3), IDSs are a reactive measure because they detect potential attacks only once they occur [RGKS20]. This is done, for example, by analyzing the data traffic within networks, the system log files, or the user behavior. They are essential if the architecture of a system relies on dynamic communication. In the automotive industry, for example, with the introduction of service-oriented protocols (Chapter 2.5.3 & Appendix A.9), firewalls based on static filter tables do not suffice to secure a system [HS18]. In addition, IDSs based on machine learning can help to identify abnormal behavior, e.g., in side-channel parameters or communication patterns [KRS20]. This can be

achieved by extracting non-stationary and non-linear features from measured data [IBZ19], while these features must be cautiously selected to maintain an effective measure [GWS18]. However, machine learning models have the disadvantage of consuming more energy [KRS20] and must be used judiciously, especially for battery-powered medical devices. Furthermore, machine learning-based security bears inherent security vulnerabilities [STB+18] [HKP+18], which increase the attack surface in CPSs [ZLK+19] [KRS20].

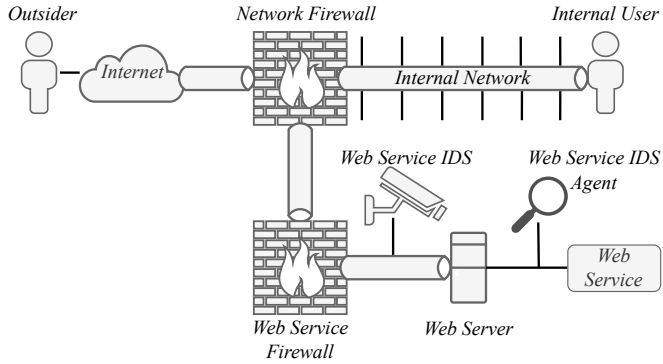


Figure 2.5: Distributed IDS placement in a network based on [NA10]

In automotive in-vehicle networks, the semiformal network and data specification leads to a more static design of E/E architecture than dynamic IT-Systems to build an anomaly-based IDS [Web19] [GWS18]. This also applies to internal networks of medical devices and networks of medical devices in the OR, as they remain stable because each medical device is introduced with a commissioning process [PDDL15]. With the introduction of service-oriented communication, data exchange will be more dynamic in the future for vehicles [JWH17][Web19] as well as for medical devices (Chapter 2.7.3).

Especially with the use of more SOA-based networks, distributed approaches, as proposed by Najjar et al. [NA10], will become more attractive in the future (Fig. 2.5) to secure an overall context of a system or SoS, such as an OR Service-oriented Device Connectivity (SDC) network, to secure it.

Compared to classical IT systems, CPSs additionally consist of sensors, processing logic, and actuators, and therefore, there are also physical features to examine for anomalies [AJMD+19]. Another classical approach is the care of an attack signature database, where medical devices can "search" for attacks that have already happened. Two essential enablers for this kind of IDS for medical devices were the founding of an Information Sharing and Analysis Center (ISAC) for the health sector [Hea22] similar to the automotive ISAC and the regular update of attack signatures [MGF10].

2.3 Anomaly Detection

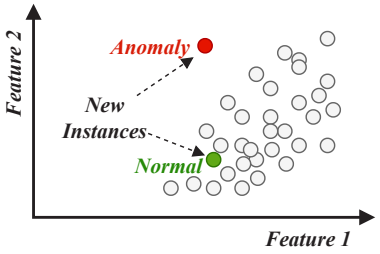


Figure 2.6: Anomaly detection based on [Gér19]

In everyday clinical practice, the amount of alarms caused by medical devices in the OR and Intensive Care Unit (ICU) is a significant burden for patients and clinicians [Kas20]. Although medical device alarms were designed as risk mitigation measures (Chapter 2.1.3) to protect patients' lives, in the aggregate, they have already become a hazard themselves. Alarm fatigue and the resulting desensitization may cause the death of patients [Cva12], and the noise caused by alarms hinders the recovery process of patients and harms caregivers [BVWB+05] [RWL08]. Since *72 to 99%* of these alarms are false alarms [SF13], it is questionable if it is a necessary side effect. Therefore, alarm triggers must be carefully designed, e.g., when detecting anomalies in the system behavior.

The IEC62304 describes *anomalies* in the context of medical devices as "any condition that deviates from the expected based on requirements specifications, design documents [...]" and "may be found during [...] test, analysis compilation,

or use [...]" [IEC16b]. In the context of this dissertation, anomalies are considered deviations from normal behavior [WKSZ18] and represent "patterns in data that do not conform to a well-defined notion of normal behavior" [CBK09]. It is close to outliers defined by [Haw80] as "an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism" (Fig. 2.6). Especially in terms of zero trust (Chapter 2.2.3), *anomaly detection* used in IDSs (Chapter 2.2.4) take a central role.

The concept that an unknown mechanism has caused suspicious behavior makes anomaly detection valuable for safety and security measures. Chandola et al. define anomaly detection as the technique of identifying patterns in data that are different from expected behavior [CBK09]. They also differ between two types of anomalies: Point anomalies describe individual data points that are irregular relative to the rest of the data. In contrast, contextual anomalies can only be classified as irregular by consideration of the context (e.g., other data points). The performance of anomaly detection algorithms can be evaluated with metrics, such as the False Positive Rate (FPR) or the False Negative Rate (FNR) (Table A.7). In the case of surgical robots, in particular, the safety of operations can be improved by accurate models of the characteristics of the robot and the human body [GBMS18]. Yet, checking the plausibility of positions, for example, presents some unique challenges in the medical field. As stated by Zhu et al. [ZCD+22], manipulation of deformable objects is an emerging research problem in robotics, with one of these robotic applications being the medical field, e.g., in surgical procedures. Since surgical robots directly interact with human tissue, deformable object manipulation is directly addressed, and the robots can be deformable in addition (Appendix A.6.2) to ensure the manipulation's safety [RNC+22].

Most current anomaly detection methods are based on protocol features and not on physical features (Chapter 2.3.3). But eavesdropping in a Controller Area Network (CAN) network (Appendix A.3.1), for example, can only be detected by considering physical transmission properties, such as changed power consumption [Mag17].

2.3.1 Kalman Filter (KF)

Anomaly detection has traditionally been based on manually created models [CLX+20] using expert knowledge, which compare the estimated and measured results to identify deviations. For example, Dynamic Bayesian Networks (DBNs) are used to detect anomalies in real-time, with new inputs being added dynamically at each time step. Therefore, adding new measurements is guided by a predefined template that outlines the conditional dependencies between features and their relationships with the existing network. Furthermore, KFs [Kal60] are an example of DBNs, and researchers such as Hill et al. have employed them to detect anomalies in low-resolution time series data [HMA09].

KFs are used, e.g., in robotic systems to improve the measurements by sensors with sensor fusion [SK16] and use a process model in state space [KRS11] (Fig. A.18). It is a method for state or signal estimation of time-independent signals in transient random processes, where the moments such as mean and variance are dependent on time [PB17]. Although it is called a filter, it is an estimator for the linear quadratic problem that is statistically optimal for an arbitrary quadratic estimation error function [GA08]. Strictly speaking, it goes beyond an estimator since it additionally determines, through the dynamics of a system, the subsequent state based on the current state [GA08]. This ability also qualifies it to detect outliers and, thus, anomalies in the state estimation or the measurement if they diverge more than a specified amount. Therefore, l_k norms are appropriate to determine the amount of deviation, such as the l_1 norm, also called Mean Absolute Error (MAE), or the l_2 norm [Gér19], also called Root Mean Square Error (RMSE), whereby larger norm indices emphasize larger values [Gér19].

The iterative process (Fig. 2.7) divides into two main steps, where, at first, the current state $\mathbf{x}^*$ is propagated. The error covariance $\mathbf{P}^*$ of the system state is updated based on an initial guess of both. In a second step, the observations $\mathbf{y}$ from the measurements are incorporated by updating the so-called Kalman gain $\mathbf{K}$, which is then used to update the estimated state $\hat{\mathbf{x}}$ with the observations $\mathbf{y}$ of the system. Afterward, the error covariance $\hat{\mathbf{P}}$ is updated considering the observation $\mathbf{y}$, and the process is repeated for each discrete step k. The corresponding variables

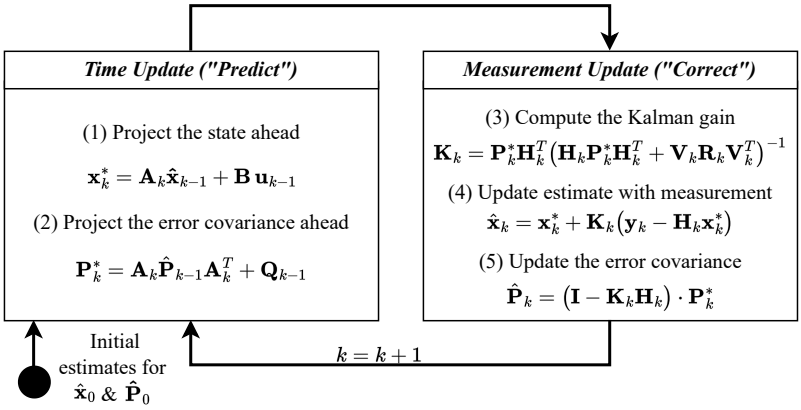


Figure 2.7: KF operation for each iteration k based on [WB06] (Table A.3)

for a system with an m-dimensional state, n-dimensional measure vector, and l-dimensional input vector are defined in Table A.3. Since $\mathbf{v}_k$ is expected to be white noise in the scenarios examined here, $\mathbf{L}_k$ is neglected [Wen11].

Which type of KF is suitable depends on the requirements and constraints, such as the modeling problem and available resources (Table 2.1 & Jin et. al [JRS+21]). While the classic KF is straightforward to implement and computationally efficient, it can only handle linear systems and does not perform well for nonlinear systems. The Extended Kalman Filter (EKF) (Appendix A.4) partially solves these issues, but not for significant nonlinearities, and also, determining the Jacobian matrices out of the nonlinear functions f and h can be challenging. Nonetheless, these issues are targeted with an Unscented Kalman Filter (UKF) (Appendix A.4). Yet, executability on a resource-constrained system is not guaranteed, especially when hard real-time is required.

According to Jin et al. [JRS+21], state estimation with sensor data using data-based models (Chapter 2.3.2) results in inadequate quality, further deteriorating as sensor noise increases. Integrating these approaches with conventional model-based estimation, resulting in so-called *hybrid filters*, demonstrates a more promising

Table 2.1: Comparison of presented KF estimators that assume a gaussian distribution [ATGSK21]

Estimator	System Model	Efficiency	Remarks
KF	Linear	Low	Only smaller systems with small number of variables
EKF	Nonlinear	Low	Linearization may lead to divergence
UKF	Nonlinear	Medium	Performance deteriorates with increasing number of state variables

direction in research. This approach is also helpful if the system becomes challenging to model mathematically based on expert knowledge. By employing this methodology, Liu et al. [LWX20] enhanced the predicted outcome of a model-based filter. They accomplished this by training the discrepancy between the estimated and reference trajectories using a Long Short-Term Memory (LSTM) network (Appendix A.5.2). Furthermore, Zhu et al. propose to combine models into a hierarchical model to improve the sensing of deformable objects. These could be, e.g., a linear model at the lower level and a deep Neural Network (NN) that learns the full model [ZCD+22].

2.3.2 Machine Learning

Machine learning (Appendix A.5), as a fundamental sub-field of AI, is a diverse discipline (Fig. A.21) whose basic principle can be reduced to a computer program adjusting its internal parameters based on existing data sets [Web19]. Arthur Samuel has defined it as “the field of study that gives computers the ability to learn without being explicitly programmed” [Sam59] or more engineering-oriented by Tom Mitchell as follows [Mit97]:

Definition 10 - Machine Learning: "A computer program is said to learn from experience *E* with respect to some task *T* and some performance measure *P*, if its performance on *T*, as measured by *P*, improves with experience *E*."

Anomaly detection is a common use case for machine learning problems, especially when it comes to unsupervised machine learning (Appendix A.5 & Fig. A.21). The advantage of anomaly detection with machine learning is that normal behavior can be trained into a model and does not need to be known or modeled beforehand [WKSZ18]. Therefore, the system's normal mechanisms are learned to detect outliers from different mechanisms.

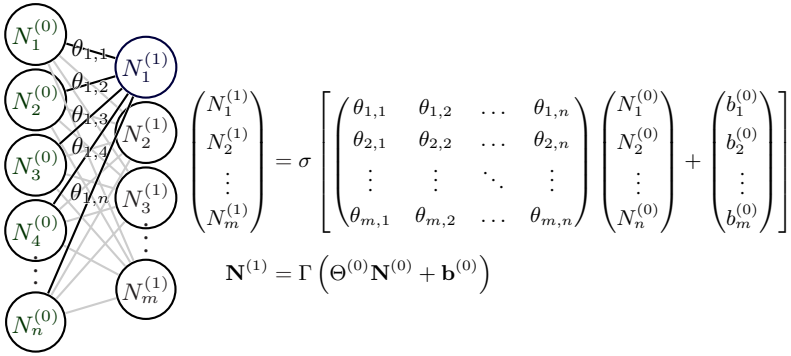


Figure 2.8: Basic NN functionality with transfer function to illustrate the transmission from the neurons of one layer to the neurons of the next layer based on [GBC16]

A NN (Fig. 2.8) is a computational model miming the structure and function of biological neural networks like the human brain. It comprises individual units, so-called neurons, arranged into layers, including an input layer, one or more hidden layers, and an output layer. These neurons are connected to adjacent layers through weighted connections, allowing information to move throughout the network. From a mathematical point of view, a NN maps an input variable tensor $\mathcal{X}$ with $\mathbf{x}_i \in \mathcal{X} \in \mathbb{R}^n$ to an output $\hat{\mathcal{Y}}$ with $\hat{y} \in \hat{\mathcal{Y}}$ that is either a space or a discrete set [GBC16]:

$$\mathcal{N}_\Theta : \mathcal{X} \rightarrow \hat{\mathcal{Y}} \tag{2.1}$$

$\mathcal{N}_\Theta$ is parameterized via a set Θ, whereby the individual parameters $\theta_{i,j} \in \Theta$ are determined by training of the NN. Furthermore, the training aims to optimize the transmission function that $\hat{y} \approx y$, which is the case if $\Theta = \Theta^*$:

$$\hat{y}_i = \Gamma(\Theta^* \mathbf{x}_i + \mathbf{b}) \tag{2.2}$$

Where Γ is an activation function used to approximate continuous transfer functions [HSW89], and $\mathbf{b}$ constants are offsets of the neurons. The Perceptron [Ros57], developed by Frank Rosenblatt in 1957, is a basic example of an artificial NN (Fig. A.22 & Fig. A.23) and the fundamental building block of a NN. Furthermore, deep learning is a machine learning approach in which an NN is realized by a multi-layer architecture [HHC+20], which makes it possible to model features that are the combinations of others [GBC16]. Moreover, this unique capability makes deep learning for anomaly detection attractive, and, in particular, Recurrent Neural Networks (RNN) are widely used in research [HHC+20]. Typical machine learning-based anomaly detection approaches are Autoencoders (AEs) [HHWB02], LSTM networks, and Isolation Forests (IFs) (Appendix A.5.2), whereby AE and LSTM are deep learning approaches. A comprehensive evaluation of these and other methods for detecting anomalies in time series can be found in [SWP22].

2.3.3 Data Science in Healthcare and Anomaly Detection Approaches

Data science for radiology is already advanced [Rea19], as the widely used Digital Imaging and Communications in Medicine (DICOM) standard (Chapter A.1.4) allows for a sufficient database [TEMH+20]. According to Teber et al. [TEMH+20], the conditions for similar progress in data science are not given in surgery. On the one hand, non-standardized data from different medical devices and, on the other hand, decisions based on interactions between the surgical team and medical devices, which are influenced by a continuous process of perception, evaluation, decision-making, and action, make it challenging to integrate algorithms. In this

context, modeling and recognition of medical workflows represent a separate field of research [Neu17] [SJT15] [Kas20]. Modeling the overall actions of the OR team plays a central role in surgical data science. Still, it is demanding due to the variety of surgical procedures, making it also challenging to collect representative data [MHVS+17]. Therefore, related research fields for medical applications, such as human activity recognition [SCSH20], must be leveraged. In [CGMB+24], for example, González et al. combine sensor data with health conditions to detect activities such as breathing or falling.

Nonetheless, intelligent alarm and support systems are possible based on the current surgery status or automatic, detailed documentation that increases legal certainty and optimizes further treatment. As a prerequisite to enable big data analytics to realize such systems, open communication standards for the exchange of data are lacking [CVK+18] as most of the medical devices use proprietary protocols (Chapter 2.7), making manual integration for each OR necessary [MHVS+17]. Furthermore, this increases the obstacles to the effective detection of anomalies. In addition, anomaly detection based on thresholds needs to be considered cautiously since the current rate of false alarms in hospitals by threshold alarms [Cli07] leads to ignoring or even turning off an alarm, reducing the quality of care [LSC+12].

Yet, these types of alarms can still save lives in emergency states, and by combining vital signs, King et al. could show that false alarms could be reduced by 57.13% without affecting true alarms [KRA+10]. Moreover, fusing patient data with real-time information provided by medical devices can improve the precision and usefulness of alarms [LSC+12] [MMR+11] [Cha01]. Context awareness created with a patient model and the data streamed by the medical devices is needed to create more intelligent alarm systems [LSC+12], e.g., based on anomaly detection. Most propositions concentrate on routing-based attacks, with only a few addressing the processing of raw data, such as physical quantities [RMB+22].

While anomaly detection can be used for security purposes in anomaly-based IDSs (Chapter 2.2.4), it can also ensure a system's safety [Web19] [Hof19], especially when physical quantities are considered. Therefore, different types of

virtual *sensors* are suitable to detect such anomalies. Müter et al. [MGF10] define eight classes of anomaly detection sensors (Table 2.2) for automotive in-vehicle network properties, such as CAN (Appendix A.3.1), that can be leveraged for medical applications as well.

Khalid et al. present a low-power and machine-learning-based runtime anomaly detection for CPS in [KRS20], while a similar approach has been presented by Weber [Web19] (Fig. A.26). This approach for general data is based on [WKSZ18]. As algorithms, Weber decides to evaluate AEs and Lightweight On-line Detector of Anomalies (LODA) [Pev16] that need the adaption for a sliding window to analyze anomalies in signals according to the ISO26262 standard (Table A.9). He proposes a hybrid approach to anomaly detection that combines static and learning checks. Therefore, time series of signals are used as input for machine learning algorithms. When a CAN message is received, the static checks extract individual signals. Subsequently, these signals are normalized and the resulting data is forwarded to a feature extraction block, which creates and manages time series based on these signals. For signal plausibility checks, Weber analyzes the signals contained in the CAN traffic of vehicles.

Based on Weber's approach, Grimm et al. propose an extension for automotive Ethernet (Appendix A.3.2) in [GWS18]. They categorize the features based on communication protocols on top of Ethernet, such as Internet Protocol (IP), User Datagram Protocol (UDP), or Transfer Control Protocol (TCP), and local and global anomalies [GWS18]. Local anomalies refer to the connection between two ECUs, while global anomalies refer to multiple ECU connections. Another anomaly detection approach by Rumez et al. is a framework based on Natural Language Processing (NLP) [RLF+20]. The language model uses n-gram algorithms adapted to diagnostic communication in the automotive industry. Therefore, they analyze the probability of byte or CAN message sequences.

The design of both Grimm [GWS18] and Weber [Web19] allows for ensemble-based learning checks, as proposed by Theissler [The17]. In an ensemble, multiple algorithms check for the same anomalies in parallel. An anomaly analysis then reconciles the different outputs of the algorithms and ultimately decides whether

Table 2.2: Sensor types for anomaly detection based on [MGF10]

Sensor Type	Description
Formality	Ensuring the formal correctness of a communication protocol involves checking aspects, such as the size of messages, headers or fields, and the integrity of the checksum, etc.
Location	Determine if a message is permissible within a particular subnetwork of a domain.
Range	Verify if the payload conforms to the prescribed data range.
Frequency	Check the accuracy of timing parameters, such as the cycle frequency or the timing interval between non-cyclic request-response messages.
Correlation	Ensuring that the correlation of messages & signals across different bus systems conforms to the specification.
Protocol	Check correctness of internal request-response protocols, including their order, start time, and other related parameters.
Plausibility	Ensure the plausibility of a message payload and prevent any infeasible correlation with previous values. A formal specification of these relations is presented in [LNJ08].
Consistency	Employing redundant data sources to validate the consistency of the data.

to classify the input as an anomaly. In comparison, Kao et al. [KSKC22] propose a sequential two-stage deep learning approach that improves the precision of a denoising AE with Gate Recurrent Units. Furthermore, Jiang et al. [JKL21] propose a three-stage approach by pre-classifying time series data with a Pearson correlation coefficient and the Dickey-Fuller test and applying different algorithms such as wavelet transform and deep learning AEs.

2.4 Rigid-Body Dynamics

Rigid-body systems are physical systems that consist of multiple mass points m_v. The distances between these points, denoted by $|\vec{r}_{v\mu}| = |\vec{r}_v - \vec{r}_\mu|$, are constant [Fli15]. Furthermore, they are connected to move together as a single entity. The dynamics of these rigid bodies is a branch of mechanics that studies their motion

in response to forces and moments. By combining appropriate parameters and models of a rigid body system that represents a robot, it is possible to simulate the motion and behavior of a robotic mechanism and control its movements [SK16], e.g., in a digital twin (Chapter 2.1.4).

2.4.1 Links

In a rigid-body system, the links are the building blocks of the robotic system and determine the overall shape and size of the robot [SK16]. The link geometry parameters define the shape and size of each link. Furthermore, the link inertia parameters describe the mass, center of mass, and moments of inertia of each link and are defined in the coordinate system of the link. These three properties describe the effort needed to move an object. The object's center of mass is the point where the object's weight is concentrated. It affects how the object moves when a force is applied to it, while the mass of an object is the amount of matter it contains, which affects how much force is required to accelerate the object. While, in general, the Center of Gravity (CoG) is used interchangeably with the center of mass, it specifically refers to the center of mass in a uniform gravitational field. Hence, the CoG and center of mass are practically the same for purposes on Earth's surface, where this condition is approximately given. Lastly, the object's rotational inertia measures its resistance to rotating around its center of mass. The rotational inertia depends on both the mass distribution of the object and the shape of the object [Fli15].

2.4.2 Joints

Robot joints are mechanical components that connect two or more links of a robot so that they can move relative to each other. Furthermore, they determine the Degrees of Freedom (DoF) and range of motion of the robot. In a robotic system, there are $2N_J$ joint-attachment frames, where one half is numbered 1 to N_J, and

the other half is numbered $J1$ to JN_J. The frames form a pair from both halves, so joint i links from frame Ji to frame i (Fig. 2.9) [SK16].

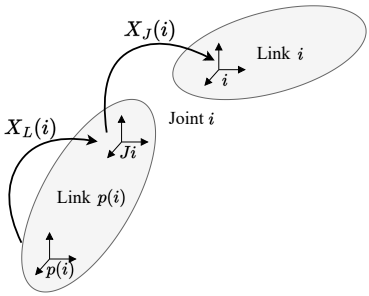


Figure 2.9: Coordinate frames $p(i)$, Ji, i and corresponding transforms $X_L(i)$, $X_J(i)$ associated with a joint based on [SK16]

The transformation of the coordinates from frame $p(i)$ to frame i can be calculated as follows [SK16]:

$$ {}^{i}X_{p(i)} = {}^{i}X_{Ji}\,{}^{Ji}X_{p(i)} = X_J(i)X_L(i) \tag{2.3} $$

The fixed link transform $X_L(i)$ transforms the base frame Ji of a joint i relative to its predecessor frame $p(i)$, while $X_J(i)$ transforms over i from Ji to i coordinates and is a variable joint transform [SK16].

Table 2.3: Joint types of robotic systems

Joint Type	Description
Revolute Joint	Hinge-like joint that allows rotation around a single axis (Fig. 2.10)
Prismatic Joint	Linear joint that allows motion along a single axis (Fig. 2.11)
Spherical Joint	Ball-and-socket joint that allow rotation in any direction (Fig. 2.12)

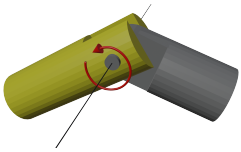


Figure 2.10: Revolute joint

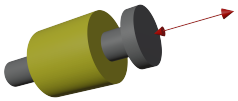


Figure 2.11: Prismatic joint

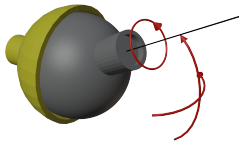


Figure 2.12: Spherical joint

Joint models describe how the joints move and are actuated and can include revolute, prismatic, and other types of joints (Table 2.3). The relationship between

connected links can be defined in their general joint model [RS88]. The velocities $\vec{v}_{rel}$ and accelerations $\vec{a}_{rel}$ over joint i and ϕ_i represent the free modes of the joint and can be calculated as follows [SK16]:

$$\vec{v}_{rel} = \phi_i \dot{\vec{q}}_i \tag{2.4}$$

$$\vec{a}_{rel} = \phi_i \ddot{\vec{q}}_i + \dot{\phi}_i \dot{\vec{q}}_i \tag{2.5}$$

ϕ_i is a $6 \times n_i$ matrix and results for a prismatic joint along a z-axis in $\phi_i = (000001)^T$ and for a revolute joint about the z-axis in $\phi_i = (001000)^T$. Here, only revolute and prismatic joints are considered, as well as 6-DoF, where ϕ_i is the identity matrix. The velocity of a link i to the velocity of its parent link $p(i)$ can be related to the $n_i \times 1$ vector $\dot{q}_i$, in which n_i is the number of DoF at the joint that connects the two links. Moreover, it describes the relative velocity of coordinate frames i to Ji. An overview of the joint model formulas of different joint types can be found in [SK16].

2.5 Electric/Electronic (E/E) Architecture

The architecture of a system is independent of its domain or purpose as described by the ISO/IEC/IEEE 42010:2011 - “Systems and software engineering - architecture description” as follows [ISO11b]:

Definition 11 - Architecture: “Fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution.”

Architecture, therefore, already describes the connection to other systems in its environment and implies the incremental improvement of the system with “*evolution*”. Furthermore, it does not specify the kind of system. The term E/E architecture is more specific and excludes domains from the system architecture, such as most of the mechanical design, but includes the connection of

ECUs via bus systems like CAN, Local Interconnect Network (LIN), or Ethernet [ZS14] (Appendix A.3). During the E/E architecture design process, the bus system topology is defined, and appropriate bus systems are selected [SWLP11]. Zimmermann and Schmidgall define E/E architecture as follows [ZS14]:

Definition 12 - E/E architecture: "The concept for the distribution of vehicle functions among different control units, their networking, the optimization of installation locations as well as the distribution and control of the necessary electrical energy in the vehicle."

In the field of medical devices, the use of the term E/E architecture is unusual. As there is no suitable equivalent, and the automotive industry has significantly pioneered embedded systems, the architectural concepts and terms are used as a reference here. Furthermore, as the automotive industry was originally based on mechanical engineering and the E/E architecture was developed evolutionarily by integrating more and more electronics, it has a similar historical background on development to OR tables.

During a development process, a bottom-up approach or a top-down can be chosen [SBHS06]. The bottom-up approach, which is more hardware-oriented, starts from an existing architecture and adds new devices and functions [ZZL+21]. Meanwhile, the top-down approach starts with the functional requirement analysis [SKKS10]. While the bottom-up approach is more focused on reusing older components, preventing the unintentional redevelopment of already existing components, the top-down approach is focused on the whole system, helping to maintain the *vision* of the developed product. Both approaches have their advantages and should be applied within a development process since they are not mutually exclusive but complementary [Gha20]. In practice, a *middle-out* approach is generally chosen [Bly19], which is ultimately a mixture of both.

Architectures can be described with Architecture Description Language (ADL) such as EAST-ADL [EAS23], which has a similar approach but is more automotive specific as it is built upon AUTOSAR [AUT19] (Appendix A.9.1). At the same time, SPES_XT (Software Platform Embedded Systems) does not rely

on AUTOSAR and is inspired by embedded systems from the automotive, automation, and avionic industry [PBDH16]. The PREEvision model [Vec24] is based on the E/E architecture (Appendix A.2, Fig. A.13) and already has practical relevance.

2.5.1 Software Architecture Patterns

Software architecture (Appendix A.2) is a sub-discipline of software engineering that originated in a conference held in 1968 in Garmisch, Germany [Dij78]. It is, therefore, a young field compared to other areas of technology with a history of several thousand years, such as civil engineering. Moreover, a uniformly accepted definition of the term is still needed, which results from software architecture being even younger, as it was in the 1990s that there was an increasing number of publications on software architecture [Gha20]. Balzert defines the term as follows [Bal11]:

Definition 13 - Software Architecture: "A software architecture describes the structures of a software system through architecture *building blocks* and their relationships and interactions with each other as well as their physical distribution. The externally visible properties of an architecture module are specified by *interfaces*."

Especially *interfaces* and *building blocks* are fundamental terms in engineering [Gha20]:

Definition 14 - Interface: "An *interface* represents a defined access point to a system or a contained building block and describes the properties of this access point."

Definition 15 - Building Block: "A *building block* offers interfaces and guarantees them in the sense of a contract as long as it is provided with its necessary interfaces. It hides implementation details behind its *interfaces* and can therefore be exchanged with other *building blocks*."

Architectural patterns are classified into top-level patterns, describing proven ways to realize and organize logical functions into software components. They help in decomposing and composing subsystems and define how they interact. Software design patterns support the implementation of functionality and define the structure of subsystems [Bal11] [Gha20]. Although software design patterns are essential for the software architecture of a system and the borders to architectural patterns are fluid, they are less relevant for the abstract levels of the architecture [GVHJ15] [Gha20] [Bal11].

Software architectural patterns can be distributed into four categories: *Adaptable Systems*, *Interactive Systems*, *Mud-to-Structure*, and *Distributed Systems* [BMR+96] [Gha20], whereby adaptable and interactive systems are not relevant here. *Mud-to-structure* patterns help separate a system's tasks into smaller partial tasks. Here, only the layer architecture is relevant, but other examples are the *Pipes and Filters* and the *Blackboard pattern* [Gha20]. The distribution of components is organized in hierarchical abstraction layers (Fig. 2.13), whereby components on the same abstraction level are combined into one layer.

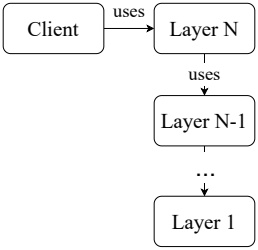


Figure 2.13: Generic layer architecture based on [Gha20]

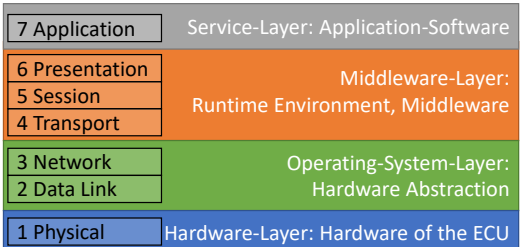


Figure 2.14: Open Systems Interconnection (OSI) layer architecture based on [IT94] [Sto21]

The OSI layer architecture (Fig. 2.14) is a well-known example of a layer-architecture (Fig. 2.13). It was developed to describe the architecture of communication systems and to support the creation of corresponding standards. Seven layers are built upon each other, with the first layer (physical layer) having the

lowest and the last (application layer) having the highest abstraction [IT94]. Logical communication, standardized with protocols, occurs within a layer. While a client only communicates with another client at the same layer, the actual data flow occurs through the layers below, including the physical medium [TW11].

Distributed systems patterns are suitable for systems divided into spatially distributed subsystems, whereby a distributed system consists of multiple processes that communicate with each other through messages [PK13]. In the *Broker* pattern, also known as Common Object Request Broker Architecture (CORBA), a broker component mediates servers and clients based on their interfaces [Sta18]. Therefore, each server registers its provided service interfaces to the broker, which then can forward a client request to a suitable server. After the server processes the request, the broker returns the response to the client (Fig. 2.15).

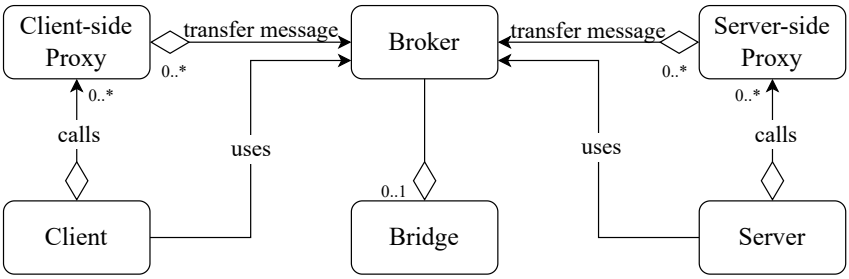


Figure 2.15: Class diagram of the Broker architecture pattern based on [Gri23] [Gha20]

Another distributed system pattern is SOA, where several definitions of SOA are based on the viewpoint. Furthermore, SOA is also a paradigm according to the SOA Manifest. Here, the definition by the World Wide Web Consortium (W3C) is used [Käb13]:

Definition 16 - Service-Oriented-Architecture: "SOA is a set of components which can be invoked, and whose interface descriptions can be published and discovered."

A *component* is a software object interacting with other components and having certain functionalities. SOA is a base/template architecture whose structure is driven by domain-partitioning [MN19] and, therefore, strongly correlated with the domain-driven design [Dow20] paradigm. Furthermore, it describes "an abstract design concept for developing software applications in a distributed environment" [Käb13]. An SOA represents technology-independent functional interfaces of software modules as services [Käb13] and describes the interactions between these [Vec19]. Three roles are defined (Fig. 2.16): A *service provider* offers services and registers them in the *service registry*, which publishes them for *service consumers*. These services are searched for by a service consumer in the service registry and then requested by the service consumer from a service provider. A *service* is a logical representation of a repeatable activity with a specified result that is self-contained [Sto21]. For this purpose, the service consists of the interfaces, the contract, the implementation, the logic, and, if necessary, associated data.

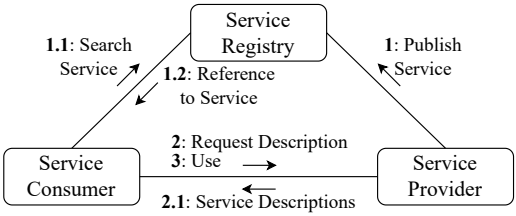


Figure 2.16: Communication diagram of an SOA based on [Kas20] [Gha20]

Composing the logic into small units with clearly defined interfaces is intended to improve reusability and enable more functionality by combining these services. Therefore, SOA primarily addresses the quality attributes reusability (maintainability), interoperability (compatibility), and flexibility[2] (Appendix A.2.2). Thus, it does not directly affect safety or security but binding services at runtime improves fault tolerance (reliability) at the application level [And13]. Furthermore, the combination of finer granularity combined with self-contained services can positively affect the safety of a system [Sto21]. This coincides with the evaluation

[2] Adaptability as part of portability and modifiability in terms of maintainability.

by Mark et al. [MN19], who emphasize reliability and maintainability (testability and fault tolerance). Göb investigates quality characteristics and models of SOA and possible tool support to ensure them [And13].

While reusability in the SOA context means that services can be combined to create new functions, quality assurance must examine all nested services and their combination [And13]. Moreover, Richards et al. [MN19] rates testability of a single service as improved. Schindewolf et al. [SSG+22] see this characteristic as a combination of all those services. Thus, SOA is more challenging to test, and Göb and Tsai see traditional procedures for software as insufficient for SOA [And13] [TZCB08].

2.5.2 E/E Architecture Topologies

The E/E architecture topology is part of the networking architecture and, thus, one abstraction layer below the software architecture layer (Fig. A.13). It defines how the ECUs are connected and the nature of the spatial distribution. The latter is more related to the wiring effort, especially for sensors and actuators, and thus indirectly influences the network of the ECUs. The topologies presented here are primarily deployed in the automotive industry but are valuable as a reference for OR tables, as they have a similar structure.

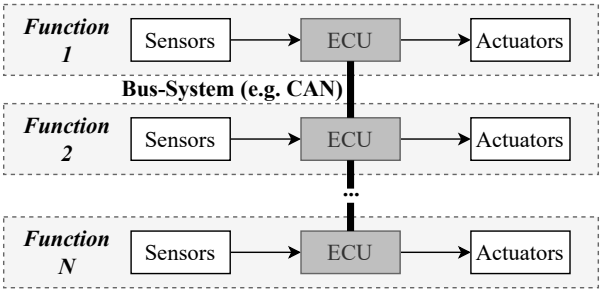


Figure 2.17: Distributed E/E architecture based on [Sto21]

Distributed Architecture

The distributed architecture is considered a function-oriented concept (Fig. 2.17) because, for each function, separate components and designated connections are provided [ZS14]. Individual ECUs are connected via a bus system, such as CAN, but they generally do not need to communicate with each other as their functionality is self-contained. This modular structure increases the reusability of the individual components but also increases the wiring efforts and the number of ECUs [BP20] [Sto21].

An automotive example is the rain-sensitive windshield wiper, which detects rain on the windshield and wipes it until it is dry. Yet, this approach tightly couples hardware and software, preventing software reusability between different hardware platforms and leading to vendor-lock-in [BP20].

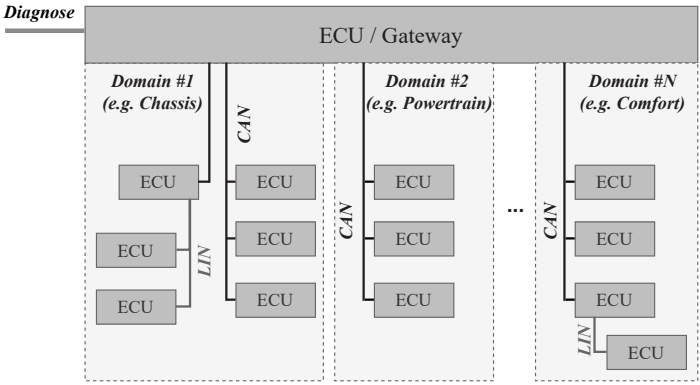


Figure 2.18: Domain-oriented E/E architecture based on [MBB18] [Web19] [Sto21]

Domain-oriented Architecture

In a modern vehicle, the E/E architecture is divided into several subnets that are dedicated to a specific domain (Fig. 2.18) like the power-train, which is a CAN-bus for motor and gear ECUs [Web19]. The evolution from the distributed architectures to this architecture was considered to improve scalability, robustness, and

maintainability, and the architecture was divided into a component/sensor/actuator layer and a domain-control layer. Only ECUs belonging to the same domain can communicate with each other, which is enforced by domain gateways that handle communication between the different domains [BP20].

Zone-oriented Architecture

Zone-oriented architecture is a hybrid between centralized and domain-oriented architecture. Here, the ECUs are grouped according to their location in the system (Fig. 2.19). All calculations are done on the central *server*, and the *zone controllers* only control actuators on the *server's* command and send the measured values of the connected sensors to the *server* [BRKW17] [Sto21]. Although this architecture is considered a transition between domain-oriented and centralized architectures, it is similar to state-of-the-art system OR tables (Chapter 2.6 & Fig. 2.19), primarily due to their modular design.

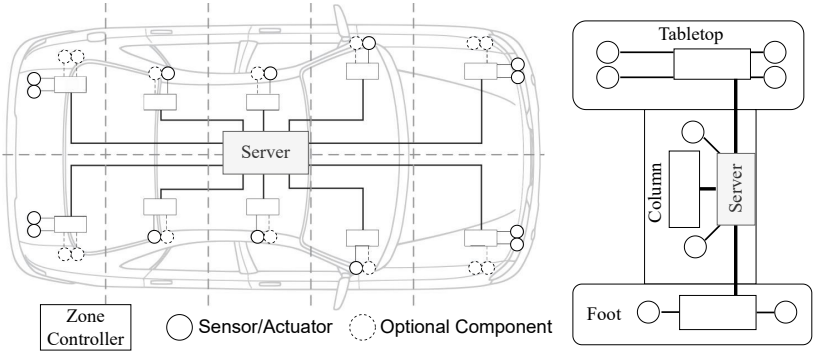


Figure 2.19: Zone-oriented E/E architecture in cars based on [BRKW17] (left) and OR tables (right)

Centralized Architecture

A centralized architecture (Fig. 2.20) aims to reduce the amount of ECUs by using more performant controllers instead [BP20]. Also, the functionality might be distributed according to needs and performance requirements. Furthermore, a service-oriented architecture (Chapter 2.5.1) is preferred here [Sto21]. In the automotive industry, the computing power of the central ECUs is necessary to realize features such as automated driving, which also need information about the

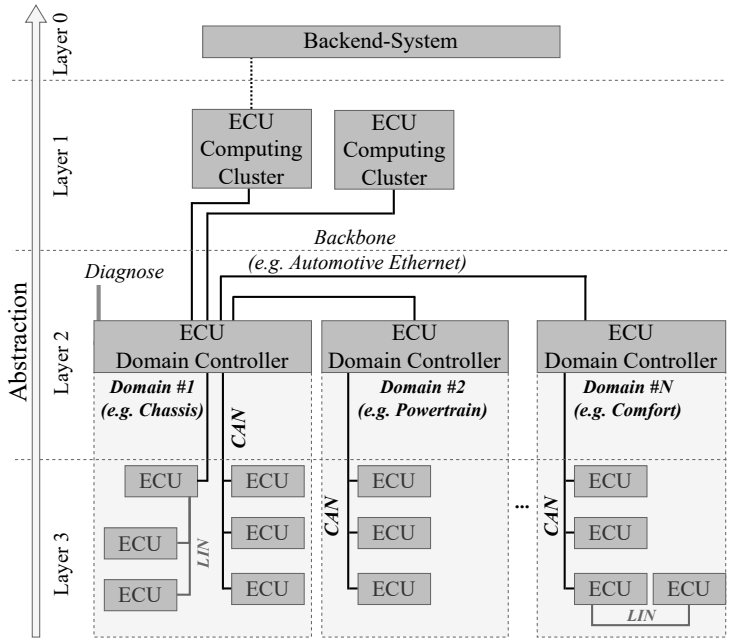


Figure 2.20: Centralized E/E architecture based on [Web19]

whole system. Therefore, sensor data from the periphery must be collected and fused into a central system state. For OR tables, future assistance systems and features like collision prevention inspired by robotic kinematic calculations also need a centralized system model (Chapter 2.6). Another essential factor for the automotive industry is the increased flexibility for software update procedures, e.g., by Over The Air (OTA) updates [GDS18].

2.5.3 Communication Paradigms

The communication and connection of software components resulting from the software architecture (Chapter 2.5.1) depend on their deployment on ECUs within an E/E architecture topology (Chapter 2.5.2) and vice versa. Two different paradigms describing their communication over the network of ECUs have been

predominant for E/E architectures: The signal-based communication paradigm that is suitable for static architectures and the service-oriented communication paradigm that is suitable for flexible architectures. Flexibility is determined by changing the deployment of software components in a network or the network topology itself being changed when the network and the networked ECUs change.

In signal-based communication, common for distributed E/E architecture topologies, signals such as temperature or speed are provided in digital form. A signal is specific information identified by its name or number and is transmitted within messages, also called frames, which contain multiple signals (Fig. 2.21). A message is identifiable by its unique identifier. Bus systems such as CAN, LIN, and FlexRay are typical representatives and transmit their data in a broadcast-based manner. Thus, each bus subscriber can read all the data and has to decide whether the received data is relevant for itself or not. Additionally, senders are transmitting messages regardless of whether receivers need them.

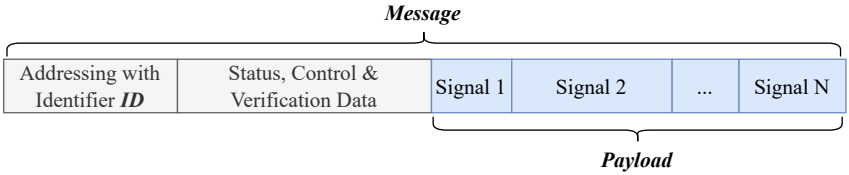


Figure 2.21: Structure of a message containing N signals based on [SZ16]

Signals manage the information exchange between software components running on different ECUs (Fig. 2.22). Furthermore, within a signal-based network architecture, the signals and their routes are statically defined between the ECUs [VS21]. All information needed to determine the physical representation of a signal based on its abstract design is stored in a *communication matrix*, a central database [ST12]. In a modern car, approximately 45.000 signals are needed for communication between up to 150 ECUs, and the compatibility surveillance of a specific ECU to a communication matrix needs to be done manually [VS21].

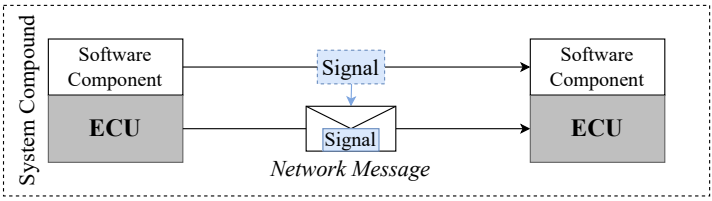


Figure 2.22: Signal-based communication over a physical communication network between software components running on different ECUs based on [Obe21]

Because signal-based communication in domain-oriented E/E architecture does not provide the adaptability and flexibility needed to address challenges such as more frequent updates during a vehicle's life cycle [VS21], the automotive industry transitions towards service-oriented communication [TMB17]. The progress in embedded device technologies in recent years has enabled technologies from IT to be feasible for the embedded domain [Käb13]. Therefore, the service-oriented communication paradigm is based on the architectural pattern SOA (Chapter 2.5.1). Furthermore, the primary intention of this paradigm is based on the so-called SOA-Manifesto, while the OASIS group has elaborated a reference model and defined it as a "paradigm for organizing and utilizing distributed capabilities[...]" [OAS06].

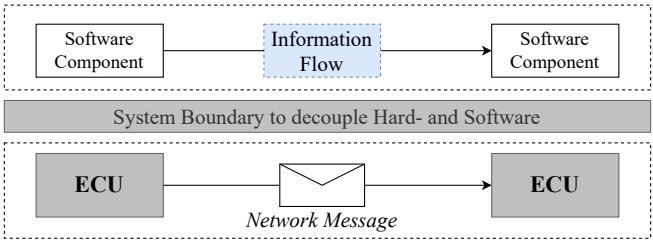


Figure 2.23: Decoupling of software and physical network communication in SOA based on [Obe21]

In embedded systems, the software architecture and the E/E architecture topology correlate more intensely than in traditional IT systems due to the interaction of the software with the physical environment. Therefore, SOA partly includes the hardware design, unlike in traditional IT (Chapter 2.5.1), such as the bus

system used (Appendix A.3). This is essential to achieve the aim of decoupling the software and the physical communication network (Fig. 2.23) to improve the flexibility of a device. In practice, this is generally facilitated through the use of a dedicated SOA middleware (Fig. 2.14, Appendix A.9.4). Since the service interfaces and their intercommunication are defined abstractly, the system is more understandable ("beauty", Appendix A.2.1) in the design phase [Vec19].

2.6 Operating Room Tables (OR Tables)

OR tables are medical devices with the "intended use of supporting and positioning a patient during surgical procedures for not more than 24 hours" [IEC16a]. They take a central role within the OR, and their positioning (Fig. 2.24) is the basis for arranging all other devices in the OR [KAKA06]. Therefore, other devices within the OR profit from being connected to the OR table and receiving, for example, its current position (Appendix A.1.2 & A.1.3).

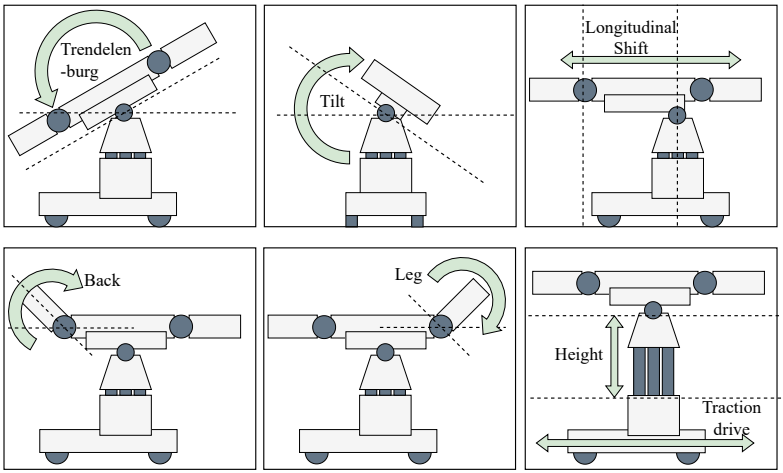


Figure 2.24: Typical joint movements of an OR table

Various surgical disciplines have developed over the last decades, creating new and unique requirements. Thus, special OR tables like the Maquet Yuno II [Get22b] (Fig. A.1), which is designed for orthopedics, traumatology, and neurology, or the Steris OT 1000 [Ste22], which is designed for orthopedics, were developed to meet these requirements. As a single specialized OR table cannot fulfill all surgical requirements, accessories can extend the application possibilities (Chapter 2.6.2). As accessories have limits, the so-called system OR tables provide a more flexible exchange of components. Hence, today's OR tables can be divided into two types of systems: mobile OR tables (Chapter 2.6.1) and system OR tables (Chapter 2.6.3). The main difference is that the translocation of mobile OR tables is supported by its fixed wheels [IEC16a] (Fig. 2.24). Another main difference is the interchangeable tabletop of system OR tables, which allows the extended adaption to the different surgery disciplines [CEP19].

2.6.1 Mobile OR Tables

Mobile OR tables consist of three main elements [KAKA06]: (column) foot, column, and tabletop (Fig. 2.25). They also have a traction drive for maneuvering the OR table (traction drive Fig. 2.24). The traction drive's castors are retractable or extendable to increase the stability of the OR table during surgery. Due to the need for stability, especially when supporting a patient, mobile OR tables have a large and heavy column foot to prevent tipping.

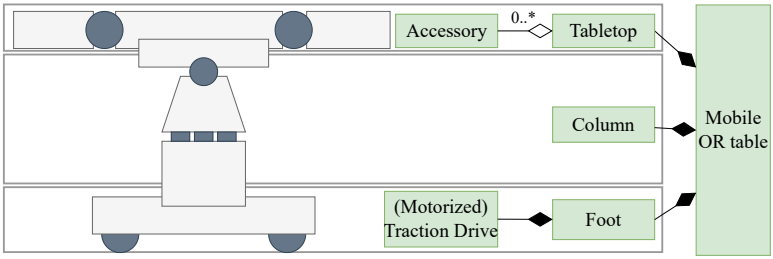


Figure 2.25: Structure and modules of a mobile OR table

The column is equipped with joints (Chapter 2.4.2) to slant (Trendelenburg[3]), tilt, and lift (Height) the patient (Fig. 2.24). The tabletop allows for individual movements of the different parts of the patient, such as moving the back or the legs. In some cases, it is possible to shift the tabletop with the patient on it in the longitudinal or lateral direction. Radiolucent tabletops to enable imaging during surgery are realized with interchangeable accessories, and there are also specialized OR tables, which provide a fully radiolucent tabletop.

2.6.2 Accessories for OR Tables

A patient may be harmed or develop pressure sores during surgery due to improper positioning techniques. Hence, the OR table must be adapted individually to the patient and the surgical intervention to prevent patient harm. Therefore, several classes (Fig. 2.26) of accessories (definition 6) ranging from various pad types to back section boards (Fig. A.2) and side rail extensions to head extensions (Fig. A.3) exist. The accessories can also include new motorized or non-motorized joints that offer a further DoF for adjusting individual body segments (Fig. A.4).

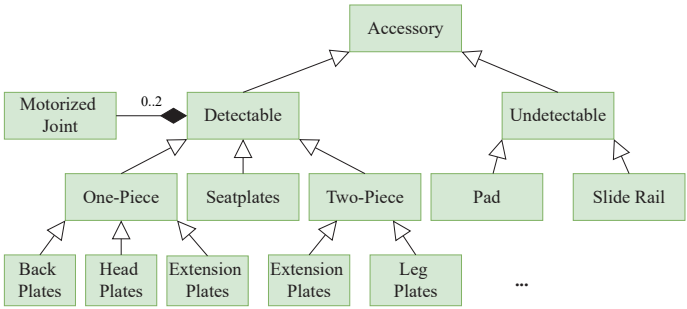


Figure 2.26: Overview of accessory classes

[3] Trendelenburg is "a supine patient position where the body is in a single plane, with that plane inclined so that the head is lower than the pelvis" [IEC16a].

In addition, accessories are classified according to their intended use and mounting point (Fig. 2.26). Not all of these can be detected by the OR table system since, for example, side rail accessories can be mounted anywhere on the side rail that has no electronic interface. Furthermore, for one product line like the Maquet Otesus from Getinge, over 100 accessories exist. At the same time, combining a head plate, two extension plates, and a leg plate alone results in 70 theoretical combination possibilities, which are even doubled if the patient's orientation is considered [PVR+22].

2.6.3 System OR Tables

New demands in the 1960s resulting from new ORs also created additional requirements that were not fulfilled by mobile OR tables (Chapter 2.6.1). To meet these new demands, with the *Maquet 1120*, the world's first system OR table was introduced in 1964 [KAKA06]. Additionally, the interchangeable tabletop resulted in the transporter being required as a new component within the OR table ensemble (Fig. 2.27).

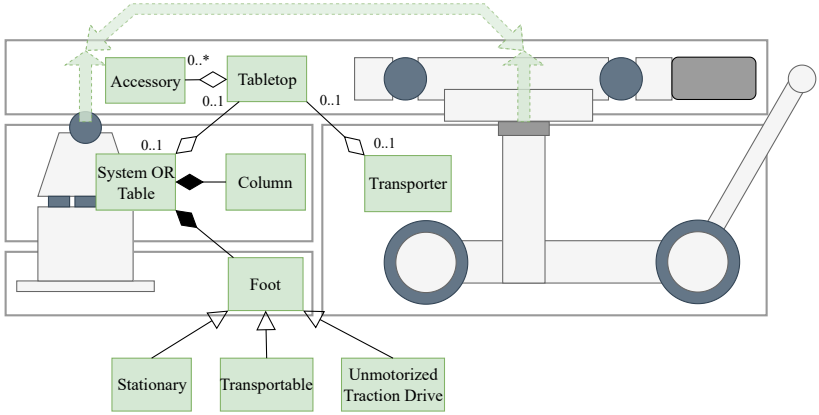


Figure 2.27: Modules of a system OR table including a transporter, which is used to exchange the tabletop and to move the column or the whole OR table

With the transporter, it is unnecessary to transport the column and the foot, which are the heaviest parts of an OR table, in and out of the OR. This improves the mobility of the OR table and, thus, its maneuverability by its increased castor size, while the castor size for mobile OR tables is limited because they need to fit into the foot. Hence, with the castors being left out, their interference can be reduced by reducing the foot size. Stationary variants have no foot and allow a turn of nearly 360° of the OR table. In addition, the foot is no longer a physical obstacle for the surgeon, and the OR table can have a lower height relative to the floor than other column types (Chapter 2.6.3).

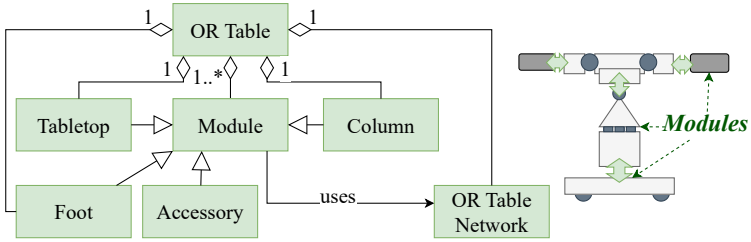


Figure 2.28: Generic OR table composition

Furthermore, the modular design of system OR tables enables more flexibility: The Maquet Otesus, for example, can be configured in 88 ways without accessories, combining 11 tabletop variants with eight columns from two product generations, each with four variants. As a rough estimate, if only four of the more than 100 accessories (Chapter 2.6.2) are additionally used, the combinations with these accessories result in 12,320 possibilities (70 accessory combinations $\times$ 2 patient orientations $\times$ 11 tabletops $\times$ 8 columns). Fig. 2.28 depicts the composition of a generic OR table independent of its type. The corresponding modules can be decomposed into joints (Fig. 2.29). An example of the OR table decomposition for the Maquet Magnus system can be found in [Käf17].

Also, with the system OR table, the circulation principle (Appendix A.1.6) was established, which saves valuable time in the OR due to the improved transportability of the patient: A patient can be prepared and anesthetized for surgery outside the OR while another surgery is still ongoing. Afterward, another transporter can

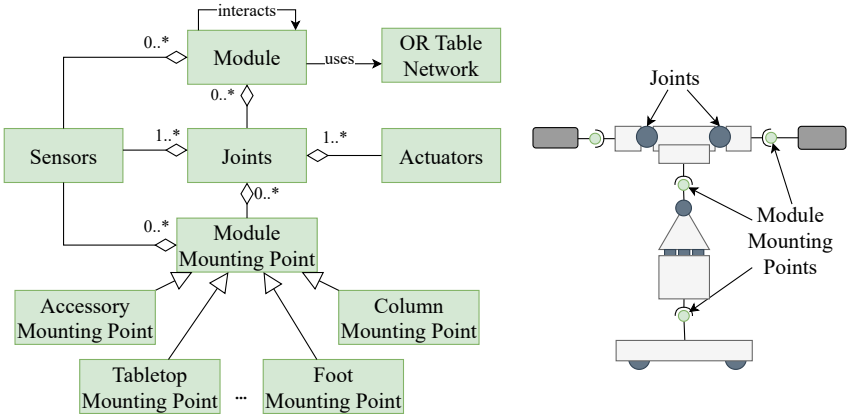


Figure 2.29: Generic OR table modules

bring the preceding patient outside the OR. Meanwhile, the waiting patient is brought to the OR. With the possibility of a stationary column in an OR, a fixed reference point for the patient determined by the fixed point and the kinematics of the OR table also enabled the HOR (Appendix A.1.2).

2.6.4 Related Research and Smart Features

Smart Technologies

The trend towards assistance systems in medical devices (Chapter 1) is also rising for OR tables. As functionality and customization options continue to expand, there is an increasing demand to support clinical staff in the OR. OR tables with built-in sensors can give surgeons and other clinical staff real-time feedback, e.g., on table movements and patient positioning during surgery. This can help to reduce the risk of fatigue, injury, and errors. Furthermore, smart technologies improve the user experience in the OR, enhance patient safety and reduce the burden on clinical staff while reducing the potential for human error. By minimizing technical issues and providing necessary support, these systems allow the clinical staff to focus on their primary medical tasks without distraction. An example is the load

recognition system [DSG+22] that can determine the patient's positioning and weight on the OR table and enable further applications like the novel anomaly detection approach in this dissertation (Chapter 3).

The TS7000 OR table from Hillrom Holding [Hil22a] and Corin from Getinge [Get23c] [DSG+22], for example, can determine the risk of tipping in case heavy patients and the CoG of the OR table is outside a stable area with load recognition systems based on load cells [WDG+23]. The HyBase v8 from Mindray provides a collision protection system that recognizes the accessories' dimensions [Min22]. This way, it can prevent the internal collisions of the different OR table components with itself and collisions with the floor. Furthermore, the HyBase v8 can detect a collision with external devices and stop its movement in such a case, similar to Corin [Get23c]. The remote controls are also becoming smarter and provide real-time position indication or touchscreen control, such as the Smart Control for the Otesus System from Getinge [Get22a] or the HyBase v8 from Mindray [Min22]. The ambient light system provides visual feedback functions, e.g., to indicate the battery status like status.Light of the Sim.Move 800 from Simeon Medical [S.I21] or Corin [Get23c].

Improvements in Ergonomics

OR tables must be designed to accommodate the ergonomic needs of surgeons and the surgical team. One way to achieve this is by making the OR table more comfortable, adjustable, and customizable for different procedures. Thus, the number and types of modules constantly expand. While "new" surgery types, such as minimally invasive surgery, enable faster patient recovery time and improved outcomes, these procedures lead to increased surgeon fatigue and musculoskeletal injuries [AWMC17]. When the patient is optimally positioned, it provides the best possible access to the surgical site. In addition, it prevents long-term complications, such as nerve damage or pressure ulcers, that can delay rehabilitation and recovery, ultimately leading to improved patient care [Get23b].

The use of robotics in the OR (Appendix A.1.3) is anticipated to enhance the ergonomics of surgery for both the surgeon and the patient. With the assistance of robotics, the surgeon no longer needs to have direct contact with the patient

during the procedure and can sit in a specialized chair with a control console (Fig. A.9). This setup provides the surgeon more comfort and ease of movement during the operation. Moreover, the patient can be positioned more flexibly since the surgeon no longer requires a specific approach angle for optimal surgical site access. As a result, both the surgeon and the patient can benefit from the improved ergonomics that robotic technology brings to the OR. The German Aerospace Center (DLR) has also developed a multi-armed surgical robot called MiroSurge, which is directly integrated into the OR table (Appendix A.1.3).

Wee et al. [WKN20] investigated the ergonomics of robotic surgery, which is becoming increasingly common in medical specialties such as gynecology, urology, and general surgery. The results suggest that robotic surgery is ergonomically superior to open and laparoscopic surgery, reducing workload and decreasing self-reported discomfort.

Relation to Reconfigurable and Modular Robots

OR tables are comparable to reconfigurable and modular robots, which consist of interconnected smaller modules that are self-contained building blocks. A more capable robot can be built by combining the modules on their equipped mounting points [MBT12] (Fig. 2.29). Most current concepts consider tiny modules with fewer DoF than a typical OR table module. As pointed out in [MBT12], there are thought experiments for modular robots changing from legged to rolling robots. This is comparable to OR tables by exchanging its heavy column and base with a transporter to improve the maneuverability of the system (Chapter 2.6.3), although it is not autonomous. The automotive industry also has similar intentions in projects like U-Shift that separates the drive unit (driveboard) and the transport capsule [MBH+21]. Also, the combination of an angiography system or a surgical robot with an OR table (Appendix A.1.2 & A.1.3) can be considered a multi-robot system. According to Moubarak et al. [MBT12], modular robots theoretically provide functional and economic advantages, which still have to be practically validated compared to traditional fixed-structure robots.

2.7 Interoperability and Connectivity in the Operating Room (OR)

Distributed systems that monitor and control the patient's physiology will continually replace self-contained devices designed and certified to treat patients without dependence on other systems [LSC+12]. The future OR will develop towards an intelligent environment [TEMH+20], where different devices like surgical robots and ventilators communicate with each other, as well as stand-alone sensors and actors. Therefore, future robotic surgical devices will have to compete, especially on their supported connectivity interfaces for information exchange and integration of information from other devices [TEMH+20]. At the same time, risk assessment and certification by regulators will become more demanding for interoperable medical devices, especially regarding safety and security [CVK+18] (Chapter 3.1).

Still, the need to improve the integration of the environment to support medical staff, e.g., in decision-making, is also reflected in the development of rapidly growing knowledge in medicine [Den11]. Nearly 90% of the knowledge in 2020 in human history has been collected in the two years before [Bey20]. This also means that medical devices used for decision-making must be updated more frequently, as the knowledge on which development was based also becomes outdated more quickly.

The interaction of different devices is called interoperability and has been defined by the MDR [Eur17] as the "ability of two or more devices, including software, [...] to exchange information and use the information, [...] communicate with each other and/or work together as intended". To enable interoperability, also compatibility needs to be ensured between the devices, whereby the MDR defines it as follows [Eur17]:

Definition 17 - Compatibility: "*compatibility* is the ability of a device, including software, when used together with one or more other devices in accordance with its intended purpose, to:"

- "perform without losing or compromising the ability to perform as intended, and/or"
- "integrate and/or operate without the need for modification or adaption of any part of the combined devices, and/or"
- "be used together without conflict/interference or adverse reaction."

Interoperability for an OR table has already been realized in several examples. Within the HOR (Appendix A.1.2), integrating an OR table with an angiography system to enable imaging-guided surgery is a well-established example of interoperability for existing challenges. Another example that is not yet as established is the integration of surgical robots (Appendix A.1.3). Medical device manufacturers have begun to take an increasing interest in this topic, as the market was expected to double from 2020 to 2023 to 12.6 billion Euro [Int20] [ITK21]. The market amounted to ∼22.1 billion for the entire medical robotics sector for 2022, with an expected annual growth rate of 16.9%, with surgical robots accounting for ∼50% [Gra22].

Also, medical device manufacturers have noticed the demand for OR integration and offer solutions for managing and controlling other medical devices. Some of the best-known are Karl Storz (OR1™ [Kar21]), Stryker (i-Suite™ [Str21]), Getinge (Tegris [Get21]), Olympus (EndoALPHA™), BrainLab (Brainsuite), Richard WOLF (CORE and core nova) [PDDL15]. Thus, manufacturers already enable partial interoperability between medical devices controlled by a central computing entity. These systems shall facilitate the data access to patient-relevant data, improve the ergonomics (Chapter 2.6.4) in the OR infrastructure control, and enhance video routing.

Although these systems have already been successfully deployed in hospitals, criticisms exist of these integrations, especially in the scientific community. One point is that these solutions are still based on the proprietary communication protocols established bilaterally by the manufacturers [MJJ15]. Due to the proprietary protocols, open and cross-manufacturer data exchange is inhibited [PDDL15], reducing the possibility of data science in the OR (Chapter 2.3.3). Furthermore,

this prevents smaller companies from entering the market, hindering innovations in the OR. In addition, by using a proprietary solution, hospitals depend on a single vendor, known as vendor-lock. Also, each device with only proprietary protocol support requires a manual integration of software solutions for each OR [MHVS+17] [TEMH+20].

According to Goldman et al. [GSJW05], interoperability between medical devices is a familiar idea, as serious efforts were already made in the late 1980s. As manufacturers recognized the opportunity in medical device interoperability around the turn of the millennium, they tried to own as much of this value chain as possible, slowing down the development process of these systems. Furthermore, the absence of an open standard is driven by the assumption that the operator becomes a medical device manufacturer if the devices' networking uses the hospital networks [Mil14].

To overcome these obstacles, innovative, open solutions to enable manufacturer-independent communication have been developed as part of research projects [KSA+18], proving their suitability for practical use in various demonstrator setups. For example, Kasparick et al. [KRS+16] present an ensemble of medical devices consisting of a surgical shaver, a surgical pump, and a High Frequency (HF) device to demonstrate a safe mechanism for remote activation via a network (Chapter 2.7.3). Arney et al. [AGWL09] synchronize a ventilator with an X-ray machine to reactivate the ventilator automatically after image generation (Chapter 2.7.2). An incident in which a patient died because the clinicians forgot to turn the device back on prompted this setup.

While standardized image data exchange through DICOM (Appendix A.1.4) already enables machine-learning applications (Chapter 2.3.3), the standardized exchange of EHR data, also known as Electronic Patient Records (EPRs), for example, has not been fully realized yet. Experts estimate that 1.000 to 2.000 clinical entities and concepts must be standardized to create a complete EHR [UP16]. Goldmann et al. [GSJW05] motivate the need for a standard by giving the designers information about what they can expect of other device's features and performance provided by an interface. Furthermore, they know what others can expect

from their device's interface, preventing heterogeneous, manufacturer-dependent communication between devices. According to Teber et al. [TEMH+20], future challenges in medicine will require long-term projects with support from the industry and will not be solved by isolated research projects. The decades-long efforts of industry and research, mostly isolated from each other, and the goals finally achieved speak for themselves.

A manufacturer-independent and open standard is crucial for security (Chapter 2.2). Because cyber-criminals will be able to infiltrate systems based on proprietary protocols and security measures [LSC+12], the cooperation of multiple manufacturers to submit to a common standard is necessary [TEMH+20] to ensure the safe and secure communication of all medical devices in the OR.

2.7.1 Smart Cyber Operating Theater (SCOT)

The aim of the Smart Cyber Operating Theater® (SCOT) project in Japan [The20], which the Tokyo Women's Medical University leads, is a connected OR by introducing CPS (Chapter 2.7.4) in ORs to connect medical devices with computing systems [OMIM18]. Interoperability is expected to improve the safety and efficiency of patient care. This is done by building the OPeLiNK® (Fig. 2.30) based on the industrial middleware Open Robot/Resource interface for the Network (ORiN) [OMIM18]. Although SCOT is still a research project, four hospitals in Japan have already operated a SCOT system since 2021 [SOMM21]. Today, OPeLiNK® is distributed and developed as a product by the company opeXpark [ope22b].

To capture and store data on a synchronized time basis generated by medical devices, such as interoperative diagnostic images, OPeLiNK® is used as a standardized data format interface [SOMM21]. Thus, the main research focus of the SCOT [OMIM18] system is to collect data from and distribute data to all connected devices, including the clinical IT, on a consistent time basis to improve the objectivity, reliability, and usefulness of medical information, which will further enable the objective evaluation of a medical process. In addition, this enables a continuous feedback loop for intraoperative analysis and treatment by taking the biological signals from the devices in the OR into account at once.

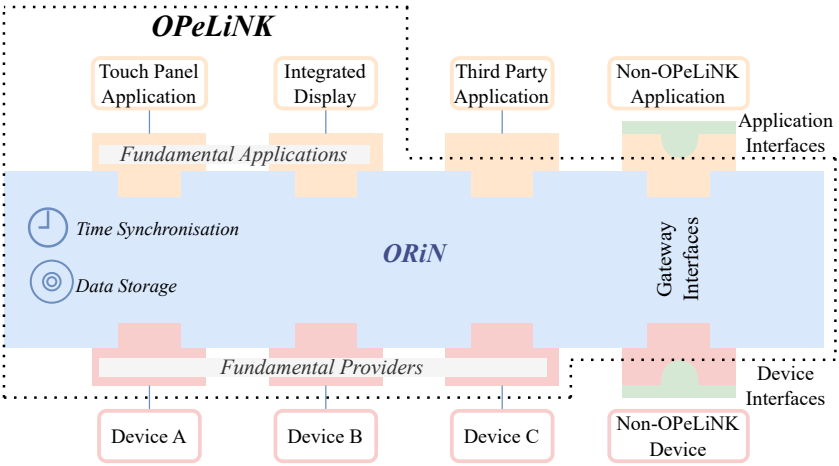


Figure 2.30: Concept of the system architecture of OPeLiNK® based on [OMIM18]

The overall system architecture of a SCOT system (Fig. 2.30) relies on *fundamental applications* and *providers* that provide and consume their data in a standardized form, such as Health Level Seven (HL7) (Appendix A.1.5) or DICOM (Appendix A.1.4). Since specific devices are abstracted behind a device interface, these can be exchanged manufacturer-independently. Furthermore, third-party applications are integrated by accessing the OPeLiNK server as clients. Gateway interfaces enable communication in both directions when integrating a non-OPeLiNK-compliant device or application into the system.

2.7.2 Medical Device Plug and Play (MDPnP)

The Medical Device Plug and Play (MDPnP) program develops integrated clinical environments and began as an offshoot of the *"operating room of the future at Massachusetts General Hospital"* project. It started as ORF PnP initiative based on a symposium held in May 2004 at CIMIT in Cambridge, MA [GSJW05]. The main research topics are forward-looking concepts and capabilities for integrated clinical environments [PDDL15].

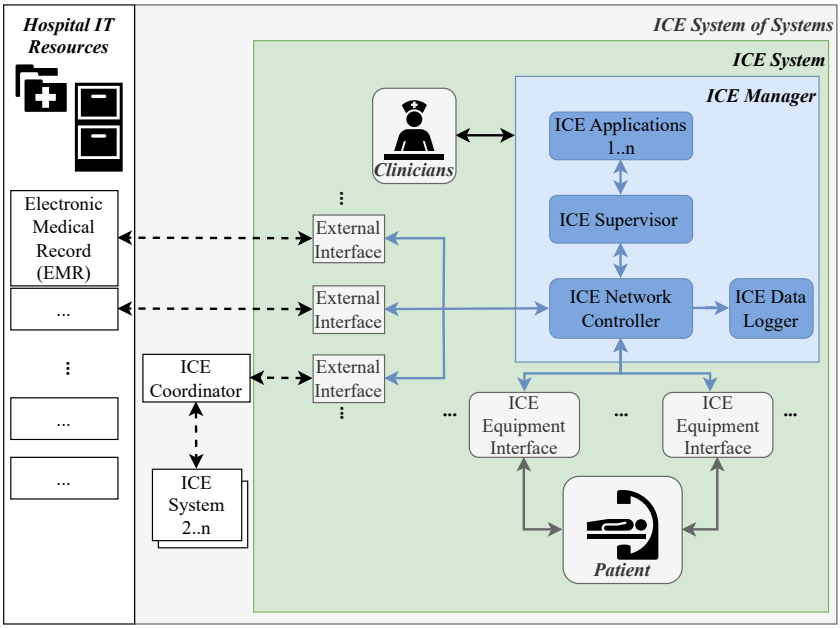


Figure 2.31: System architecture of the Integrated Clinical Environment (ICE) standard according to [APG18] based on [AST13]

The term Integrated Clinical Environment (ICE) was established by the research association MDPnP. It refers to an artificial environment of a patient in a potentially critical state (Fig. 2.31), whereby creating that artificial environment is the main point of research and the differentiating feature from the other projects. It consists of heterogeneous medical devices from different manufacturers working interoperably in a medical device system [AST13]. The ICE-standard ASTM F2761 [AST13] defines the associated requirements and conceptual models, and there is a reference implementation called OpenICE [Kas20]. Furthermore, the standard was updated through the ANSI/AAMI 2700-1 standard [ANS19].

In the ICE concept [APG18] (Fig. 2.31), the *ICE Network Controller* connects medical devices and other equipment that interact with a patient via *ICE Equipment Interfaces*. Hospital IT resources such as Electronic Medical Records

(EMRs) or other ICE systems are integrated by the ICE Network Controller. These tasks can be fulfilled by a middleware such as Data Distribution Service (DDS) (Appendix A.9.4), which is used in OpenICE. The *ICE Data Logger* gathers communication data generated by the ICE Network Controller, while the *ICE Supervisor* is used for generic services such as patient identity management. At the same time, the *ICE Applications* built the core of ICE since they are considered to fulfill the interoperability of the connected devices and external systems.

2.7.3 Service-oriented Device Connectivity (SDC)

The German OR.NET association has been working on standards for the dynamic interconnection of medical devices within the OR since 2012 [PDDL15]. As a result, the ISO/IEEE 11073 SDC, a family of communication standards [MBF+19], has been created [PDDL15]. The OR.NET project, which had preceding projects [Kas20] such as Dienst-orientierte OP-Integration (DOOP) [GBM12] and smartOR [KWL+12], was the trigger for the foundation of the OR.NET association. After the finalization of the OR.NET project, succeeding projects with more specific research areas were started that contributed to the standards' family [Kas20]. These projects concerned, for example, a test platform for dynamically networked medical devices (MoVE [OR.19]), the definition of specific medical device profiles (PoCSpec [OFF24]), and process optimization in OR through integrated medical devices (PriMed [BJB+19]). Fig. 2.32 shows the overall concept of the different OR.NET projects and the integration into the OR environment, including the IT infrastructure. Furthermore, by using the Medical Device Information Base (MDIB) as a data model to describe the state, features, and parameters of a medical device (Fig. A.31), a uniform approach to cross-manufacturer interoperability is created [PDDL15].

The *Medical Device Description* includes the Medical Device System (MDS) as the main element, serving as the logical representation of the medical device [Kas20]. It consists of at least one Virtual Medical Device (VMD), which acts as a virtual component of the device and aids in organizing it into logical objects.

Each VMD can contain multiple channels to group the metrics. These metrics represent the leaf nodes in the object-oriented description tree and contain the services the medical device offers through SDC. The metrics within these channels can be accessed for reading or writing using the optional corresponding Service and Control Objects (SCOs) (Fig. A.31). Ultimately, the data model is the distinguishing feature compared to other OR integration research projects and is one of the remaining central focus points of research, as most medical device profiles (PoCSpec [OFF24]) still have to be created (Chapter 3.4.1).

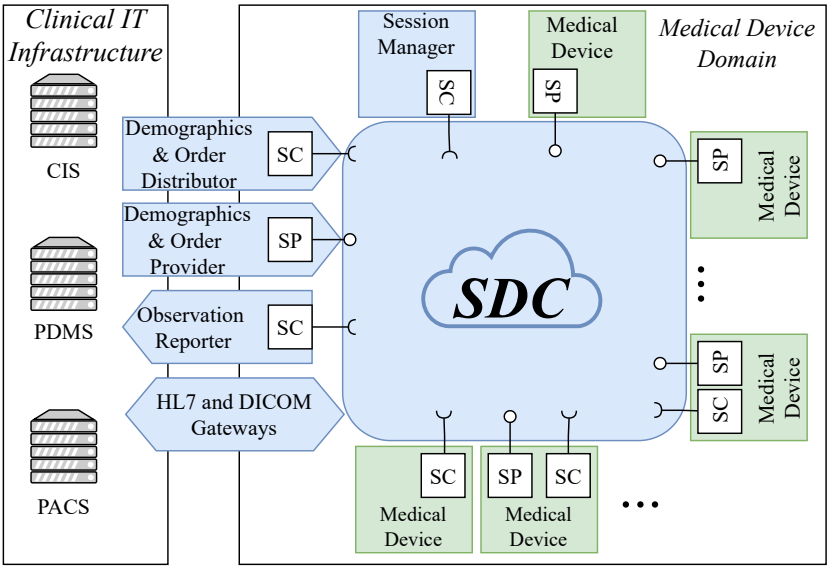


Figure 2.32: Overall concept of multiple medical devices (green) as Service Consumers (SC) and Service Providers (SP) in an SDC network including connections (blue) with clinical IT infrastructure consisting of Clinical Information System (CIS), Patient Data Management System (PDMS), and Picture Archiving and Communication System (PACS) (based on [KSA+18])

The SDC standard is a (web) services-(based) architecture (Chapter 2.5.1) with a hybrid approach for device-to-device and device-to-IT-system data exchange (Fig. 2.32). It also aims to unify the communication of medical devices similar to

standards such as HL7 (Chapter A.1.5). It uses SOAP web services and focuses on medical devices in the OR. SDC is currently based on Medical Device Profile for Web Services (MDPWS), but other middlewares like DDS (Appendix A.9.4) have been evaluated as usable [Kas20]. Non-real-time data is provided via an SOA-based implementation of web services using TCP/IP Ethernet or Wi-Fi. For this, the SDC network is used, which realizes the logical connectivity between medical devices based on the SDC standards family. Furthermore, real-time data is still being researched, as the Surgical Real-Time Bus (SRTB) with OpenPOWERLINK [PDDL15] is not being investigated further as a solution. The current research approach envisages a Time-Sensitive Networking (TSN)-based solution under the name RT-SDC [Tec23].

The concept of SDC is based on Service-Oriented Device Architecture (SODA) [DCK+06], which facilitates the integration of devices into distributed IT systems. In this context, the paradigm of service orientation (Chapter 2.5.3) is extended to devices. The device is encapsulated, and its functionality is exposed through a service. Services that realize the functions of a physical device are referred to as device services. Access works similarly to the services in an SOA, utilizing a well-defined interface that abstracts from the underlying implementation. SODA has furthermore been adapted to the Service-Oriented Medical Device Architecture (SOMDA) concept for medical applications by Kasparick et al. [KSA+18] to take the safety requirements of medical devices into account. They argue that a standardized interface description, one of the fundamental principles of SOA and SODA, is insufficient for dynamic medical device interconnection. To ensure interoperability through secure and accurate data interpretation, it is also necessary to have a standardized way of describing the provided and exchanged data. This includes describing a device's capabilities and state and modeling measurement quality or intended use of a value as in the MDIB.

2.7.4 Medical Cyber Physical Systems (MCPS)

Systems that integrate physical and cyber ("digital") components within a system are called Cyber-Physical System (CPS). They control physical objects and constantly interact between the cyber and the physical space (networked real world) [GBMS18]. Furthermore, CPS can solve social problems or industrial development by analyzing collected data on a large scale [OMIM18]. In particular, the social problems concerning healthcare improvement (Chapter 1) are an aspect that can be addressed by the application of CPSs, e.g., in hospitals. These systems are expected to reduce costs within the healthcare sector and improve patient care, e.g., by more minimal invasive technologies [KSFWK20], which is an essential aspect of robotic surgery (Chapter 2.6.4 & Appendix A.1.3). One of the Co-initiators of the term, Edward A. Lee, defined CPS as "integrations of computation with physical processes" where "embedded computers [...] control the physical processes, usually with feedback loops[...]" [Lee06]. A more recent definition by Khalid et al. [KRS20] also takes humans into account, which is a central factor for medical devices:

Definition 18 - Cyber Physical System: "A CPS is a tightly coupled communication network where several embedded computing devices, smart controllers, physical environments, and humans systematically interact with each other."

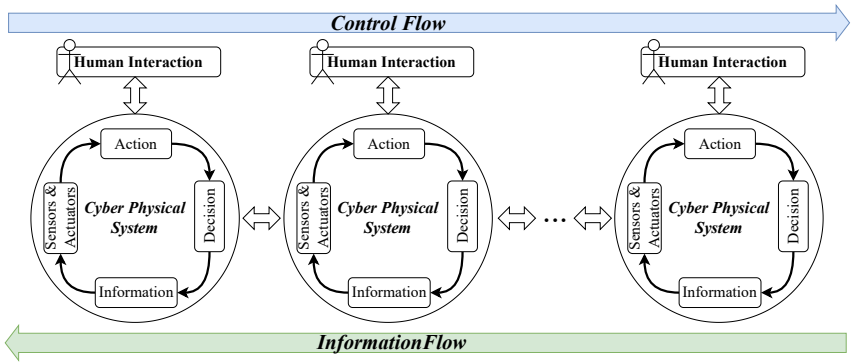


Figure 2.33: Key features of multiple CPSs interacting with each other and users based on [KRS20]

Fig. 2.33 illustrates the essential features of CPSs in a network of multiple devices interacting with humans. First, these systems collect and share sensor data via the communication network. After that, the ensemble of CPSs analyzes the data and generates control commands to control the physical environment via its actuators.

According to Langmann [Lan21], CPSs are embedded systems that used to be part of self-contained mechatronic systems and are extended with modern communication interfaces such as Ethernet (Appendix A.3.2). Closely coupled requirements use global networks such as the Internet, software services, and support for open standards. Thus, CPS can be considered an evolutionary step of embedded systems. According to Lee [LSC+12], MCPSs are a separate type because they combine the typical properties of CPSs with the physical dynamics of patient bodies. Depending on their primary purpose, they can be divided into systems that provide information about the patient's physiological state (e.g., cardiac monitors or angiography systems) and systems that can alter the patient's physiological state (e.g., OR tables or surgical robots) [LSC+12].

Since also the related term *Internet of Medical Things (IoMT)-device* is used, for example, by Gatouillat et al. [GBMS18] or Papaioannou et al. [PKM+20], there is a distinction between those terms similar to the relationship between CPS and IoT-devices. While IoMT-devices do not necessarily influence their physical environment or even control them, it is a central feature of MCPS. This is of central concern for OR tables as they influence their physical environment and directly affect the patient. Furthermore, an MCPS/CPS does not necessarily have a connection to the Internet, but the IoMT is a technology that an MCPS can use. Since these can also be connected to the Internet, the requirements and characteristics of IoMT also apply to MCPS.

There are three major challenge categories for MCPS of which all are cross-disciplinary [GBMS18][RLSS10]: First, they need to comply with reliability, robustness, and security requirements, and second, they need to be able to rely on accurate system models. Furthermore, specific verification and validation mechanisms are needed [GBMS18]. Lee et al. list six main challenges for MCPS [LSC+12] (Chapter 1.3):

- **Reliable ("High Assurance") Software:** The reliability (Appendix A.2.2) is mandatory for the MCPS's safety and effectiveness (Chapter 2.1.3).
- **Interoperability:** With an increasing number of connected devices, the according interfaces must be safe, secure, and effective.
- **Context Awareness:** Interoperability of medical devices improve understanding of the patient's health status and help in the early detection of diseases or generation of emergency alerts (Chapter 2.3).
- **Autonomy:** With the increased computational power, safe and effective therapy initiation can be achieved based on the individual patient's health status, whereby the medical device makes decisions.
- **Security and Privacy:** Patient medical data processed by these systems carries the risk of harm if tampered with or stolen (Chapters 1 & 2.2).
- **Certifiability:** Certifiability is vital to providing a cost-effective way to demonstrate the reliability of medical device software.

Despite the possible advancements enabled by CPSs, the vulnerability to cyber threats rises with networking with other systems [KRS20]. Security measures become demanding due to resource limitations, especially in battery-powered systems compared to classical IT systems. In addition, CPSs can include backend systems, most of which are under the manufacturer's control. While these services can make things more reliable overall, the potential failures can be more complicated and correlated, affecting multiple systems simultaneously [And20]. To adapt the device to unpredictable operating conditions and the growing number of connected devices over the whole product life span, it is necessary to design them as flexible and sustainable [KRS20]. Therefore, abstraction for device interfaces, which can be achieved in CPS with an SOA (Chapter 2.5.3) [GBMS18], is necessary to achieve loose coupling of the individual software items.

3 Novel Concept for Anomaly Detection in Interoperable and Modular OR Tables

3.1 TARA Inspired by the Automotive Industry

Future medical devices will be integrated with their direct environment, backend services, and hospital IT (Fig. 3.1) with increasing software functionality (Chapter 2.7). These challenges will require architectural changes (Chapter 3.4) impacting safety and security (Chapter 3.3). A networked environment allows data to be accessed and made available from other devices but also enables external misuse that directly compromises the safety of a system if adequate security measures are lacking [PDDL15]. Furthermore, a compromised MCPS raises privacy concerns [AFB+10] (Chapter 1) and can harm or kill patients (Chapter 2.2), e.g., by reprogramming the devices [HHF+08]. As security measures in the life cycle process improve the security of a system (Chapter 2.2), a TARA (Chapter 2.2.1) must be carried out (Chapter 4.1) for the OR table architecture (Fig. 3.1) to derive additional requirements (step 1 - system requirements, Fig. 2.4) and thus reduce the attack surface. A rough architecture is necessary to assess the effects of requirements realistically [Gha20].

Most current processes proposed by medical device standards and guidelines originated in the IT domain [PHS23] and thus are not adapted to the medical device needs that are characterized by a life-threatening physical environment. Therefore, the medical field lacks a concrete set of security guidelines and standards compared to ISO/SAE21434, which mandates a specific TARA workflow to be followed

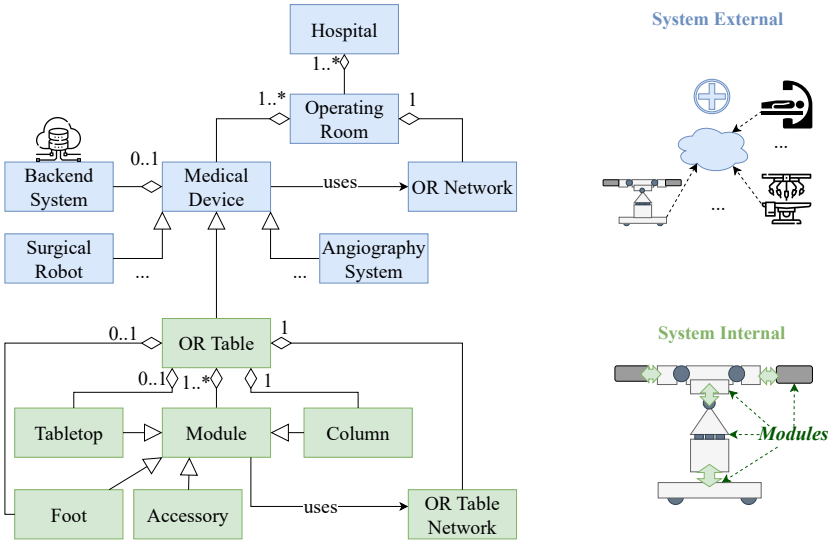


Figure 3.1: Class diagram of the system structure of a cyber-physical (Chapter 2.7.4) modular OR table (Fig. 2.29) connected to its OR environment (using surgery icons designed by Linector from Flaticon)

step-by-step in the automotive domain. Moreover the guidelines and standards for medical devices only necessitate a threat model and a corresponding risk analysis following ISO 14971. As a result, misinterpretation can lead to insufficient analyses and measures.

TARA processes that are established in the automotive industry can be adapted to the medical field. Although there are several possible methods and frameworks for a TARA execution (Chapter 2.2.1), HEAVENS 2.0 is considered one of the most suitable for medical devices by the author [PHS23] as it covers the ISO 14971 risk analysis in combination with the threat model in a single process (Fig. 3.2). Additionally, the proposed TARA contributes to the first step, "*Identify*", of the National Institute of Standards and Technology (NIST) cybersecurity framework for critical infrastructure [Nat18] in terms of medical devices.

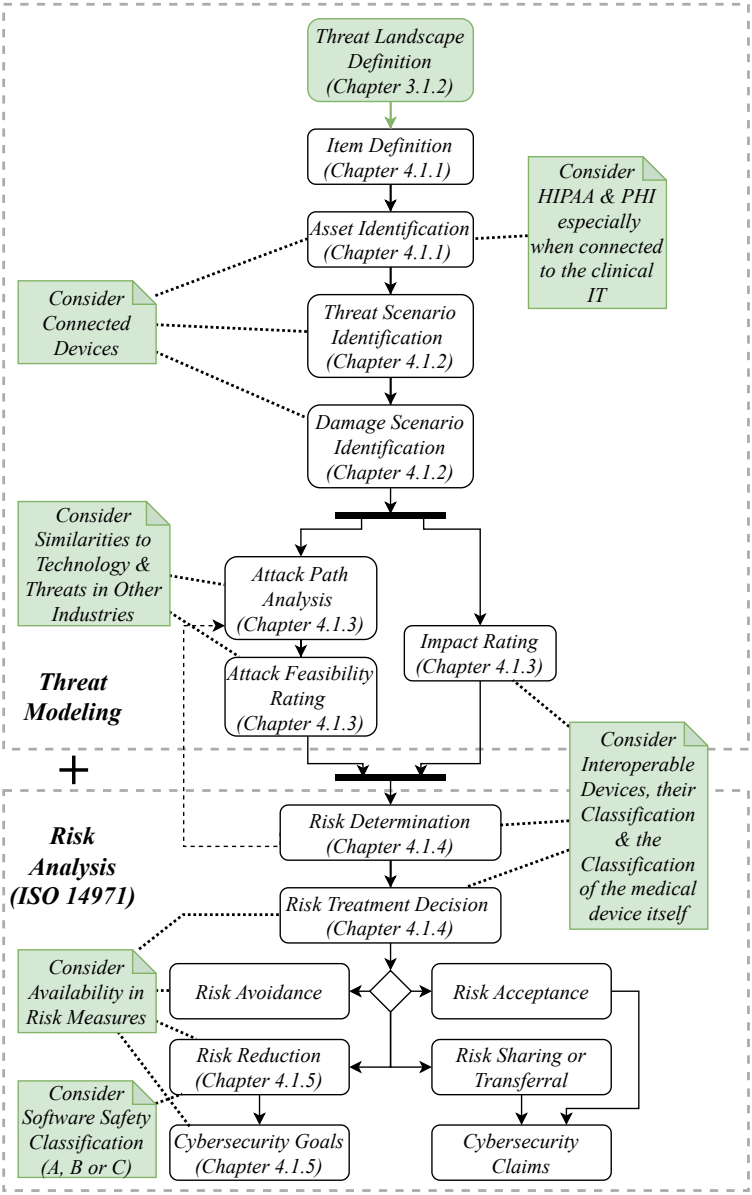


Figure 3.2: Activity diagram for Heavens 2.0 workflow based on [LAO21] considering a medical context (green) [PHS23]

Therefore, the HEAVENS 2.0 process must be adapted to match the needs of the medical field. First, the threat landscape must be defined (Chapter 3.1.2) since it differs due to the heterogeneity of medical devices. Once these threat landscapes are identified, they can be utilized for medical devices operating in similar environments, such as the OR. It is essential to consider HIPAA and PHI requirements[1], even if the connected devices do not generate relevant personal data. This is because connected devices may possess assets that require protection. Clinical IT systems or devices with higher medical device classification (Appendix A.1.1, Table A.1) can be potential targets. They must be protected accordingly since the medical device itself does not have to be the primary target of protection (Fig. 3.2 & Chapter 2.2.2).

3.1.1 Election of Standard for External Communication

The required rough architecture for further analysis forces a decision on the architecture of the unknown entity OR network (Fig. 3.1) based on fundamental requirements (Table 3.1). As open communication standards for medical devices in the OR are still in their infancy, the trend and increasing importance can not be ignored (Chapter 2.7). Whether they replace the proprietary solutions is unclear, as this depends on the medical device manufacturers. Yet, they are likely to be influenced by these open standards, and as medical devices start supporting interfaces, e.g., for SDC, proprietary OR integration system manufacturers will benefit if they start supporting these interfaces, too. Furthermore, they can focus more on providing new solutions that create value for the customer and patients instead of spending resources on supporting various manufacturer-dependent protocols.

Although interoperability today primarily relies on proprietary protocols, they are unsuitable for future challenges such as increased connectivity and security (Chapter 2.7). Hence, the connection to other devices is designed based on an open, manufacturer-independent protocol standard that is considered more future-proof

[1] Attack target types: access to a *patient's health data* or institution's data, Chapter 2.2.1

(Requirement 58). Therefore, to enable interoperability with other medical devices (Requirement 57), the options SCOT (Chapter 2.7.1), MDPnP (Chapter 2.7.2), and SDC (Chapter 2.7.3) remain.

SCOT systems are already on the market, but they do not target all medical devices in the hospital as SDC or MDPnP do because they focus heavily on the HOR [KSA+18]. Furthermore, plug-and-play, facilitated by using SOA in SDC or MDPnP, is not supported. Thus, new devices introduced in the system need changes at the code level [OMIM18]. In addition, there is a lack of published information, which hinders an accurate indication of the degree of maturity.

Table 3.1: Requirement overview for external interfaces (x: fulfilled, 0: unfulfilled, -: unknown)

Requirement	SCOT	MDPnP	SDC	Proprietary
Life Cycle overarching Interoperability (Req. 57)	-	-	x	x
Plug-and-Play (Req. 25)	0	x	x	0
Intended Use (Req. 51)	x	x	x	x
Unified Standard Interface (Req. 58)	-	-	x	0
Industry Adoption	x	0	x	x
Σ	2	2	5	3

Reference implementations of MDPnP and SDC are publicly available, and with sdcX by Surgitaix AG, a commercially available library for SDC exists [Sur22]. Also, Vector Informatik GmbH, known for its automotive solutions, is starting to contribute to this standard [Vec21]. Furthermore, with the anesthesia platforms "*Atlan*" and "*Perseus A500*", Dräger launched the first device on the market in 2020 that supports SDC [Kuc20]. Although both are designed as SOA, they use different middlewares since MDPnP relies on DDS (Chapter 2.7.2), generating less overhead for embedded devices than MDPWS, used in SDC. Moreover, the structure of SDC allows using other transport technologies (Chapter 2.7.3), such

as DDS (Appendix A.9.4), without influencing the overlaying layers. Therefore, the manufacturers' implementation will not be affected [Kas20]. In addition, SDC has already reached the level of a technical specification and standard [Kas20], which supports retaining the intended use of the devices (Requirement 51). Thus, SDC is chosen as a standard for external communication (Table 3.1). Additionally, Berger et al. propose an integration of SCOT with SDC [BRS+19], and cooperation with either MDPnP or SDC is also considered possible by Okamoto et al. [OMIM18]. Thus, future advancements in supporting SCOT or MDPnP devices will remain possible. Yet, compatibility of safety, security, and interoperability is not sufficiently investigated, and a *"high threat potential"* must be assumed in a system networked with SDC [Kas20].

3.1.2 Threat Landscape for Medical Devices in the OR

In future OR networks, several possible attack paths must be considered (Fig. 3.3). To identify these, threats discovered in the past in the medical field and other domains with safety-critical devices are examined (Table 3.2) and summarized in a threat landscape [ENI23] [ENI22]. Furthermore, identifying potential threats has become increasingly important due to reported vulnerabilities and cyberattacks on hospital equipment and medical devices in the past [PHS23].

Therefore, these entry points could allow an attacker to gain access to an MCPS:

1. External Storage Devices: External interfaces such as Universal Serial Bus (USB) ports can be used to charge smartphones and to exchange data with USB storage devices, which could be compromised (Threat 1).

2. Diagnostic & Maintenance Tools: The diagnostic and maintenance interfaces can be compromised and provide access to update and configuration functionality. In the automotive industry, interfaces like On-Board Diagnostics (OBD) can be exploited by attackers to gain unauthorized access to vehicle systems and sensitive data, especially during OTA software updates, when malicious malware can be transmitted [MTK19].

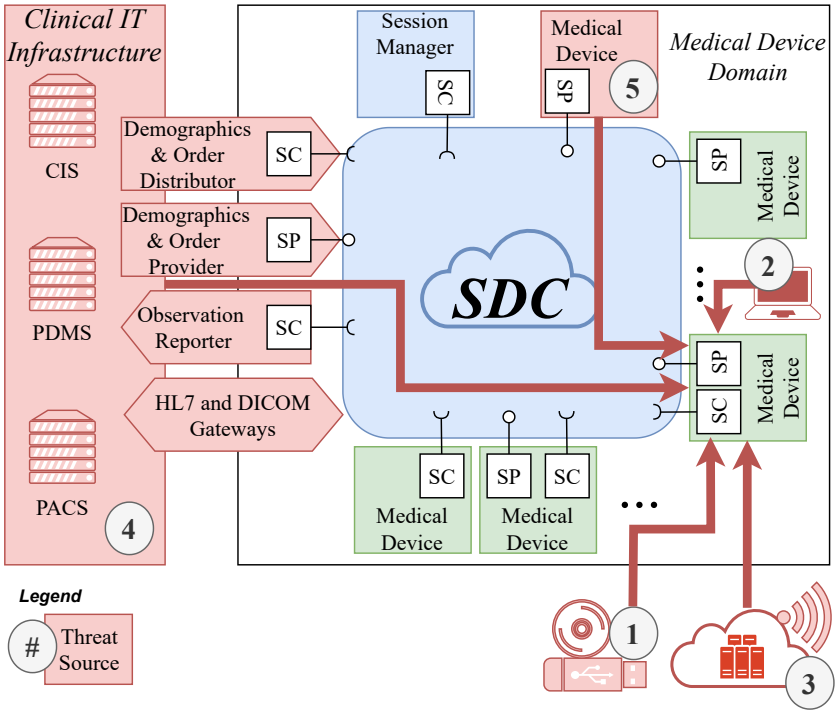


Figure 3.3: Threat landscape [PHS23] on layer 5 (*External System Connections*, Chapter 2.2.2) for a medical device (Chapter 2.1.1) within an SDC network (Chapter 2.7.3) based on the overall OR.Net concept for SDC (Fig. 2.32) and a cyber-physical OR table (Fig. 3.1)

3. Backend Systems, Internet Connection & OTA-Communication: Since manufacturers want to enable updates in the field and collect life cycle data, devices in the OR establish connections with backend systems through the Internet, potentially serving as entry points for attackers (Threat 5). Furthermore, with the elimination of the need for physical access in OTA communication methods like WiFi or Bluetooth, the attack surface expands, potentially enabling access from outside the hospital (Threat 3, Threat 4).

4. Clinical IT-Infrastructure: In case the clinical IT infrastructure was compromised similar to the Wannacry attack (Chapter 1), these attacks in the future might spread into the ORs too (Threat 1, Threat 2, Threat 5).

5. Connection to Compromised Devices: Due to a shared attack surface, other interconnected medical devices that may be compromised pose a threat as they use the SDC network (Threat 1, Threat 5, Threat 6).

Table 3.2: Examples of medical device cyberattacks and threats [PHS23]

Threat	Year	Description
1	2016	Ransomware attack on Hollywood Presbyterian Medical Center, leading to the shutdown of computer systems [Win16]. Two additional hospitals in California [Hea16] and one in Canada [Pil16] were targeted in the same year. WannaCry ransomware attack 2017 targeted specific gantry and robot imagers, which were infected through methods like malicious emails or infected memory sticks used by clinicians [Bun17a].
2	2017	Cleaning and disinfection equipment vulnerability discovered, potentially allowing access and data manipulation during a network attack [Bun17b].
3	2019	Insulin pumps recalled due to the potential for remote dosage adjustment by attackers [U.S19b], [KH19].
4	2019	Certain models of implantable cardiac devices, clinic programmers, and home monitors are found vulnerable to cyberattacks [U.S19a].
5	2019	Sterilizers found to be potentially manipulable via remote access [Bun19].
6	2020	Vulnerabilities discovered in certain models of central stations and telemetry servers, potentially allowing remote control and interference with alarms [U.S20].

3.2 Requirements for Anomaly Detection in Interoperable and Modular OR Tables

The system requirements are not further refined by software or hardware requirements, so this abstraction step is neglected in the following. Thus, steps 3, 4, and 5 for hardware and software of the V-Model (Fig. 2.4) are considered fulfilled with the system requirements (step 1) and system design (step 2). A summarized requirements list for a modern OR table can be found in Appendix A.10, derived for six use cases. In addition, the functional requirements derived are structured as activity diagrams (Fig. A.43) that model each use case (Fig. A.42, Appendix A.10). Only requirements relevant to the research questions (Chapter 1.3) are discussed and referenced here.

Accessories (Chapter 2.1.1, Definition 6 & Chapter 2.6.2), which extend the functionality, and OR table modules such as columns and tabletops, which contain core functions and thus are medical devices themselves, can be differentiated. In the following, accessories and OR table components will be considered generically as *modules*. Furthermore, reconfigurability is limited to the exchange of these modules. The exchange of sensors, actuators, and ECUs, as proposed by Stoll et al. [SGS+21a] or Ramesh et al. [RCS+23], and the dynamic distribution of software components, as proposed by Stoll [Sto21], is not considered here.

3.2.1 Additional Functional Requirements

Additional functional requirements (Table 3.3) are a consequence of the trends of connectivity, interoperability and automation (Chapter 1.3). The ability to exchange modules is not new, but its requirements will change with these trends. First interoperability examples for OR tables have entered the ORs for roughly 20 years now in the form of HOR (Appendix A.1.2). The broader scope of modularity and interoperability of OR tables and OR as a whole, including more than two devices, is a new aspect in this context. The possible combinations of OR table modules and the devices used in the OR (Fig. 3.1) will rise.

Table 3.3: Overview of additional/changing functional requirements (Appendix A.10)

Req. ID	Requirement Title
Req. 10	Synchronize Movements
Req. 11	High Precision Positioning
Req. 13	Provide Joint Positions
Req. 14	Provide Patient Weight
Req. 15	Provide Patient Positioning
Req. 16	Provide Collision Information
Req. 17	Provide Remote Control Command
Req. 21	Exchange Module
Req. 23	Radiolucent Module
Req. 24	Plug and Play Module
Req. 25	Plug and Play Device
Req. 58	Unified System Interface

Current approaches of certifying a combination of two interoperable systems as one medical device and integrating all proprietary interfaces from different manufacturers into one platform (Chapter 2.7) will no longer be possible with a rising number of devices, including all variants and possible configurations. State-of-the-art solutions show that manufacturers cannot easily exchange the core system of a device once the interface is integrated and the product is already on the market. An example of these are OR table systems integrated with angiography systems of different manufacturers: Different manufacturers cannot develop the systems independently and, as a result, require an additional coordination since standards such as AUTOSAR are missing. Thus, technical standards chosen during design will be used for several decades. This, in turn, also leads to increased product sustainment efforts over the products' life cycle.

Because medical devices will be evaluated on their ability to integrate data about the patient and other devices in the hospital in the future [TEMH$^+$20] (Chapter A.10.3, Fig. A.46, Requirements 13, 14, 15 & 17), the system must provide a unified interface representation to other devices while adapting to its current configuration (Requirement 58). This allows hospital operators to choose the most suitable product freely, as the functionalities are automatically conveyed through a plug-and-play approach (Chapter 3.1.1). The system's architecture should support the interoperation with other systems to enable future assistance systems (Chapter 2.3.3), increase the safety of a system, and enable new clinical applications to improve surgical procedures.

Furthermore, medical devices like angiography systems (Appendix A.1.2) require the current position of the OR table (Fig. A.49, Requirement 13) to scan the patient correctly and to avoid collisions (Requirement 16). During some of these scanning procedures, synchronized movements with the OR table are necessary (Requirement 10). To enable not only conventional intraoperative fluoroscopy (2D) but also the creation of 3D reconstructions through the use of intraoperative Computer Tomography (CT) technology [TEMH+20], high precision of the OR table positions will be increasingly more relevant (Requirement 11).

To adapt to different surgical disciplines (Chapter 2.6), the OR table must allow the exchange of modules (Fig. A.48, Requirements 21 and 23). Since nurses mostly do the preoperative set-up of the medical equipment [PDDL15], the exchange of modules should be plug-and-play (Requirement 24). Thus, the system could support complicated procedures or constraints, such as the combination of configurations for different situations dependent on surgical discipline and patient [GEKW22]. In addition, usability aspects consider that new technology will only be successful if non-technophile personnel can use and install the devices [PDDL15]. Furthermore, since unpredictable medical situations may arise during surgery in which a device needs to be dynamically integrated into the OR, the capabilities of this device and the OR table must be exchanged during runtime [DH12] [PDDL15] (Requirement 25).

Since OR tables take a central role within the OR (Chapter 2.6), it is in a predestined position to provide information about the patient (e.g., position or weight) (Requirement 14) used for monitoring systems (Chapter 2.3.3). This information is valuable for patient modeling and simulation (Chapter 4.5.3), which will be necessary to enable closed-loop controls and safety analysis of scenarios [LSC+12].

3.2.2 Additional Non-Functional Requirements

As non-functional requirements are more of a technical concern and also result from technical and medical standards and regularities (Chapters 2.1.3 & 2.2), they are usually not meaningfully assignable to use cases. Therefore, they are analyzed and summed up as a requirement list structured in system/quality attributes (Appendix A.2.2) in Appendix A.11 (relationships in Fig. 3.4).

Communication capability, confidentiality, integrity, performance, secrecy, reliability, security, safety, traceability, and usability are central non-functional requirements in medical and healthcare systems [Bal11]. Similar to other CPS, availability, authentication, and physical/administrative security must be guaranteed [AVSL11]. Not all of these can be targeted at the same demand (Appendix A.2.2). Therefore, some significantly impact patient safety and patient care. Consequently, it is necessary to determine how these quality characteristics will be achieved to contribute to architecture and design. Reliable and safe medical devices must ensure that the patient is not exposed to unacceptable risks [PDDL15] (Chapter 2.1.3). Thus, single fault safety is mandatory for functions that pose a hazard, like movements (Requirement 33) [IEC20].

As the hospital and, thus, the clinical staff are required to use the device according to the manufacturer's instructions [KAKA06], this will become more demanding as medical devices become more functional and connected. For an OR table, the user must know which modules can be combined, as not all combinations are permitted (Chapter 2.6.3). Afterward, it has to be decided on which combinations of accessories are allowed at which patient weight - keeping in mind that some maximum OR table positions are not allowed with heavy patient weight to prevent damage to the table, deadlocks due to overload of the different joints or even tipping of the whole OR table (Chapter 2.6.4). This might threaten the patients' health (Requirement 36). Furthermore, it is necessary that these support systems are reliable and do not have the opposite effect so that the system state as input for these support functions must be monitored for plausibility (Requirement 37, Chapter 4.1.5). Also, the plausibility of the system state of interoperating systems should be checked in combination, leading to the necessity of anomaly detection

(Chapter 2.3). Thus, each device must monitor its system state, but the system states must be checked for plausibility with each other, e.g., during synchronized movements (Requirement 10). Hence, both systems must have the same reference point in the coordinate system to avoid collisions (Requirement 38).

Furthermore, the clinical staff must "maintain a medical product logbook" and "report any incidents" concerning the product [KAKA06]. Manual logging is time-consuming, error-prone, and lacks practicability. In addition, documentation is already a burden, with an average of 3:50 hours per day spent by hospital staff in surgical departments. The trend is rising [HIM15], so even if the clinical staff notices an incident, minor incidents, in particular, may not be reported directly because of the reporting effort involved. Thus, automated documentation (Chapter 2.3.3) by data collection and detection of anomalies in terms of incidents will improve the safety and security of the system (Requirements 35 & 44). IEC81001-5-1 [IEC21] demands activities to gather and review information about software vulnerabilities to monitor incidents [IEC21]. Thus, it is necessary to enable the collection and analysis of data since manually reviewing all data generated by all systems in the field creates an unmanageable workload (Requirement 35). Furthermore, the IEC81001-5-1 [IEC21] also specifies activities during the whole life cycle to continuously improve security development, including security defects already deployed to the field. In addition, regulators and product users must be informed of vulnerabilities at an early stage [IEC21]. Besides a TARA (Chapter 2.2.1), an analysis of the products in the field must be executed (Requirement 41).

Since cybersecurity is already seen as an essential quality feature by customers in the automotive industry [OJB+20], this will likely be the case for medical devices as well. During surgery, personal health data (Chapter 1) is collected and shared over the OR network between the medical devices, which requires them to be HIPAA compliant (Chapter 3.1). Apart from legal requirements, it is essential to secure this to protect patient privacy and prevent attackers from compromising data, which can lead to an incorrect diagnosis [AVSL11]. In addition, manipulated data may lead to wrong control of medical devices or even direct control of a medical device. Therefore, unintended access must be prevented to keep the networked OR safe and secure [PDDL15].

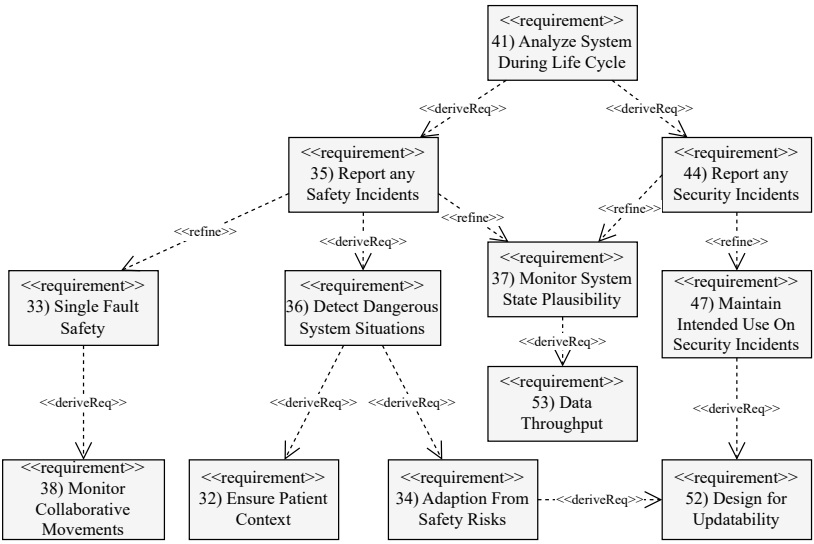


Figure 3.4: SysML requirement diagram summarizing the relationships of new non-functional requirements (Appendix A.11)

In some cases, risk mitigation of threats is realized through technical measures related to a system's intended use or essential functions [IEC21] (Requirement 47). This can be, for example, when overly complicated and cumbersome authentication procedures result in delayed urgent care or when computation-intensive encryption algorithms drain batteries faster in a medical device. For an OR table, this means that the movements after an interruption of the power supply [IEC16a] cannot be fulfilled anymore, decreasing the availability of the systems.

As safety risks discovered after deployment need to be solved quickly (Requirement 34) [PVR+22], risks resulting from security threats require faster and more frequent updates [RAMG20], e.g., for the ISAC (Chapter 2.2.4). Thus, it is advisable to design the system architecture to adapt affected components quickly (Requirement 52). Also, the IEC 81001-1-5 requires the manufacturer to establish life cycle activities in their product development process regarding security updates and patching for health software [IEC21].

The bandwidth and real-time guarantees provided by an E/E architecture influence the safety and security of a system [PDDL15], also in case of anomaly detection in the form of IDSs or Network Intrusion Detection Systems (NIDSs) (Chapter 2.2.4). Low data rates are typically required for control data of medical devices such as motor controllers. These kinds of data are subject to strict time constraints determined by the potential safety consequences for the patient if the constraints such as maximum allowed delay or jitter are violated (Table A.21). Thus, the used communication protocols (Fig. A.21) need to provide data throughput for control data of at least 100 kbit/s at maximum of 10 ms latency and for non-control data of 1 Mbit/s to 10 Mbit/s at 50 ms to 1 s [PVR+22][PDDL15] (Requirement 53). For interoperability scenarios such as synchronized movements (Chapter 2.7), it must be ensured that the medical devices are connected to the same patient [LSC+12] (Requirement 32).

3.2.3 Requirements from Organizational and Technical Constraints

Organizational constraints, e.g., due to hospitals, or technical constraints because of, for example, legacy modules and devices, limit the implementation options (Fig. 3.5). Since existing hospitals usually have legacy modules from other OR tables in their inventory, they have an interest in continuing to use these rather than replacing all their equipment. In addition, using existing work processes and infrastructures is mandatory, and adjustments are only tolerated if the effort and costs are reasonably related [CVK+18]. Thus, legacy modules should be compatible with newer ones over their life cycle (Requirement 49) to profit from the economic advantages (Chapter 2.6.4).

An example is the Otesus system OR table (Chapter 2.6.3), which was designed to be compatible with the Alphamaquet system released in 1995, which used an 8-bit controller (Fig. 3.6). In 2010, a new Otesus column was introduced, which incorporated a 16-bit controller, and a new generation of tabletops based on 32-bit controllers was launched in 2017. This shows that when a module is

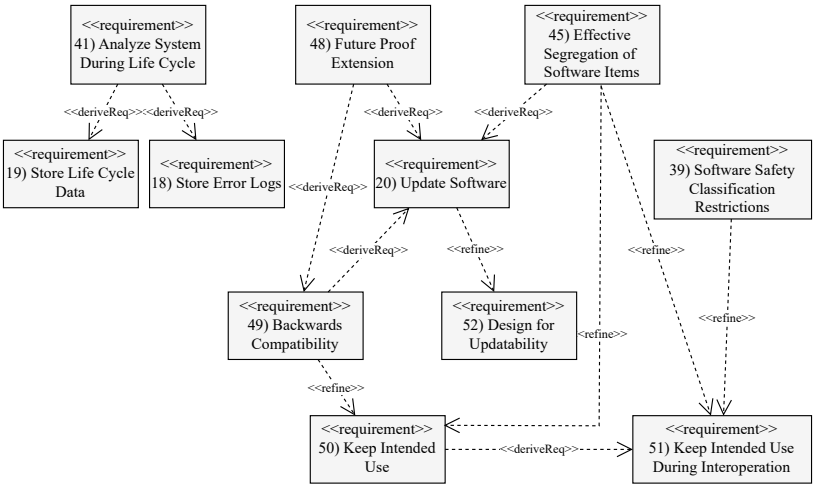


Figure 3.5: SysML requirement diagram summarizing the relationships of relevant requirements from organizational and technical constraints (Appendix A.10 & A.11)

developed, it is unknown which modules will have to be compatible in the future, which technology will be used, and what the corresponding interface provides (Requirement 48). At the same time, this compatibility is also required for other connected medical devices as requirements change over the device's life cycle (Chapter 2.7).

In addition, legacy devices and modules need to be integrated into an OR network without breaking their existing certifications (Requirement 50 & 51), which may be adversely affected by interoperability functionalities other than the intended use [PDDL15]. This is more challenging due to the long life cycles of 15 - 20 years [PVR+22] of POC medical devices.

This is in contrast to current automotive systems, which typically do not allow users to exchange modules during the product's life cycle, especially not modules developed before or after the initial product development. Furthermore, the OR can be characterized as a SoS, where its systems are represented by the medical devices it comprises (Fig. 3.6). Considering the modules within the OR table, it can be further decomposed into a system-of-subsystems, which also applies to

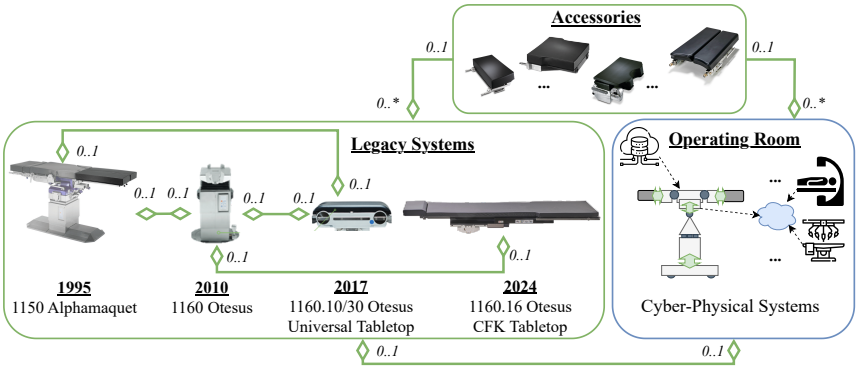


Figure 3.6: Example for a system OR table product line from Getinge/Maquet with possible modular combinations (using surgery icons designed by Linector from Flaticon)

other medical devices in the OR. This makes ensuring compatibility in an OR and its equipment a unique challenge due to the necessary individuality and adaptability that must be maintained over decades in a context where lives are at stake (Requirements 48 & 49). Additionally, the introduction of updatable software for each system/subsystem adds another dimension (Chapter 2.6.4), increasing the potential combinations (Requirements 20 & 52).

Furthermore, the IEC81001-5-1 standard demands a "timely delivery of security updates" [IEC21], and, thus, fast updates in the field are required (Requirements 20 & 52). As the demanded activities also imply a policy for the time frame and the potential impact of a vulnerability, the pressure on the manufacturer to deliver solutions to close vulnerabilities rises. In the automotive industry, OTA updates are considered to speed up the distribution of updates [NSN08]. As these bring additional security concerns, they must be thoroughly evaluated as they are not mandatory to enable fast updates due to the static networks in ORs. Since some medical devices frequently alter the surgical environment, this can be considered more practical as it saves valuable time in the OR.

In the automotive industry, cars could be returned to a workshop for diagnosis and updates, but this practice is uncommon in the medical field. Consequently, medical devices, particularly POC (Chapter 2.1.1) devices, must be updated in the field. Service technicians play a crucial role in this process, as ensuring compatibility of the software variants and versions across various OR table modules within the OR and with other devices presents a significant organizational challenge. Today, coordination of updates involves manual efforts between the medical device manufacturer, service technicians, and the hospital. As interoperability continues to evolve, managing this will become an increasing burden in the future.

In the context of interoperability, it needs to be ensured that the SSC (Chapter 2.1.3) of different medical devices only uses functions they are allowed to use (Requirement 39). Class C software may invoke *writing* functions on all other devices, but Class A software may only *read* from higher and write to equal class software [Eur17]. This also applies to the OR table modules (Chapter 2.1.3) and causes the need for an adequate segregation of software items. The IEC81001-5-1 demands this as well [IEC21], as an cyber attack (Chapter 2.2) will be restricted to the compromised software (Requirement 45).

Additionally, the MDR [Eur17] imposes requirements for the systematic and active collection of post-market information about a medical device and its experience with the device. The collected data and the experience from corrective or preventive measures shall be used to update technical documentation, especially risk and clinical evaluation. Therefore, they must cooperate with national authorities. Furthermore, each statistically significant increase in situations leading to unacceptable risks must be reported. Thus, it is necessary to implement processes and enable the collection of this data on technical sites (Requirements 18 & 19) and automatic analysis of *anomalies* (Requirement 41).

In the SoS context, the hospital is more similar to a production line (see also Appendix A.9.2) where the strict separation of hazardous devices (such as robots) is not universally feasible, for instance, by enclosing them in a cage. Consequently, the architectural requirements of an OR table can be compared to those of a robot with an emphasis on safety and security requirements, such as for an automotive

system, within a production line. In addition, the hospital is then an entire production hall with different production lines (ORs), whereby - aside from ethical concerns - in this case, the patients are the “products”. Therefore, inspiration for technical solutions can be found in other domains, but transitioning these to the medical field without further adoption is insufficient. This also shows that there is not a single domain that is predestined to have the best solutions that must be adapted for medical devices.

3.2.4 Anomaly Detection Function

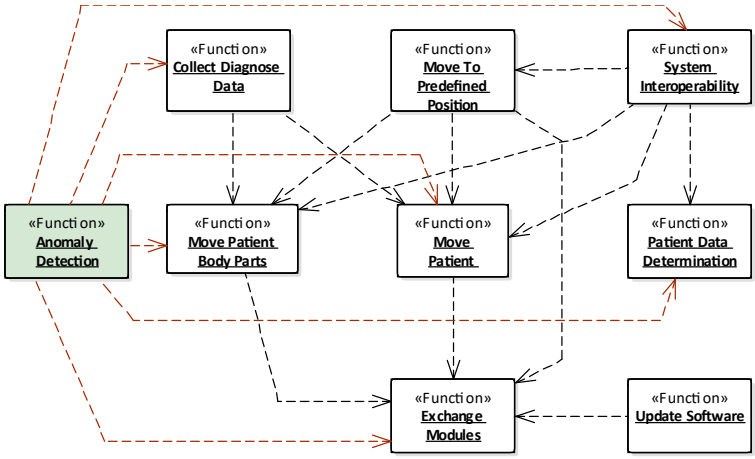


Figure 3.7: Function dependencies (Appendix A.10.7) for anomaly detection (Chapter 4)

The functional requirements can be consolidated into functions (Chapter 2.5, Fig. A.13) and their corresponding features (Chapter A.10.7). These form the foundation of the modular and interoperable OR table architecture. The function *Update Software* is not considered further, as secure software updates are an extensive field of research and, therefore, cannot be sufficiently addressed within the scope of this dissertation. Instead, reference is made to Guissouma et al. [GSS21b] [SHG+21] [GKMS21], who examined software OTA updates for safety-critical

product lines in the automotive industry and CPS in general. The requirement analysis shows that *Anomaly Detection* is a crucial element for the interoperable and modular OR table and interacts with all the remaining functions (Fig. 3.7). *Move To Predefined Position* is indirectly covered through its dependency on the other functionalities. Furthermore, the anomaly detection function is considered as the functional safety and security concept (Fig. 2.4).

To improve the safety of a system, redundant sensors (Requirement 33) have been applied in OR tables and other medical devices to cross-check their measured physical quantities (Chapter 2.1.3). Nevertheless, several obstacles and disadvantages exist for this approach:

1. **Different Sensor-Types**: Two different types of sensors must be used to exclude a hidden systematic error, which affects the system's mechanical design and can be challenging to implement. For a position sensor, for example, this can be realized with an absolute sensor like a potentiometer and an incremental sensor like an incremental encoder.
2. **Costs**: Each additional sensor increases material and production costs.
3. **Sensor Failure**: If a sensor fails, the system must run in a degraded mode to reach a safe state while ensuring system availability. This can also be caused by false alarms, e.g., due to different sensor quality or aging.
4. **Susceptibility to Defects**: As the number of mechatronic components increases, a system's availability can decrease as more components could fail, which further increases costs.

If a sensor error occurs, the need for a service technician can further reduce availability. A third sensor could be introduced to enable Triple Modular Redundancy (TMR) so that the incorrect one can be determined based on the functioning ones to decrease the susceptibility to defects and increase the availability again. This negatively affects all the other points mentioned. Furthermore, the presence of redundant sensors does not provide the capability to identify whether an attacker has tampered with the corresponding values in the system's memory. Consequently, due to security considerations, this measure will prove ineffective.

3.3 Novel Approach for Anomaly Detection using Hybrid Checks

The signal plausibility monitoring, as proposed by Weber (Chapter 2.3.3), focuses on anomalies contained in an isolated signal, such as *Plateaus* or *Positive/Negative Step Plateaus* (Table A.9). These anomalies do not consider the system context, so the correlation between different signals is neglected. Attackers can exploit this vulnerability by manipulating these signals individually in a plausible way similar to the Stuxnet attack [Kus13]. This gap is to be addressed by the system outlined below. Therefore, the anomaly detection approach considered here can be classified as a *plausibility* and *consistency* sensor (Chapter 2.3.3, Table 2.2) for contextual anomalies (Chapter 2.3).

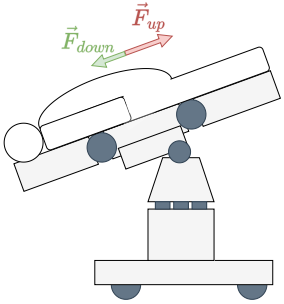


Figure 3.8: Scenario of incorrect Trendelenburg angle during longitudinal shift movement

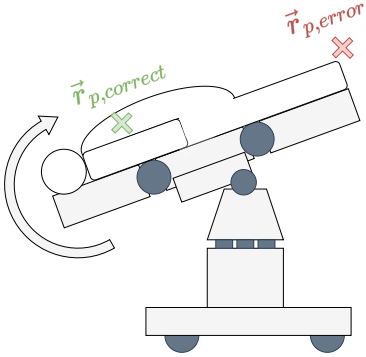


Figure 3.9: Trendelenburg movement with erroneous working patient position $\vec{r}_p$

For example, the motor current or torque signal of an OR table for the longitudinal shift can be plausible when observed alone, although it is implausible in the system context. In case the Trendelenburg position is at an angle >0°, the movement of the longitudinal shift against the slope results in increased motor torque and current than with the slope (Fig. 3.8, Force $\vec{F}_{up}$ > Force $\vec{F}_{down}$) [Kin22]. Another example is a working load measured on the head side of an OR table, which would

result in increased torque needed to drive reverse Trendelenburg than to drive Trendelenburg (Fig. 3.9). This indicates that the position value of Trendelenburg has been tampered with or that the corresponding sensor has a defect.

Deriving rules in the traditional way of programming leads to discontinuities. They are challenging to design comprehensively (Chapter 2.3.2) due to the dimensionality and continuous value range of possible physical quantities. This excludes these types of anomalies from being detected with static checks. For a rigid body system with rigid loads (Chapter 2.4), a model-based estimator such as KFs (Chapter 2.3.1) can be created to estimate the current state based on the system's dynamics. Current measurements can then be compared to the estimated values to detect anomalies (Chapter 3.4.3). Hence, plausibility checks based on manually created and mathematically described physical models are referred to in the following as *dynamic checks* [PZSS24]. Therefore, the plausibility of the joint data can be checked by monitoring the deviation of the joint dynamics from expected behavior (Requirements 37 & 46). Thus, it is not sufficient to analyze the communication on the protocol layers but to inspect the content of the exchanged messages or services (Requirement 42), which enables independence from protocols (Requirement 56).

Classical approaches, such as cross-correlation with the expected signal, are challenging to apply since the signal searched for in the data depends on features such as weight, size, and positioning of the patient or current position of the OR table. When applying statistical methods such as the Z-score [HAZ00, KM09] that assume normally distributed data, most of the data retrieved with a non-normal patient are possible outliers. The challenge of static interpretability and generalizability increases with the dimension of the data, e.g., using multivariate z-scores. Even statistical methods such as the Inter Quantile Range (IQR), which are not based on normally distributed data, assume a probability distribution that does not adapt to individual patients. Since the n-gram approach by Rumez et al. (Chapter 2.3.3) is also based on the statistical evaluation of sequences, it is considered unsuitable. In addition, the analysis of the sequences only indirectly checks the payload's plausibility, so the adaptation to different patients cannot be realized. An approach based purely on machine learning bears challenges in obtaining the

necessary data in an OR or surgery context (Chapter 2.3.3). Thus, it is more challenging to apply collected data as training data in case the context is unclear, and it cannot be stated if the collected data represents the expected behavior of a system. In addition, statistical and information-theoretical methods are unsuitable for modeling sequential data [TMR20].

The challenges of anomaly detection for surgical robots (Chapter 2.3) also apply to an OR table since the human body - especially when anesthetized - is a deformable object manipulated by the joints of an OR table that is padded to avoid pressure sores (Requirement 54). The body, especially of heavy patients, deforms due to gravitational forces during movements of the OR table. Furthermore, as the OR table deforms under heavy patients, while the maximum patient load for a middle-class OR table is $\sim$250 kg (e.g., Getinge Maquet Meera [Get23a]), deformation also applies here. These deformable elements are intricate to model analytically, so the dynamic checks only work reliably in the range of physical quantities where the underlying modeling assumptions hold. This approach reaches its limits, especially in exceptional scenarios involving a wide range of potential patients, ranging from a 2-meter tall and 250 kg heavy person to a child, and taking into account the positioning of the patient based on the modular design of the OR table.

Each model has constraints as the most likely states are assumed based on standards and assumptions, e.g., by checking a measured value for its standard deviation. These approaches might be sufficient in terms of a technical and rigid system. Standardizing the human body is demanding due to the wide range of body shapes, sizes, proportions, and physiological factors known as anthropometric data. The shoulder width, for example, examined by Kilgore, varied from 1.5 to 3.25 head lengths for male subjects [Kil12]. This diversity presents a significant challenge in creating a straightforward and universally applicable standard. Furthermore, it is even harmful if technical systems are primarily designed and optimized for standardized human proportions: Tests with crash-test dummies today are mainly used in the automotive industry. For example, the current approval process in the European Union explicitly requires that car seat belts be tested on an average male dummy. This had the result that the likelihood for a woman to die in a traffic

accident is 18,3% higher [Har22]. Furthermore, this also means that individuals deviating from the standard human proportions may have an increased mortality rate in safety-critical systems if it is optimized for a standard human model. Still, adaptions must be applied to keep technical solutions realizable.

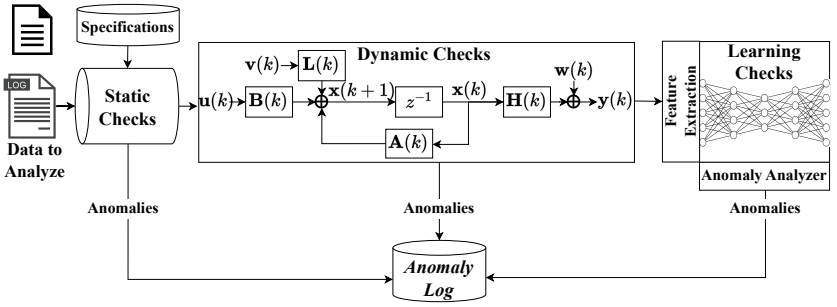


Figure 3.10: Anomaly detection concept based on [Web19] and extended with dynamic checks

Nonlinear behaviors are more likely to be determinable not analytically but via machine learning (Chapter 2.3.2) and, thus, via learning checks. Therefore, the concept for the *automotive observer* from Weber (Chapter 2.3.3) is used as the foundation for the anomaly detection in the OR table (Chapter 3.4.3) and extended with dynamic checks (Fig. 3.10). This is expected to help reduce the number of features required (curse of dimensionality [GBC16]) and still reduce the FPR [PZSS24]. The combination of dynamic and learning checks is called *hybrid checks* in the following. Furthermore, Weber only uses algorithms with reduced calculation effort due to the limitations of embedded devices [Web19]. This limitation does not apply in the following research, as distribution on different runtime environments is considered. For example, in a centralized architecture or in a backend system, computers can also execute algorithms with increased computing effort (Chapter 3.4.3). In addition, hard real-time is not required because no direct technical risk measures are yet associated. Backend systems help collect more meaningful data on incidents in the future. These are even more necessary when clinical personnel are unaware of security incidents, as attackers ensure they remain undetected (Requirements 35 & 44).

3.3.1 Dynamic Checks

When describing a system's behavior and all its occurring states mathematically as a set $\mathfrak{S} = \{\mathfrak{s}_0, \mathfrak{s}_1, ..., \mathfrak{s}_n\}$, the conventional states to be modeled via a physical system description can be written via $\mathcal{S} = \{\int_0, \int_1, ..., \int_n\}$, so that $\mathcal{S} \subset \mathfrak{S}$. By transferring this relationship to a singular state $\mathfrak{s}_t \in \mathfrak{S}$ in the specific moment t, relation through an addition of a disturbance is $d_t \in \mathcal{D}$ of the modeled system:

$$\mathfrak{s}_t = \int_t + d_t \tag{3.1}$$

The dynamic checks monitor $\int_t$ of $\mathfrak{s}_t$, while $\int_t \in \mathcal{S}$ is described with a mathematical model, such as a KF. If the first part of the equation is represented by a KF $\mathcal{KF}$ to predict physically explainable system behavior, the filter is parameterized through $\mathbf{A}, \mathbf{B}, \mathbf{Q}, \mathbf{H}, \mathbf{R}, \mathbf{L}$, (Table A.3) and receives an input tuple $\mathcal{U}, \mathcal{Y}$ (Table A.3) to predict the subsequent state $\hat{\mathcal{X}}$:

$$\mathcal{KF}_{(\mathbf{A},\mathbf{B},\mathbf{Q},\mathbf{H},\mathbf{R},\mathbf{L})} : (\mathcal{U}, \mathcal{Y}) \rightarrow \hat{\mathcal{X}} \tag{3.2}$$

In the case of UKF/EKF, the functions h and f replace $\mathbf{A}$, and $\mathbf{H}$; for UKF, the sigma function s is an additional parameter. The output of the KF can then be used to detect anomalies: When constructing the difference vector of the predictions or estimations of the KF with the measurements, a deviation score can be generated using l_k norms (Chapter 2.3.1), e.g., the MAE:

$$MAE_{state} = \frac{1}{n} \sum_{i=1}^{n} |\hat{\mathbf{x}}_i - \mathbf{y}_i| \tag{3.3}$$

If the deviation exceeds a defined threshold value for an iteration k, the corresponding state or measurement vector of that iteration is classified as an anomaly.

3.3.2 Learning & Hybrid Checks

Like Weber's approach [WKSZ18], the measurements can be directly used as inputs for the learning checks. A similar state estimation can be generated by choosing the same inputs as for the dynamic checks, e.g., for an LSTM. Nonetheless, as current pre-trained machine learning algorithms do not predict the next state based on past inputs, which would require online training, the time series assessment necessitates the examination of window slices. While exclusively relying on time series is not obligatory, outliers can still be identified. At the same time, limiting the information accessible to the model influence the algorithm's overall performance.

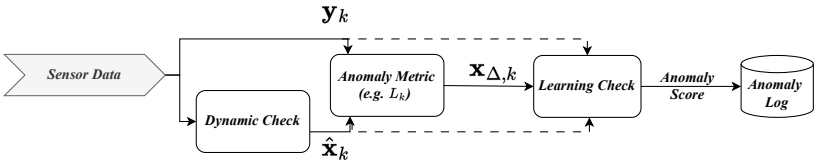


Figure 3.11: Hybrid check data flow diagram with sensor data input

Another approach is to combine dynamic and learning checks, forming hybrid checks. The dynamic checks cannot take the non-modeled behavior d_t of a system into account, so it cannot be readily determined by a threshold if the deviation from the prediction is abnormal. Therefore, data-based models such as AEs or LSTMs networks (Appendix A.5.2) can model this deviation. The data-based model $\mathcal{N}$ then only has to model d_t from $\mathcal{D}$ (Chapter 3.3.1) to describe the system behavior that is manually challenging to model mathematically:

$$\mathcal{N}_\Theta : (\hat{\mathcal{X}} - \mathcal{Y}) \rightarrow \mathcal{D} \tag{3.4}$$

With a KF, these inaccuracies of the system model on which the prediction step is based can be considered by a covariance matrix of the process noise. Therefore, it is unclear how to determine which error is plausible and which is an anomaly only by a threshold. As an anomaly metric (Fig. 3.11), the input can be chosen as

follows, when evaluating each state of the dynamic check individually and not as single anomaly score as the MAE:

$$\mathbf{x}_{\Delta,k} = \hat{\mathbf{x}}_k - \mathbf{y}_k \tag{3.5}$$

The difference between measurement and estimation is a reduction of the features needed as input for the data-based models (Chapter 2.3.2). If the raw estimation and measurement were not used, some aspects of the provided training data could not be learned. Therefore, the measurement can also be used alongside the estimation as input to predict the estimated in arbitrary combinations (Fig. 3.11). Nonetheless, the possible combinations of dynamic models with data-based models and the chosen features and window sizes, among others, are not examined but exclusively promising variants to compare. Therefore, AE, LSTM, and IF (Chapter 2.3.2) are suitable as learning checks (Table 3.4), especially since resource consumption does not play a significant role here (Chapter 3.4.3).

Table 3.4: Comparison of state-of-the-art machine learning algorithms for anomaly detection (adaption from [Koc22], Table A.8)

Requirement	**OCSVM**	**AE**	**DNN**	**LSTM**	**IF**
Unsupervised Learning	+	+[2]	+	+	+
Resource Intensive	0	-	-	-	+
Multidimensional Feature Set	0	+	+	++	-
Amount of Hyperparameters	0	0	0	-	+
Successful Adoption in Literature	0	++	0	+	0
Σ	1	**3**	1	2	2

The anomaly detection performance will be influenced, e.g., by the patient's anthropometric data. Thus, it is expected that a single trained model for all scenarios will not yield optimal results in all scenarios. Therefore, several trained

[2] Mathematically, AEs are no unsupervised learning algorithms, although they are suitable for these tasks.

models must be optimized for a specific value range and must be selected based on the current scenario. These ranges must be discretized, for example, based on the following criteria:

1. **Patient Position:** Body CoG & positioning pose
2. **Patient Anthropometry:** Gender, body shape, weight & height
3. **OR Table Configuration:** positions & mounted modules (Chapter 3.4.2)

3.3.3 Systems-of-Systems Extension in OR Networks

The anomaly detection approach is also scalable for monitoring, e.g., SDC data of multiple medical devices in an OR (Chapter 3.2.3). This leads to a holistic approach that considers security and safety on a broader scale of an OR context, as Lee et al. demand [LSC+12]. Furthermore, this is a gap in medical field research as current approaches only focus on the interface of a medical device or the direct communication of different devices [PRGS22]. In addition, combining complementary technologies can be harnessed by amalgamating redundant sensor systems within a network [PDDL15]. An example of this concept is witnessed in surgical navigation systems, wherein the concurrent utilization of electromagnetic tracking devices and optical systems demonstrates the exploitation of both technologies' advantages [PDDL15]. Therefore, safety and security requirements can be fulfilled more effectively and distributed in the SDC network (Chapter 2.7.3).

Due to the reduction of redundant detection systems, costs for individual devices are lowered. With service-oriented communication being prevalent in future OR networks through the usage of, e.g., SDC (Chapter 2.7), securing these will be increasingly challenging as the devices may enter or leave networks dynamically. Additionally, most of today's IDSs focus on traditional IT computer networks. Hence, NIDSs for CPSs are still in their infancy [ASW+21], and physical device or patient data are not of concern yet. Furthermore, machine learning-based IDSs are suitable for detecting anomaly patterns in medical device communication patterns (Tables 3.5 & A.10) and improving detection accuracy (Chapter 2.2.4).

Inspired by the distributed NIDS (Chapter 2.2.4), Host Intrusion Detection Systems (HIDSs) are deployed in each medical device to detect the current state of the hosting device and specific events (Chapter 2.3.3, Table 2.2). In addition, a supervisor NIDS (Chapter 3.2.4) is deployed on a backend system (see also Chapter 3.4.3). The HIDSs extract features from communication patterns and categorize anomaly patterns. Afterward, they communicate the information to the central supervisor NIDS, providing a context for the entire OR. By leveraging the HIDSs, the NIDS can consider the payload of the services and the device context, providing a comprehensive scenery of the OR's safety and security state. Furthermore, the patient is also integrated into this safety and security context, as the patient data in the medical device communication is monitored.

Table 3.5: SDC communication patterns [PRGS22]

Pattern	**Connection Multiplicity**	**Example**
Publish/Subscribe	1-N	Description/State Event Service
Request/Response	1-1	Get/Set Service, Context Service
Stream	N-M	Waveform Service

Also, the safety can be improved, e.g., for synchronized movements of medical devices (Requirement 10) as in the HOR, since systems move together while calculating the collision data of other devices with the ability to restrict or stop the other devices. At the same time, a third instance must ensure both systems' consistency to detect if a system has been manipulated for security concerns or to maintain availability during surgery through TMR (Chapter 3.2.4). Thus, a reference point for a common OR coordinate system must be introduced since operating multiple robotic devices requires a *shared coordinate space* [BUK+22] (Requirement 38), leading to a digital twin (Chapter 2.1.4) of the whole OR.

Figure 3.12 depicts a fictional scenario of different connected medical devices in the OR inspired by an interoperability example in [AGWL09] leveraging the communication patterns of SDC (Table 3.5) for anomaly detection: During an

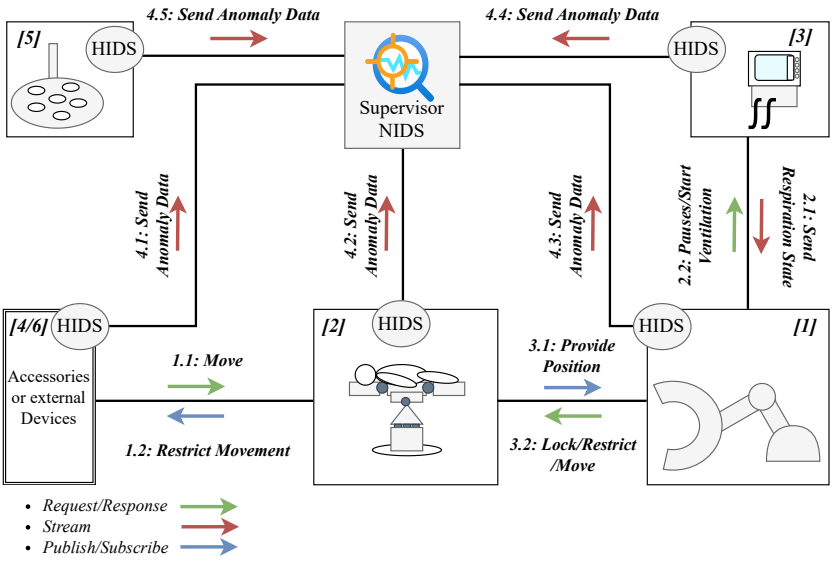


Figure 3.12: Communication diagram for an SDC network of different medical devices in an OR supervised by a distributed IDS [PRGS22]

imaging process, the angiography system (1) can restrict, move, or lock the movements of an OR table by Request/Response (R/R). This can avoid any potential collisions that may occur during the imaging process or prohibit any movements that may interfere with the accuracy of the imaging results. The OR table (2) provides its position and possibly restricted movement ranges to other devices via publish/subscribe services.

During the imaging process, the ventilator (3) can be paused and restarted to help suppress breathing movements while the patient is ventilated, improving the quality of the imaging results. One way to operate the OR table joints is using a connected remote control or foot switch (4), connected with the OR table through SDC or a proprietary protocol. Furthermore, a camera (5) integrated into an OR light can be used to detect an OR table movement or to determine the joint positions of the OR table. Thus, the movement state of the OR table is externally determinable to provide more context to the IDS. Since medical devices such as

OR tables have various different accessories, it is conceivable in a future scenario that an external accessory or device (6) might be plugged into an OR table or another device while using it as a gateway to the SDC network. In such a setup, generic SOA communication patterns or, e.g., the context payload of the services of different medical devices can be monitored for anomalies [PRGS22] (Table A.10).

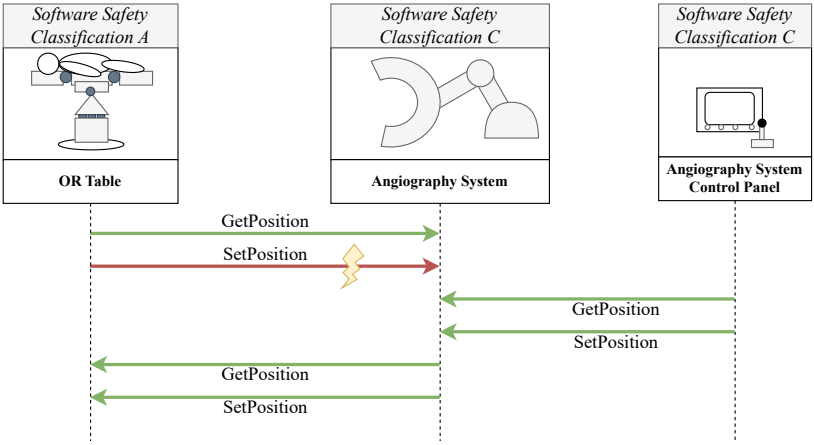


Figure 3.13: Sequence diagram for SSC-based control restriction

Furthermore, the specification of SDC and the medical device standards (Chapter 2.1.2) offer the basis for static checks [PRGS22]. In the context of sensor fusion in the OR, a camera in the OR, for example, can detect instances where the patient's actual position, weight, or height contradicts the information provided by the accessories [Pel23] [GHK+21]. It will likely involve identifying situations where the patient is in a reversed position. In case a medical device controls the data of a higher classified device, it inherits the classification. For example, it must be prevented that a service with class A invokes services with class C (Fig. 3.13) with write/control effects [Eur17] (Chapter 2.1.2).

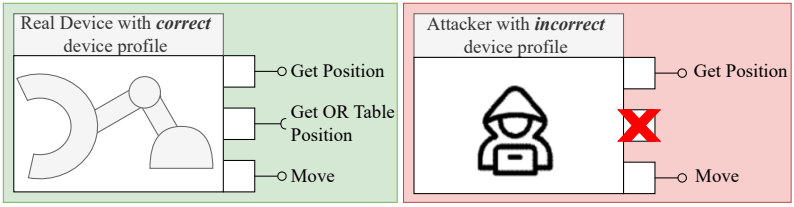


Figure 3.14: Anomaly in device profiles

In device monitoring and control, examining whether a device that employs the SDC extended device profile exhibits any anomalies regarding the services it consumes becomes essential. This entails investigating if the device presents an unusual assortment of services that deviates from the expected profile interface (Fig. 3.14). Furthermore, when considering the participation of different entities within the system, it becomes crucial to determine whether a medical device aligns with the specifications of a designated device role. This includes verifying whether they adhere to the role's defined characteristics up to a specific revision level (Fig. 3.14). Since monitoring these roles in HIDSs creates unnecessary redundancy and the information is available to every device in the network, monitoring in NIDS is more effective.

An alternative approach involves device-specific roles (Table A.22), realizing a *principle of least privileges approach*. Access would only be granted similarly to an Identity and Access Management (IAM) (Appendix A.7.4) upon satisfying the requirements of the device class, such as utilizing the SDC extended device profile as a role identifier. Therefore, each medical device would be classified with roles such as "OR table" or "Ventilator" that represent its intended use (Chapter 2.6). In that case, services that are not part of the intended use need individual handling. The operator could tailor this configuration, allowing for customization based on variations across different healthcare facilities.

A fundamental prerequisite for role-based approaches involves authenticating the devices and services. This authentication process involves IT efforts to create and maintain security certificates. Thus, the role-based approach, including the

authentication, necessitates that operators (typically hospitals) possess deep technical knowledge of the medical devices. They need to know which services each device requires. This task is impractical and burdensome as hospitals cannot currently be expected to integrate two individual devices (Chapter 2.7). In addition, these roles need to be defined in advance, and an anomaly-based IDS would still be required to keep this IAM up-to-date due to zero-day attacks. Furthermore, restricting access would counteract the idea of SDC to seamlessly make data available to all medical devices in the OR (Chapter 2.7.3).

3.4 New E/E and Software Architecture for Interoperable and Modular OR Tables

By leveraging E/E architectures (Chapter 2.5), the medical device industry can benefit from the established methods and practices of the automotive sector. This cross-domain knowledge transfer contributes to developing safer and more reliable medical devices. In addition, the medical device industry has no joint, dedicated approach to designing the E/E architecture and software architecture. Generally, medical device standards refer to the less related IT field in terms of software. Hence, medical device manufacturers must rely on established methods and tools from other domains or create these themselves, which is time-consuming and expensive. Therefore, the following architecture is intended to contribute to dedicated methods for the field of medical devices based on established methods from the automotive industry.

Considering current developments in areas such as robotics and automotive technology (Appendix A.9), a service-oriented architectural approach (Chapter 2.5.3) for medical devices can be derived, focusing on interoperability and modularity while considering medical constraints. Thus, function 5 (Appendix A.10.7) is supported by adopting an SOA middleware (Appendix A.9.4) due to the dynamic discovery of services (Requirements 24, 48 & 49). In addition, service-oriented communication helps to segregate software items (Requirement 45, Chapter 2.1.2)

and supports the development and deployment of updates [VOG+20] (Requirements 34 & 52). Also, criticism of SOA in medical devices regarding, e.g., issues of standards, medical errors, and adverse effects, as addressed in [NM07], has gone quiet lately.

Modeling approaches for E/E architecture can be applied to derive the design and architecture for a medical device (Chapter 2.5) from requirement analysis as well (Step 2, Fig. 2.4). The zone-oriented E/E architecture (Chapter 2.5.2, Fig. 2.19), currently on the rise in the automotive industry as suitable for SOA, is most comparable to the E/E architecture of state-of-the-art OR tables (Chapter 2.6). Hence, this may remain unchanged as this E/E architecture is still appropriate for OR tables. Furthermore, it best supports the modular design of OR tables (Chapter 3.2.3) since the interface design is limited to the power supply and connection to one physical bus. Using the centralized E/E architecture with domain controllers and networks, the OR table module interface design and cabling effort rise without a benefit. Domain-oriented and distributed architectures are not further considered, as they have already been deemed obsolete by the automotive industry (Chapter 2.5.2) and are not flexible enough for future challenges (Chapter 1).

Considering organizational and technical constraints (Chapter 3.2.3), an alternative solution would be to adopt a zone-oriented or even a centralized architecture for the entire OR. In this approach, devices would solely receive commands and control sensors and actuators through zone controllers (Fig. 2.19). Moreover, this design introduces a single point of failure and renders the devices unusable during network shutdowns, which is unacceptable from a safety and security standpoint. Furthermore, this contradicts the intent of SDC (Chapter 2.7.3), which aims to facilitate interoperability between medical devices without relying on a central coordinating instance.

Because CAN-XL (Appendix A.3.1) and Ethernet (Appendix A.3.2) are suited for realizing a service-oriented E/E architecture [PVR+22], the proposed architecture does not restrict to one of these as both are appropriate for an OR table. Additionally, today, there are no requirements for increased bandwidths, which can

only be reached with switched Ethernet architectures. Farzaneh et al. [HNNK14] evaluated different protocols' characteristics, such as transmission latency, extensibility, and costs, and concluded that ethernet-based protocols best match these requirements [PDDL15].

The connection to a backend system is treated as another external system, and this system is considered to realize requirements regarding data collection and remote updates (Requirements 18, 19, 20 & 35). Furthermore, the automotive industry already has solutions for these approaches, e.g., Security Information and Event Management (SIEM) systems [GSS21a], so it is not further examined here.

3.4.1 Proposal for an SDC Interface

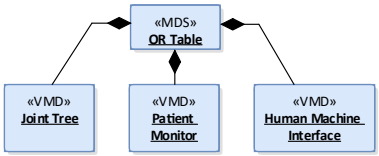


Figure 3.15: VMDs of an OR table SDC interface

The SDC context (Chapter 2.7.3) ensures that the devices are connected to the same patient (Requirement 32), and the structure of the MDIB provides the SSC of different services (Requirement 39). For configuration, SDC uses the initial so-called *commissioning process*, in which new equipment is introduced and tested in the OR, after which it can be used in surgery (Chapter 2.2.4). This is an advantage in the sense that configuration data can be exchanged in advance, which is usable in the RT-SDC real-time architecture (Requirement 53), enabling reliable plug-and-play during runtime [PDDL15] (Requirement 25). The compatibility of the devices can then be checked during this process according to the automatic testing presented in [DDP+15]. Furthermore, the clinical IT data is considered retrievable over the SDC network connection, as proposed by Andersen [AKU+18].

The foundation of a medical device's SDC interface is its object-oriented and tree-structured MDIB (Chapter 2.7.3), which must be initially derived for the OR table. In addition, a standardized data description and an interface are needed

to fulfill the safety requirements in a reconfigurable device network [KSA+18]. Thus, the services provided by an OR table need to be generic and apply to all possible variants and configurations of an OR table (Requirement 58). Since the core functionalities of an OR table are mostly covered with movements for positioning the patient (Appendix A.10.7, functions 1, 2 & 3) and providing the current positions and its current configuration (Appendix A.10.7, function 4), the proposal for an SDC interface base is focused on movements and positions (Requirements 13, 30), patient data (Requirements 14, 15) and Human-Machine-Interface (HMI) (Requirement 26) related services (Fig. 3.15).

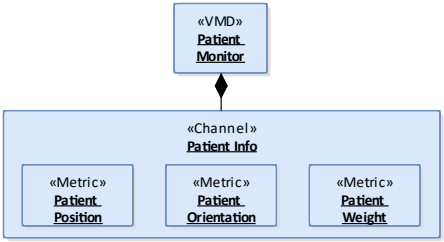


Figure 3.16: Channel structure of the patient monitor VMD

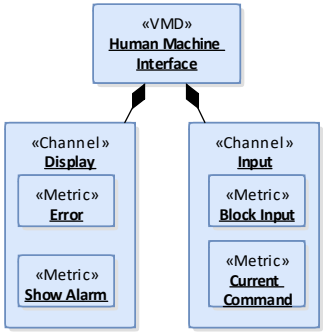


Figure 3.17: Channel structure of the HMI VMD

The SOA paradigm conflicts with technically structured designs and is strongly related to the domain-driven design (Chapter 2.5.1). Thus, providing each module as VMD (Chapter 2.7.3), similar to the proposal by Berger et al. for a two-armed robot [BUK+22], will be less suitable as it does not hide implementation details (*Information Hiding* [Gha20]) and will set specifications for the technical design and implementation of the system. Therefore, the *Joint Tree* is one VMD handling the OR table's whole topology, including the different positions and geometries for the physical components used for collision prevention (Chapter 3.4.1).

Patient-related services (Appendix A.10.7, function 6) are bound to the *Patient Monitor* VMD (Fig. 3.16), which provides information about the patient, such as its current position, orientation (Requirement 15), and weight (Requirement 14).

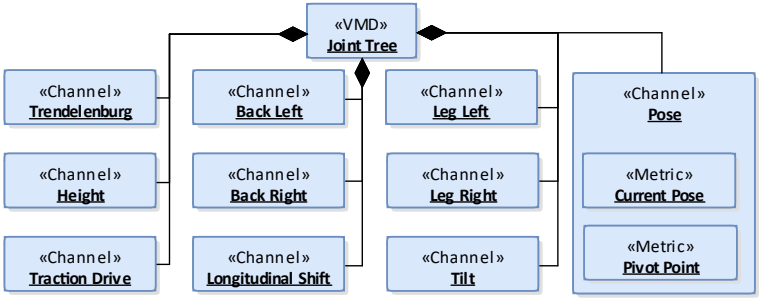


Figure 3.18: SDC channels of the VMD *Joint Tree* (Appendix A.8.2)

For the interactions with the user, the *Human Machine Interface* (Fig. 3.17) is considered. The contained input channel provides metrics (Chapter 2.7.3) to block the remote control input (Requirement 31) or display the currently executed command with the remote control (Requirement 17). Furthermore, for other devices, the display channel can be used for detailed error messages or to show alarms resulting, e.g., from detected anomalies, including those from other devices (Requirement 27).

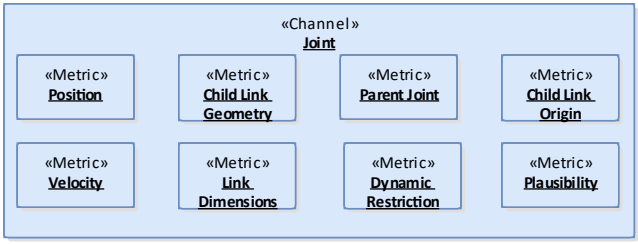


Figure 3.19: Generic joint channel template (Appendix A.8.2)

The *Joint Tree* VMD (Fig. 3.18) holds the channels to provide the configured joint and topology information independent of the used OR table modules and handles movement-related services of an OR table. Furthermore, the structure of the Joint Tree inherently describes the current configuration of the OR table. The purpose of the information provided is also to ensure that the connected devices do not

need to know which module is mounted. Thus, for the service consumer device, it is sufficient to know all joints contained in the Joint Tree VMD to determine the locations of the individual modules, e.g., to prevent collisions. Therefore, the possible joints of an OR table need to fulfill a standardized naming convention (Table A.34, SDC coded values [ISO19b]) that has a defined movement axis (Fig. A.33).

Each joint is represented as a channel in the Joint Tree based on a generic joint channel template (Fig. 3.19) to provide the services of specific joints. Each joint is designed to have only one DoF, so a one-dimensional position (Requirement 13) and velocity metric (Requirement 12) is sufficient. Because the interfaces for each joint should be homologized, joints with multiple DoF are split.

3.4.2 Reconfigurable Anomaly Detection through Ontology-Based Service Composition

The SOMDA approach asserts that it is "necessary to have a standardized way of describing the provided and exchanged data" (Chapter 2.7.3). Otherwise, guarantees for the functionality of a service are not given. For example, if one module considers "tilt" as the rotation of the patient around its y-axis, and another one considers it as rotation around the x-axis, it can lead to serious patient harm (Chapter 2.1.3). In addition, during a session on medical device interoperability [NH20], members of the FDA and other organizations emphasized the need for *platform-based systems*, *standardized data nomenclature*, and support for *plug-and-play capabilities*. They concluded that an *open, extensible architecture* is essential for achieving medical device interoperability. As a contribution to this, a continuous service-composition approach for medical devices, based on the SDC standard's family, is proposed using an OR table [PSS23]. Thus, an "*Ontology-Based Service Composition for Interoperable and Modular Medical Devices*" can be derived for OR tables using the proposed SDC interface (Chapter 3.4.1) [PSS23]. Ontology in this course is understood as follows [van03]:

Definition 19 - Ontology: "A set of well-defined concepts describing a specific domain. The concepts are defined using a subclass hierarchy, by assigning and defining properties and by defining relationships between the concepts, etc."

The proposed architecture builds upon the SODA/SOMDA concept that underlies SDC (Chapter 2.7.3) and Oliveira et al.'s taxonomy layer architecture (Appendix A.9.3), with modules being modeled as services. Similar to SOMDA, the safety of interaction between services is ensured by a semantic, realized with an ontology. Otherwise, the anomaly detection (Chapter 3.2.4) cannot adapt reliably to a changed OR table configuration. Moreover, a generic approach to service composition is introduced and demonstrated in a layered representation in Fig. 3.20 and applied to an OR table (Overview in Fig. A.32). Directed graphs, as proposed by Kampmann et al. [KAK+19], are used to model the service composition.

The composition concept starts with the services proposed in the SDC interface (Chapter 3.4.1) representing the API. Thus, the capabilities of an OR table are based on individual modules *modeled as services*. Composed services are created from lower-level services to obtain a continuous design. Furthermore, the services (metrics) provided by the SDC (Chapter 2.7.3) interface are decomposed into services provided by different software items (Chapter 2.1.2) and assignable to the application layer (Appendix A.9.3). For the interoperability functions implemented here with SDC, a new layer (Fig. A.39) called API is introduced, which is intended for SoS applications and abstracts the medical device as MCPS (Chapter 2.7.4). This layer concerning a SoS context is a gap in the concept of Oliveira [OON13].

The system's functionality, provided to other devices through *API* services, is divided into *Application* services that represent system functions (Chapter 2.5). These functions are further divided into *Agent* services, which rely on combined modules (Chapter 3.2) to realize the features of a function. Agent services are also responsible for coordinating services from the less abstract layers of *Task/Knowledge* services, which consist of *Driver* or other *Task/Knowledge* services (Fig. 3.20). The Driver services are hardware-related ("hardware abstraction

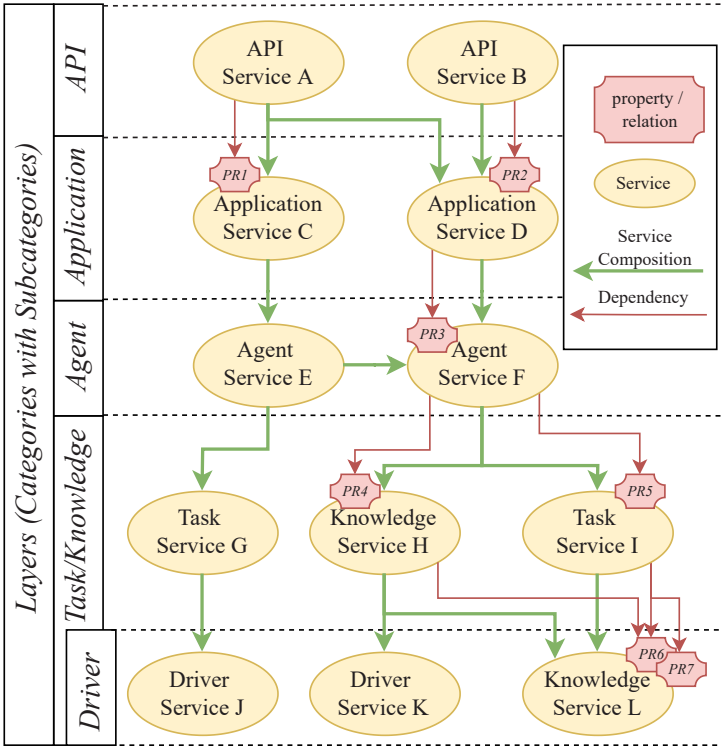


Figure 3.20: Ontology-based service composition based on [PSS23] inspired by [Bue15]

layer") and thus represent the "border" of the *Software Architecture* to the *Network Architecture* (Chapter 2.5). Furthermore, they are responsible for facilitating basic activities and processing information required to execute a task. While fluid boundaries exist between the network and software architecture, the service description is considered a part of the software architecture (Chapter 2.5.3).

In addition to the layering of services, the classification of Oliveira et al. proposes subgroups (Appendix A.9.3) that help to specify a service. A joint movement, for example, can be categorized as an *Object Manipulation* Task as it manipulates the

patient's position. Yet, Oliveira does not propose to specify the services by properties and relations as it is pointed out as possible advancements from taxonomy to ontology in his dissertation [Bue15]. Hence, properties and relationships aid in further specifying the services, allowing the declaration of their association with module types or joint types, for instance. Properties can assist in determining if a service, such as the range of motion of a joint, meets the requirements of a consumer service.

3.4.3 Distributed Anomaly Detection Architecture

Software development for embedded systems in cars or medical devices has historically differed from IT software in that devices have been developed with customized software and software architectures that run bare metal (without any Operating System (OS)) on customized hardware (Chapter 2.5.2). Meanwhile, the IT domain could rely on off-the-shelf components, middleware, and libraries, accelerating development in this domain. Due to Moore's Law[3], embedded systems have experienced significant performance improvements in the past decades. This is also evident in the programming languages used: C was initially considered too resource-intensive for embedded systems, unlike assembly language, and developers had to have confidence in the compilers. Since C++ generates more overhead than C, this triggered another debate about the overhead of programming languages and compilers. At present, C++ is becoming more and more relevant in embedded systems because even in AUTOSAR Adaptive, C++ is considered a standard programming language (Appendix A.9.1). Since object-oriented programming was introduced to improve the reusability and flexibility of software, e.g., through concepts such as dynamic dispatch or late binding, it is a consequent step to improve flexibility by decoupling hardware and software as well as breaking the static binding of ECU communication to signals (Chapter 2.5.3). In addition, the communication overhead, which results from the dynamic binding

[3] The number of transistors on Integrated Circuits (ICs) approximately doubles every two years, contrary to the initial statement of it happening annually [Moo65].

of services, is justifiable with sufficient resources if, thereby, the development and maintenance of systems and programs are facilitated or made possible. This is similar to concepts of object-oriented programming or more abstract programming languages compared to, for example, assembler and C.

This does not indicate that signal-based communication is obsolete and no longer used, but it will be found in less flexible hardware-related areas such as hardware drivers. Furthermore, this development has advantages and is even imperative, as consumer electronics such as smartphones have a higher production volume than, e.g., cars. Consequently, car manufacturers (and other industries) have to adapt to this change to obtain sufficient hardware components. In addition, the shortage of hardware components in 2021 and 2022 [Bar21] has shown manufacturers across all domains the necessity to be flexible with hardware platforms, as dependence on a particular ECU can threaten a company's existence. A flexible software architecture was already promised with hardware abstraction in signal-based architectures like AUTOSAR classic. Due to static routing and dependency on the network topology, hardware dependency is forced in the network architecture (Chapter 2.5). With an SOA, this dependency is targeted to be lifted so that only the driver layer (Chapter 3.4.2) has a hardware dependency already solved by best practices in software engineering, such as a Hardware Abstraction Layer (HAL). Due to the flexible binding of the services, hardware-independent software on the task/knowledge layer (Chapter 3.4.2) and above theoretically can be used for any ECU or server. Nonetheless, this introduces challenges in response guarantees, which are already handled in available SOA middlewares (Appendix A.9.4) and part of the so-called Quality of Service (QoS). Lastly, both communication paradigms have their advantages and disadvantages (Chapter 2.5.1), so there is always a trade-off between the quality characteristics of an architecture (Table A.19) [SSG+22].

The software architecture of the anomaly detection has a significant influence on the performance. In case resource-intensive algorithms are used, they need to be outsourced on servers that are more performant than an embedded system, which must ensure availability while being battery-powered (Chapter 2.2.4). Furthermore, making these resources available in each OR table increases costs, so

other industries, such as the automotive domain, are outsourcing these to cloud systems [WWRS20]. This requires a flexible architecture provided by SOA rather than signal-based communication. Furthermore, since the SOA paradigm stands in contrast to conventional, more technically structured systems such as signal-based architectures, interfaces that integrate (legacy) systems into an SOA are also technically structured and thus violate the SOA core principles [And13]. While signal-based software is structured in physical parts of a system ("move column motor x"), service-oriented software is structured as functions of a domain ("tilt the patient"). With SDC being service-oriented (Chapters 2.7 & 3.4.1), the devices must adapt their architectures internally.

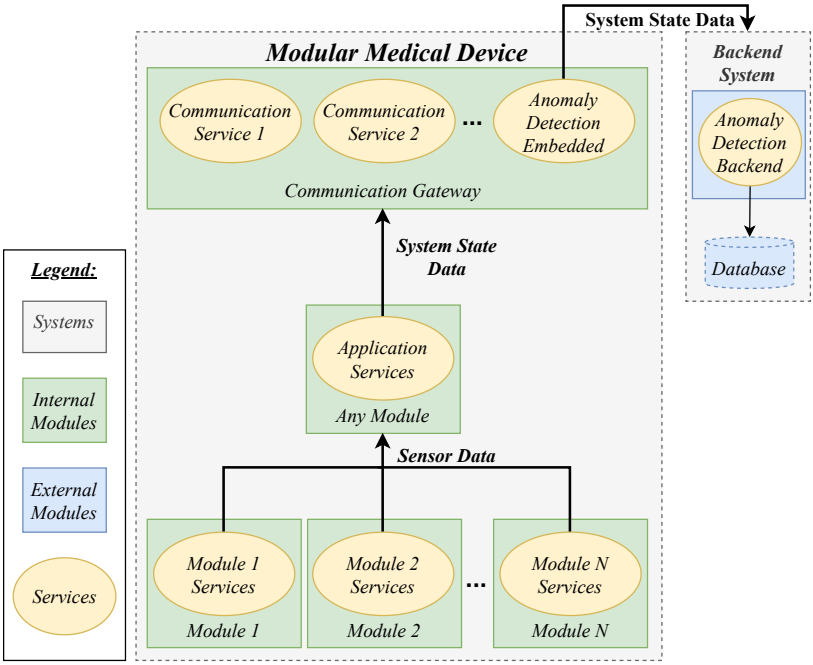


Figure 3.21: Data flow diagram for the *Distributed Anomaly Detection* (external systems: blue, internal modules: green, Fig. 3.1) on agent, application and API service layer (Fig. 3.20)

Based on the *functional* safety and security concept (Chapter 3.2.4), the distributed anomaly detection (Fig. 3.21) is derived as a *technical* safety and security concept (Fig. 2.4), enabling the zero trust core principle 3 *Inspect and log all traffic* (Chapter 2.2.3). The distributed anomaly detection is divided into an embedded and a backend part to enable computing intensive models such as LSTM networks (Chapter 2.3.2). Thus, the *Anomaly Detection Embedded* in the communication gateway provides internal and external data. Learning checks (Chapter 2.3.3) running in the backend system might also leverage other data, which are challenging to include in a mathematical model based on expert knowledge.

The *Anomaly Detection Embedded* service & other communication services of the communication gateway upload the collected data during runtime to the backend system. This data can then be used to train a data-based model after the collection or train it online during collection (Appendix A.5). Furthermore, this requires database handling in the backend system, but enables more flexibility and even the possibility to adapt the learning checks. This helps to counteract variations in series production but requires additional services to enable interaction with these data on the backend system provided to the manufacturer. Another approach is to train the data-based models offline with previously collected data. The latter approach offers a reduced attack surface since the uploaded data cannot be used to manipulate the training of the models used for anomaly detection.

Lastly, the adaptability of the anomaly detection is improved by the flexibility inherent in SOA as opposed to a signal-based architecture. This applies to the dynamic adjustment of anomaly detection algorithms and their reliance on the real-time modular configuration of the OR table. Given its role as a security feature, the imperative for flexible runtime adaptation is further underscored, necessitating improved updatability. This ensures that the services can be updated independently to close security gaps more quickly and reduce the consequences of zero-day attacks (Chapter 2.2.1). Current monolithic architectures enforced by signal-based communication cannot selectively update isolated features as the change requires the software release of the whole application, especially when running bare metal. This increases the development, analysis, and test effort, reducing the time to deployment of security patches.

3.4.4 Safe and Secure Integration of Legacy Modules

As legacy OR table modules in hospitals have been collected for at least two decades (Chapter 3.2.3), their integration must be considered (Chapter 2.1.2, Requirement 49). Furthermore, integrating legacy modules leads to systems with mixed communication paradigms, combining signal-based and service-oriented communication. Therefore, signal-based modules must be integrated into the new service-oriented architecture to include these in the distributed anomaly detection (Chapter 3.4.3). Thus, signals are converted into services to enable dynamic adaption and profit from the safe and secure ontology-based service composition (Chapter 3.4.2). Otherwise, they must be examined individually, which introduces additional challenges, rendering the process more susceptible to errors. Moreover, legacy modules pose a heightened security threat attributable to outdated technology or code, compelling their inclusion in safety and security considerations.

To keep the existing certification of legacy devices (Chapter 2.7.4) in an SDC network, Pfeiffer et al. [PDDL15] propose wrapping their interface to integrate these into an SDC network. A similar approach in the automotive domain uses gateways for signal-based and service-oriented communication of different ECUs [PVR+22]. Additionally, this solution is applicable for legacy OR table modules, but the intended use of a legacy module must be untouched to maintain the certification (Requirement 50). In this way, it is sufficient to certify the *integrated connector* without re-certifying the legacy modules [PDDL15].

Fig. 3.22 shows a mixed architecture integrating signal-based and service-oriented modules, with two ECUs - master and supervisor - used within a module for single fault safety features [PVR+22]. The addition of a *Signal-Based/Service-Oriented Gateway (SB/SO-GW)* is based on the AUTOSAR solution, where the information on how to map the individual signals to services and vice versa is taken from the *Communication Matrix* and a *Signal-to-Service Mapping* [Tis18] (Fig. A.36). This gateway is considered to bridge between Classic and Adaptive AUTOSAR so that legacy ECUs and the corresponding software can be reused.

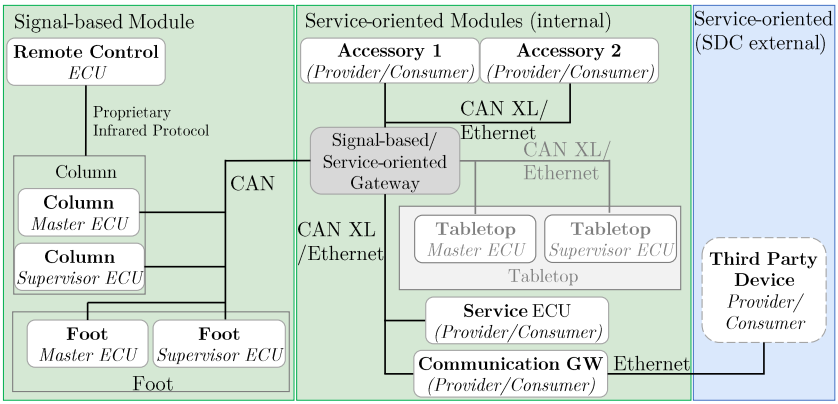


Figure 3.22: Mixed architecture OR table with exchangeable tabletop (grayed out) [PVR+22] (external: blue, internal: green, Fig. 3.1)

It is possible to wrap the interface of each legacy module into a dedicated integrated connector instead of having a central gateway. Although this would reduce the software effort, as only the signals and services relevant to the module need to be taken into account, it would also mean that each module would require its dedicated hardware as a gateway, making it less flexible and increasing costs. Furthermore, this approach conflicts with the intention of service orientation to decrease hardware dependency (Chapter 2.5.3) and requires each integrated connector to be certified instead of one central gateway. Thus, a central gateway is preferred. This preference is motivated by a software perspective to decouple from the hardware so that the mechanical and hardware effort is neglected.

An SB/SO-GW ensures communication between both paradigms and hosts various legacy module services, such as controlling the height movement of a column module that can be used from service-oriented modules. Conversely, signal-based modules can consume services by translating all signals in service invocations or providing data from service-oriented modules as cyclic signals in the signal-based part of the system. Signal-based communication with CAN indirectly uses the *publish-subscribe* pattern as all messages are broadcasted on the bus, and only those ECUs that are interested in a message read it (Appendix A.3.1). Thus,

the translation into *publish-subscribe* services is straightforward compared to other communication patterns (Appendix A.9). Furthermore, there are custom & proprietary signal-based communication patterns that need specific adaptions, e.g., during the system's startup. Registration and configuration data must be distributed from the modules at startup and then provided as a service by the gateway, which is error-prone due to the added delay. Since these registration processes are implementation-specific, they are not further considered here.

4 Anomaly Detection Design for OR Table Positions and Movements

The anomaly detection function (Chapter 3.2.4) is related to the supervision and monitoring of system states and communication data (Fig. 4.1). The recognition of unknown attacks and other security requirements can be fulfilled (Table 4.1) with anomaly detection (Chapter 2.3). In addition, requirements regarding safety can be fulfilled with this function (Requirement 41) since the positions of the OR table (Requirement 29), or the patient (Requirement 28) can be implausible in case of system failures such as defect sensors. Thus, a plausibility check for the positions helps to enable single fault safety (Requirement 33) and contributes to the precision of the OR table joint positions (Requirement 11).

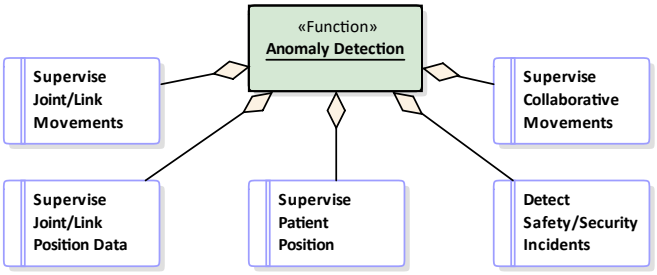


Figure 4.1: Anomaly detection function decomposition into features (Chapter 2.5)

An IDS, which can detect unknown anomalies, might supervise the safeguarding of self-adapting and machine learning-based (vehicle) functions [WKSZ18]. Furthermore, the FDA advises to protect "Code, Data, and Execution Integrity", e.g., by HIDS, as well as "Event Detection and Logging" among others [FDA22]. Lastly, communication behavior and other side-channel parameters are no longer sufficient for CPS as security measures due to uncertainties in physical behavior (Chapter 2.2.1).

Table 4.1: Overview of functional and non-functional requirements fulfilled with anomaly detection in positions (Chapters 3.2.1, 3.2.2 & 4.1.5)

Req. ID	Requirement Title
Req. 11	High Precision Positioning
Req. 28	Check Plausibility of Patient Positioning
Req. 29	Check Plausibility of System Positions
Req. 33	Single Fault Safety
Req. 35	Report any Safety Incidents
Req. 36	Detect Dangerous System Situations
Req. 37	Monitor System State Plausibility
Req. 41	Analyze System During Life Cycle
Req. 42	Payload Inspection
Req. 43	Unknown Attacks
Req. 44	Report any Security Incidents
Req. 46	Check Plausibility of Communicated Positions

Regarding the OR table, the validity of position information is essential (Chapter 4.1), and the corresponding changes, including the cause of position changes, offer input for plausibility checks. Thus, the *Position* metrics of the Joint Channels that are part of the Joint Tree VMD (Fig. 3.18) are checked for plausibility with the *Patient Position* metric of the *Patient Position Channel* of the *Patient Monitor* (Fig. 3.16).

4.1 TARA for Positions and Movements of OR Tables

The *intended use* of OR tables is "supporting and positioning a patient during surgical procedures" (Chapter 2.6), so analyzing the corresponding new threats and risks in a cyber-physical OR is the first priority. To achieve this, the adapted TARA (Chapter 3.1) for medical devices can be executed based on a rough architecture (Chapter 3.4.4, Fig. 3.22).

4.1.1 Item Definition and Asset Identification

For the item definition, a DFD (Fig. 4.2) is applied, whereas a per-element STRIDE (Chapter 2.2.1) is used as it is more effective than a per-function STRIDE [TS18]. Thus, all elements, such as the column, tabletop, or service ECU in the DFD, are assets (Figs. 3.1 & 3.22). The DFD only shows interactions concerning the movement and position of different joints and their corresponding links (including the patient position). Since, in this case, column, tabletop, and foot ECUs share the same interactions, but with different joints, they are different assets, but the data flow is identical.

The column, tabletop, and foot ECUs handle the joints by primarily storing and controlling their positions. Via the service ECU, a service technician can set the reference positions of a joint during maintenance. Furthermore, the clinical staff can move the OR table joints via remote control, which controls the central OR table application. This is also possible for other medical devices in the SDC network via the OR table API provided by the communication gateway (Com. GW), which realizes the SDC interface of the OR table and is considered a separate ECU. The joint and link positions influenced by these actions are then communicated to SDC participants consuming these services. This data is additionally communicated to a backend system for monitoring and logging.

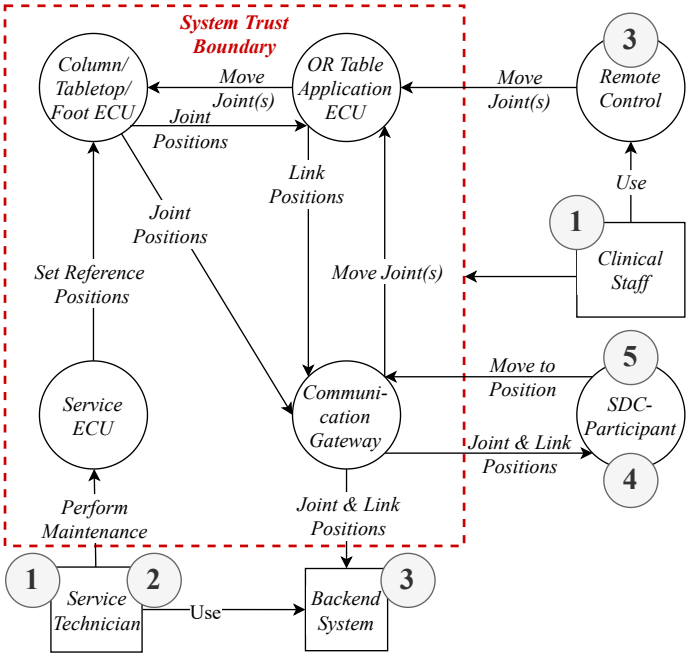


Figure 4.2: Data flow diagram for joint and link positions with threat sources (Chapter 3.1.2, Fig. 3.3)

4.1.2 Threat and Damage Scenario

The next step in the HEAVENS 2.0 workflow is the analysis of attack paths, which is realized as an attack tree (Fig. 4.3). A high risk results from communicating incorrect positions to other medical devices via the communication gateway (Chapter 4.1.4). For example, during robot-assisted surgery (Appendix A.1.3) or imaging procedures with angiography systems in a HOR (Appendix A.1.2), an incorrect position of the OR table can lead to life-threatening situations[1], e.g., due to collisions or tipping of the OR table (Chapter 2.6.4). Thus, this is considered as the *damage scenario* for further examination.

[1] attack target type: directly *harm* a patient's health, Chapter 2.2.1

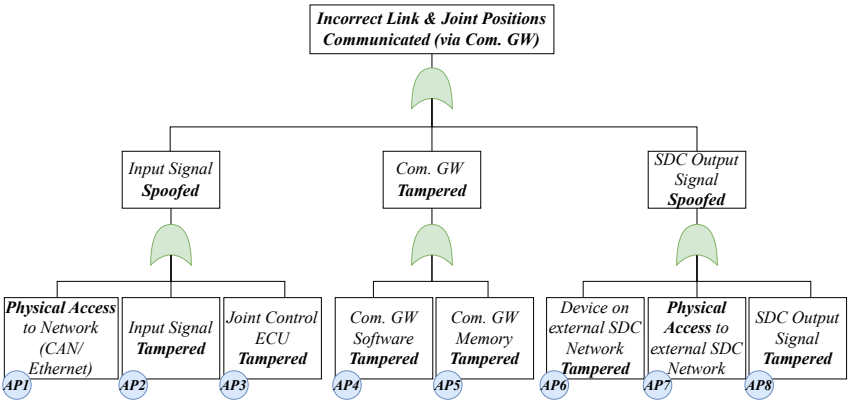


Figure 4.3: Attack tree for incorrect communicated positions of links and joints

The input signal of the communication gateway can be spoofed in case an attacker has gained physical access to the system bus, such as Ethernet (Appendix A.3.2) or CAN (Appendix A.3.1), and sends valid messages, e.g., with a replay attack (AP1). Other possibilities are to directly tamper the input messages, e.g., via a man-in-the-middle attack (AP2), to tamper the software of the ECUs directly determining the positions via sensors by either setting the reference positions via a compromised service ECU (Elevation of privileges) or by installing a manipulated software update (AP3).

The communication gateway itself can be manipulated by directly tampering with the memory of the corresponding position data (AP4) or updating compromised software (AP5). The output signal in the SDC network can be spoofed, e.g., by a compromised device sending valid messages and impersonating the valid OR table (AP6). Furthermore, in case the attacker has physical access to the SDC network, these messages can also be sent directly (AP7), or in case an attacker tampers the SDC output signals (AP8), which might be possible if the attacker already has access to the hospital IT.

4.1.3 Attack Path Analysis, Attack Feasibility Rating and Impact Rating

The feasibility ratings are summed up in Table 4.2. The four sub-parameters *expertise, knowledge of the item, window of opportunity, and equipment* (Table A.15) proposed by Lautenbach et al. [LAO21] are used for the calculation of the attack feasibility rating, whereby none are considered as especially important in comparison to the others so that no weighting of the parameters is performed.

Table 4.2: Attack feasibility rating for attack paths of the damage scenario *Incorrect Link & Joint Position Communicated*

Attack Path ID	Expertise	Knowledge of Item	Window of Opportunity	Equipment	Attack Feasibility Rating
AP1	2	1	0	2	Low (42%)
AP2	2	2	2	3	Medium (75%)
AP3	1	2	2	3	Medium (67%)
AP4	1	2	3	2	Medium (67%)
AP5	2	2	3	3	High (83%)
AP6	2	2	3	3	High (83%)
AP7	1	2	3	2	Medium (67%)
AP8	2	2	3	2	Medium (75%)

The *expertise* indicates how experienced the attacker is, ranging from a layman (value 3) to several experts (value 0) required to perform this attack. Furthermore, the *knowledge of the item* of concern reaches from publicly available (value 3) to critical (value 0). The *window of opportunity* ranges from an unlimited (value 3) to a small window (value 0) in which the device can be attacked (Table A.16). Lastly, the *equipment* column involves standard (value 3) to multiple bespoke equipment (value 0). For example, in *AP1*, proficiency is needed to access the physical network. At the same time, knowledge of the items under attack is sensitive because manufacturers keep their intellectual property under seal. In addition, the window of opportunity is negligible since the attacker must be undiscovered

within the hospital or even in the OR to physically open the OR table and connect to the system bus. Moreover, only some specialized equipment is required to open the system and connect to the internal network, leading to an overall attack feasibility rating of 42%, rated as low (Table A.12).

4.1.4 Risk Determination and Treatment Decision

Collisions with other medical devices may occur, which could injure or even kill the patient (Chapter 4.1.2). Thus, an *Impact Rating* of *severe* severity (Table A.13) is inevitable if other medical devices use the position. In addition, the attack feasibility rating is also at least medium in all scenarios (Table 4.3), leading to the highest risk value of *5* (Table A.14). The decisive factor here is the highest classification of impacts resulting from uses with a severe impact rating.

Table 4.3: Risk values for examined threat scenarios (Fig. 4.3)

Threat Scenario	Attack Feasibility Rating	Impact Rating	Risk Value
Input Signal Spoofed	Medium	Severe	5
Com. GW Tampered	High	Severe	5
SDC Output Signal Spoofed	High	Severe	5

The risk must be reduced since its value is rated at *5* in all three threat scenarios (Table 4.3). From today's technical point of view, the clinical staff still needs to decide whether more extensive measures are needed depending on the current surgery context since the technical determination of situations during surgery is still part of research (Chapter 2.3.3) and technical measures can lead to more significant harm. In addition, the availability of the OR table resulting from the intended use must still be ensured at all times (Chapter 2.6). The cybersecurity goal is, therefore, *to detect implausible positions and report the anomaly to the clinical staff* (Requirement 29).

4.1.5 Cybersecurity Goals for the Anomaly Detection

Application of IAMs (Appendix A.7.4) and encryption can lower the probability of manipulation by physical access to the network or by tampering with the messages. Furthermore, firewalls (Appendix A.7.3), combined with specification-based/signature-based IDS (Chapter 2.2.4) in the connectivity ECU will make it less accessible from the outside. Thus, attacks via a compromised clinical IT infrastructure or other compromised medical devices can be prevented. Still, there are threat scenarios and attack paths that have not been discovered here, and protective measures might fail, leading to zero-day attacks (Requirement 43, Chapter 2.2.1). Successfully spoofed input signals cannot be identified if the payload is not checked for plausibility (Requirement 42). Despite standard preventive security measures (Appendix A.7), such as network separation, Firewalls, or Virtual Private Networks (VPNs), attackers can still bypass these. Therefore, anomaly-based IDSs (Chapter 2.2.4) are necessary to cover the remaining attack surface, and particularly in CPSs, they can use knowledge of physical processes to detect anomalies [WWH20].

The proposed measures can be located on *layer 4* (Access limitation for system internal networks) and *layer 5* (External System Connections) but are not restricted to a layered security approach (Chapter 2.2.2) and thus are applicable in a zero trust security architecture as well (Chapter 2.2.3). Hence, measures such as IAMs and Firewalls contribute to the second step, "*Protect*", of the NIST cybersecurity framework for critical infrastructure [Nat18] regarding medical devices. In addition, anomaly detection contributes to the third step, "*Detect*".

Although measures such as specification-based anomaly detection can be used to reduce the attack surface, the percentage of attack surface that can be reduced will shrink. Since there is increased flexibility in communication due to the modularity and reconfigurability of OR tables (see also [Sch22]) and future reconfigurable OR networks, the detection of zero-day attacks resulting from unknown vulnerabilities will play a more critical role (Requirement 43). When relying on service-oriented communications and allowing internal and external

reconfigurability (Requirement 25), static controls, e.g., through manually set filtering rules for specific messages, IPs are more challenging than for traditional signal-based communications. Therefore, it must be ensured that the positions provided by the OR table via the SDC Interface are plausible (Requirements 37 & 46). Furthermore, security measures are needed, independent of the applied protocols, middlewares, and technologies (Requirement 56), as legacy and new modules will differ in these (Chapter 3.2.3).

A significant threat arises from external communication, which is why the output signal must be monitored. The system should additionally log the communicated positions (Chapter 2.2.3) and check their plausibility since the possibility of an attack is increased here. Moreover, the internal system data must be checked for plausibility as all threat scenarios have a high impact rating, and the communicated position is based on the internally determined positions (Requirements 28 & 29).

4.2 Static Checks for Detection of Collisions

As Weber examined, physical signals can be checked for anomalies (Chapter 2.3.3). Checks derived from the specifications of an OR table are, for example, the maximum and minimum achievable positions of a joint or the maximum speed of a joint. Also, sudden changes in the position or speed must not occur during normal operation. The patient's position can be monitored with static checks by considering the system context. OR tables that can determine the patient's weight (Chapter 2.6.4) can monitor sudden changes, indicating that the patient's position might have changed or the patient's weight has been manipulated. If this value is communicated via the SDC interface, other devices relying on this value must be informed about an implausible change. Yet, there are scenarios in which it is plausible that the weight suddenly changes. One of these is Cardiopulmonary Resuscitation (CPR), during which the weight on the OR table cyclically changes due to the force of the massage pulses. These cases should be ignored, but the pattern could be recognized only by dynamic (Chapter 3.3.1) or learning checks (Chapter 3.3.2).

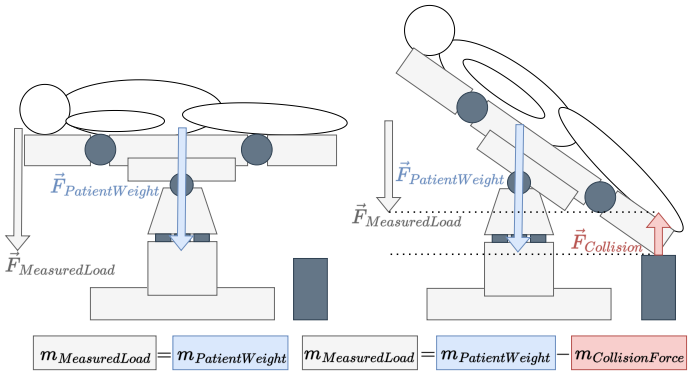


Figure 4.4: Collision detection using the measured force/weight change on movement [SPRD22]

During a movement of the OR table, the patient's weight should not change. Thus, in case an external force is applied in addition to the gravitational force applied by the patient, the overall measured weight does change (Fig. 4.4). If the measured difference exceeds a permissible value, it can be assumed that a collision has happened and needs to be prevented [SPRD22]. As this function is integrated into the Corin OR table by Getinge [Get23c], the implementation is proprietary. Furthermore, the detected event can be combined with other checks, such as learning checks, to handle the above-mentioned expected changes in weight.

4.3 Dynamic Checks for Position Surveillance

The kinematics and physical relations (Chapter 2.4) of an OR table can be used to check the plausibility of the measured (Patent [PD22]) and communicated patient CoG as a dynamic check (Chapter 3.3). For a plausibility check of signal curves, it is insufficient to check for maximum and minimum signal values, so the gradient is needed. Thus, Weber [Web19] applies data-based models to observe these non-specified properties. Nonetheless, the kinematic and physical models are not considered, and in this case, it is also possible to use a dynamic check based on a KF (Chapter 2.3.1). With the kinematics of the OR table, it is possible to

estimate the trajectory of the patient by a KF (Fig. A.52). The difference between the estimation and the measurement is then used to monitor the plausibility of the position (Chapter 3.3.1).

4.3.1 Unscented Kalman Filter (UKF) for Whole Body Movements

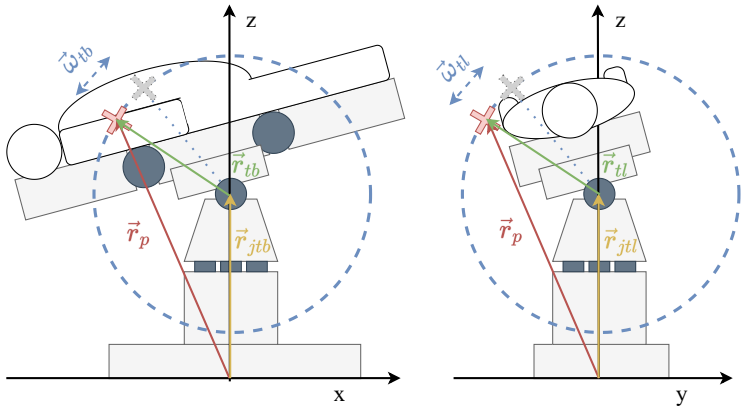


Figure 4.5: Overview of whole body movements Trendelenburg and tilt used for the dynamic check

The dynamics for whole patient body movements are independent of the patient's positioning if one neglects flexible elements (Appendix A.6.2) so that rigid body dynamics can be assumed (Figs. 4.5 & A.50). The distance between the tilt joint origin $\vec{r}_{jtl}$ and the distance between the Trend joint origin $\vec{r}_{jtb}$ to the measured CoG represented by the patient's position $\vec{r}_p$ define the radius of the circular trajectory with the angular speeds $\vec{\omega}_{tb}$ and $\vec{\omega}_{tl}$ (Chapter 2.4). In addition, the longitudinal shift with joint origin $\vec{r}_{jls}$ moves the patient with velocity $\vec{v}_{ls}$ along a parallel axis of the longitudinal shift axis (Fig. A.50). The overall velocity $\vec{v}_p$ of

the patient can, therefore, be determined with the following equation (Table 4.4), whereby the joint velocities $\vec{\tilde{\omega}}_{joint}$ are derived from local joint coordinates $\vec{\omega}_{joint}$ (Appendix A.12.1):

$$\vec{v}_p = \vec{\tilde{v}}_{ls} + \vec{\tilde{\omega}}_{tb} \times \vec{r}_{tb} + \vec{\tilde{\omega}}_{tl} \times \vec{r}_{tl} \tag{4.1}$$

Table 4.4: Overview of variables for whole body movements

Variable	Description
$\vec{r}_p$	Position vector of the patient's CoG in world coordinates[2]
$\vec{v}_p$	Velocity vector of the patient's CoG
$\vec{\tilde{v}}_{ls}$	Velocity vector of the prismatic longshift joint in world coordinates
$\vec{\tilde{\omega}}_{tb}$	Angular velocity vector of the rotational Trendelenburg joint in world coordinates
$\vec{\tilde{\omega}}_{tl}$	Angular velocity vector of the rotational tilt joint in world coordinates
$\vec{r}_{ls}$	Distance vector between the longitudinal shift joint origin and the CoG of the patient ($\vec{r}_p - \vec{r}_{jls}$)
$\vec{r}_{tl}$	Distance vector between the tilt joint origin and the CoG of the patient ($\vec{r}_p - \vec{r}_{jtl}$)
$\vec{r}_{tb}$	Distance vector between the Trendelenburg (Trend) joint origin and the CoG of the patient ($\vec{r}_p - \vec{r}_{jtb}$)

This results in the following velocity model by taking the current Trend angle φ_{tb} into account (Derivation in Appendix A.12.1):

$$\vec{v}_p = \begin{bmatrix} v_{ls} \cdot \cos(\varphi_{tb}) + \omega_{tb} \cdot r_{z,tb} + \omega_{tl} \cdot \sin(\varphi_{tb}) \cdot r_{y,tl} \\ -\omega_{tl} (\sin(\varphi_{tb}) \cdot r_{x,tl} + \cos(\varphi_{tb}) \cdot r_{z,tl}) \\ -v_{ls} \cdot \sin(\varphi_{tb}) - \omega_{tb} \cdot r_{x,tb} + \omega_{tl} \cos(\varphi_{tb}) \cdot r_{y_{tl}} \end{bmatrix} \tag{4.2}$$

[2] "World coordinates" are considered here as a uniform coordinate system with a fixed reference point in OR for each device

Simultaneous joint movements are excluded here so that the φ_{trend} is considered constant during a tilt movement, and thus, it does not need to be included in the state vector $\mathbf{x}$:

$$\mathbf{x} = \begin{bmatrix} r_{px} & r_{py} & r_{pz} & r_{x,jtb} & r_{y,jtb} & r_{z,jtb} & r_{x,jtl} & r_{y,jtl} & r_{y,jtl} & v_{ls} & \omega_{tl} & \omega_{tb} \end{bmatrix}^T \tag{4.3}$$

In the examined case, where the joint origins $\vec{r}_{jtb}$ and $\vec{r}_{jtl}$ are considered constant due to the kinematic chain, the joint origins of Trend and tilt are approximately the same. Therefore, the state vector $\mathbf{x}$ can be further reduced:

$$\mathbf{x} = \begin{bmatrix} r_{px} & r_{py} & r_{pz} & v_{ls} & \omega_{tl} & \omega_{tb} \end{bmatrix}^T \tag{4.4}$$

In both cases, the prediction is based on the following differential equation describing the dynamics of the patient position $\vec{r}_p$:

$$\begin{bmatrix} \dot{r}_{px} \\ \dot{r}_{py} \\ \dot{r}_{pz} \\ \dot{v}_{ls} \\ \dot{\omega}_{tl} \\ \dot{\omega}_{tb} \end{bmatrix} = \begin{bmatrix} v_{ls}\cos(\varphi_{tb}) + \omega_{tb}(r_{pz} - r_{z,tb}) + \omega_{tl}\sin(\varphi_{tb})(r_{py} - r_{y,tl}) \\ -\omega_{tl}(\sin(\varphi_{tb})(r_{px} - r_{x,tl}) + \cos(\varphi_{tb})(r_{pz} - r_{z,tl})) \\ -v_{ls}\cdot\sin(\varphi_{tb}) - \omega_{tb}(r_{px} - r_{x,tb}) + \omega_{tl}\cos(\varphi_{tb})(r_{py} - r_{y_{tl}}) \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.5}$$

Thus, each of the states can be measured so that the measurement function $h(\mathbf{x})$ results in:

$$h(\mathbf{x}) = \mathbf{x} \tag{4.6}$$

Another approach is adapting $f(\mathbf{x})$ and projecting the current states to the measured values. Since the cross-product is not invertible, $\vec{\omega}_{tb}$ and $\vec{\omega}_{tl}$ cannot be uniquely determined if only $\vec{r}_p$, $\vec{r}_{tl}$ and $\vec{r}_{tb}$ are known. Thus, the measured velocities for $\vec{\omega}_{tl}$ and $\vec{\omega}_{tb}$ need to be added to the state vector $\mathbf{x}$, and the system matrix $\mathbf{A}$ needs to contain the whole system equations.

4.3.2 Extended Kalman Filter (EKF) for Whole Body Movements

Since the non-linear and linear equations from the UKF of the whole body movements (Chapter 4.3.1) also build the EKF basis, they are used here as well. Moreover, they must be linearized at the current state for each filter step (Chapter 2.3.1). Therefore, the Jacobi matrix representing the transition matrix $\mathbf{A}$ of the function $f(\mathbf{x})$ is determined as follows with time T between each sampling point ($sin(x)$ and $cos(x)$ are shortened to c_x and s_x for readability):

$$T \begin{bmatrix} \frac{1}{T} & \omega_{tl} c_{\varphi_{tb}} & \omega_{tb} & c_{\varphi_{tb}} & s_{\varphi_{tb}}(r_{py} - r_{y,tl}) & (r_{pz} - r_{z,tb}) \\ -\omega_{tl} s_{\varphi_{tb}} & \frac{1}{T} & -\omega_{tl} c_{\varphi_{tb}} & 0 & -s_{\varphi_{tb}}(r_{px} - r_{x,tl}) - c_{\varphi_{tb}}(r_{pz} - r_{z,tl}) & 0 \\ -\omega_{tb} & \omega_{tl} c_{\varphi_{tb}} & \frac{1}{T} & -s_{\varphi_{tb}} & c_{\varphi_{tb}}(r_{py} - r_{y,tl}) & -r_{px} + r_{x,tb} \\ 0 & 0 & 0 & \frac{1}{T} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{T} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{T} \end{bmatrix} \tag{4.7}$$

The measurement Matrix $\mathbf{H}$ resulting from $h(\mathbf{x})$ is linear and thus does not need to be linearized at each filter step. Since the full state $\mathbf{x}$ is also measured completely, $\mathbf{H}$ is a six-dimensional identity matrix:

$$\mathbf{H} = \mathbf{I}^{(6x6)} \tag{4.8}$$

4.3.3 UKF for Partial Body Movements

From a robotic point of view, the human body can be modeled as a multi-body system based on a mass model of a standardized patient from IEC60601-2-46 [IEC16a] (Fig. A.7). Therefore, the CoG of all masses combined define patient position (CoG). Furthermore, with a multi-body model of the human body, it is possible to predict partial body movements. In this case, the position of the individual body parts must be considered, whereby the model can become arbitrarily fine-grained. Furthermore, the positioning of the patient is more critical than for whole body movements since the joints can move different body parts with different inertia and CoG (Fig. 4.6, Appendix A.51).

This only holds if the flexible elements (Appendix A.6.2) are neglected so that rigid body dynamics can be assumed (Chapter 2.4). As proof of concept, it is sufficient to sum individual body parts into two parts: the upper and lower body of the patient. For rigid body dynamics with homogeneous mass distribution, the patient's upper body CoG position $\vec{r}_{ub}$ as well as the patient's lower body CoG position $\vec{r}_{lb}$ can be used to calculate the patient's CoG $\vec{r}_p$ (Chapter 2.4):

$$\vec{r}_p = \lambda_{ub}\vec{r}_{ub} + \lambda_{lb}\vec{r}_{lb} \tag{4.9}$$

Table 4.5: Overview of variables for partial body movements

Variable	Description
$\vec{r}_{ub}$	CoG position vector of the patient's upper body in world coordinates
$\vec{r}_{lb}$	CoG position vector of the patient's lower body in world coordinates
$\vec{r}_{jb}$	Position vector of the back joint origin in world coordinates
$\vec{r}_{jl}$	Position vector of the leg joint origin in world coordinates
$\vec{\tilde{\omega}}_{jb}$	Angular velocity vector of the rotational back joint in world coordinates
$\vec{\tilde{\omega}}_{jl}$	Angular velocity vector of the rotational leg joint in world coordinates
$\vec{r}_{jub}$	Distance vector between the back joint origin and the CoG of the patient's upper body ($\vec{r}_{ub} - \vec{r}_{jb}$)
$\vec{r}_{jlb}$	Distance vector between the leg joint origin and the CoG of the patient's lower body ($\vec{r}_{lb} - \vec{r}_{jl}$)

Here, λ_{ub} is the proportion for the moved upper body part with the back joint, and λ_{lb} is the proportion for the moved upper body part with the leg joint. They might not add up to 1 since a possible middle body part with distance $\vec{r}_{mb}$ is neglected in the previous equation and is not moved by the tabletop joints of concern (leg and back joints). Since this part can be considered constant for the examined positions, it does not contribute to the differential equation for $r_{p,x}$ and $r_{p,z}$ and thus does not occur in the system matrix $\mathbf{A}$.

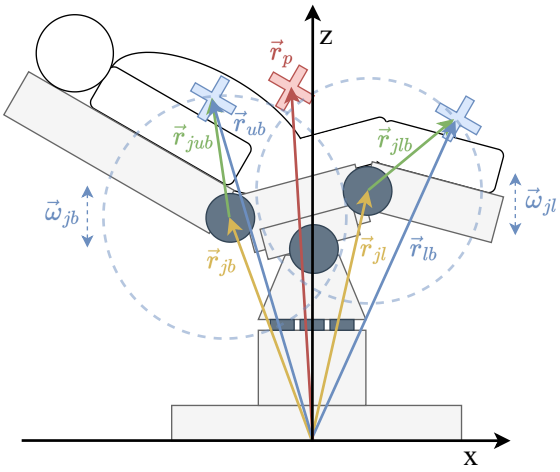


Figure 4.6: Patient in beach chair position (Fig. A.6) with a patient CoG position $\vec{r}_p$ in dependency to upper body CoG position $\vec{r}_{ub}$, lower body CoG position $\vec{r}_{ub}$, back joint distance $\vec{r}_{jb}$ and leg joints distance $\vec{r}_{jl}$ (Table 4.5) including the motion trajectories (blue dashed) of the upper body when moving the back joint and the lower body when moving the leg joints

The patient positioning techniques have a finite number in practice and thus can be separated into different scenarios. In the beach chair position (Fig. 4.6), the upper body represents approximately 63% of the whole body. The lower body part is 14,8% from the lower leg, and the remaining 22,2% (Appendix A.7) of the upper legs are not affected by the tabletop joints but only by whole body movements. In the supine position (Fig. A.51), the upper body represents only the head (7,4%), while the lower body represents 37%, including the lower and upper legs. The remaining body part represents approximately 55,5% that are only moved by whole-body motions. With the previously mentioned adaption, the patient's body is parted in the lower and upper body. Thus, they are considered $\lambda_{ub} = 0.63$ (Fig. 4.6) and $\lambda_{lb} = 0.37$ (Fig. A.51).

$$\dot{\vec{r}}_p = 0.63\vec{v}_{ub} + 0.37\vec{v}_{lb} = 0.63\vec{\omega}_{jb} \times \vec{r}_{jub} + 0.37\vec{\omega}_{jl} \times \vec{r}_{jlb} \tag{4.10}$$

Since only rotation around the y-axis for leg and back joints is possible, the y-axis can be ignored:

$$\begin{bmatrix} \dot{r_{px}} \\ r_{py} \\ r_{pz} \end{bmatrix} = \begin{bmatrix} 0 \\ \lambda_{ub}\omega_{jb} \\ 0 \end{bmatrix} \times \begin{bmatrix} r_{jubx} \\ r_{juby} \\ r_{jubz} \end{bmatrix} + \begin{bmatrix} 0 \\ \lambda_{lb}\omega_{jl} \\ 0 \end{bmatrix} \times \begin{bmatrix} r_{jlbx} \\ r_{jlby} \\ r_{jlbz} \end{bmatrix} \tag{4.11}$$

With the influence of the other joints, the tilt angle needs to be considered (Appendix A.12.2):

$$\begin{bmatrix} \dot{r_{px}} \\ r_{py} \\ r_{pz} \end{bmatrix} = \lambda_{ub}\omega_{jb} \begin{bmatrix} \cos(\varphi_{tl})\cdot r_{jub,z} - \sin(\varphi_{tl})\cdot r_{jub,y} \\ \sin(\varphi_{tl})\cdot r_{jub,x} \\ -\cos(\varphi_{tl})\cdot r_{jub,x} \end{bmatrix}$$

$$+\lambda_{lb}\omega_{jl} \begin{bmatrix} \cos(\varphi_{tl})\cdot r_{jlb,z} - \sin(\varphi_{tl})\cdot r_{jlb,y} \\ \sin(\varphi_{tl})\cdot r_{jlb,x} \\ -\cos(\varphi_{tl})\cdot r_{jlb,x} \end{bmatrix} \tag{4.12}$$

The joint positions are given since they are determinable from the OR table kinematics. Moreover, the joints will also be deformable due to flexibility in elements that are not rigid (Appendix A.6.2). This will be considered an additional modeling error in the process noise matrix. Another adaption is that no whole body motions are executed together with partial movements so that the tilt angle φ_{tl} and the joint origins $\vec{r}_{jb}$ as well as $\vec{r}_{jl}$ do not need to be integrated into the state vector. It is also considered that for further investigation, φ_{tl} equals 0. Thus, $cos(\varphi_{tl}) = 1$ and $sin(\varphi_{tl}) = 0$, as it is not expected that the factors resulting from φ_{tl} have an impact on the results of the performance evaluation (Chapter 6). In addition, this reduces the dimension of the state $\mathbf{x}$ further as $\dot{r}_{py} = 0$, $\dot{r}_{ub,y} = 0$ and $\dot{r}_{lb,y} = 0$, reducing it to a two-dimensional problem:

$$\begin{bmatrix} \dot{r_{px}} \\ r_{pz} \\ r_{ub,x} \\ r_{ub,z} \\ r_{lb,x} \\ r_{lb,z} \\ \omega_{jb} \\ \omega_{jl} \end{bmatrix} = \begin{bmatrix} \lambda_{ub}\omega_{jb}(r_{ub,z} - r_{jb,z}) + \lambda_{lb}\omega_{jl}(r_{lb,z} - r_{jl,z}) \\ -\lambda_{ub}\omega_{jb}(r_{ub,x} - r_{jb,x}) - \lambda_{lb}\omega_{jl}(r_{lb,x} - r_{jl,x}) \\ \omega_{jb}(r_{ub,z} - r_{jb,z}) \\ -\omega_{jb}(r_{ub,x} - r_{jb,x}) \\ \omega_{jl}(r_{lb,z} - r_{jl,z}) \\ -\omega_{jl}(r_{lb,x} - r_{jl,x}) \\ 0 \\ 0 \end{bmatrix} \tag{4.13}$$

Nonetheless, the measurement Matrix $\mathbf{H}$ resulting from $h(\mathbf{x})$, is linear and thus does not need to be recalculated at each filter step. Since only the overall patient's

position and the joint velocities of state $\mathbf{x}$ are measured, $\mathbf{H}$ is determined as follows:

$$\mathbf{H} = \begin{bmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \end{bmatrix} \tag{4.14}$$

An alternative approach that reduces the dimension of the state $\mathbf{x}$ further to 6 is not to include the patient's CoG $\vec{r}_p$ in the model but adapt the measurement function $h(\mathbf{x})$ accordingly:

$$\begin{bmatrix} \dot{r_{ub,x}} \\ r_{ub,z} \\ r_{lb,x} \\ r_{lb,z} \\ \omega_{jb} \\ \omega_{jl} \end{bmatrix} = \begin{bmatrix} \omega_{jb}(r_{ub,z} - r_{jb,z}) \\ -\omega_{jb}(r_{ub,x} - r_{jb,x}) \\ \omega_{jl}(r_{lb,z} - r_{jl,z}) \\ -\omega_{jl}(r_{lb,x} - r_{jl,x}) \\ 0 \\ 0 \end{bmatrix} \tag{4.15}$$

This requires a non-linear $h(\mathbf{x})$ function instead:

$$h(\mathbf{x}) = \begin{bmatrix} \lambda_{ub} r_{ub,x} + \lambda_{lb} r_{lb,x} \\ \lambda_{ub} r_{ub,z} + \lambda_{lb} r_{lb,z} \\ \omega_{jb} \\ \omega_{jl} \end{bmatrix} \tag{4.16}$$

4.3.4 EKF for Partial Body Movements

Also, the EKF for partial body movements can be derived from the equations of the UKF (Chapter 4.3.3). Therefore, the system matrix $\mathbf{A}$ is the Jacobi Matrix of the system function $f(\mathbf{x})$:

$$\mathbf{A} = \begin{bmatrix} 1&0 & 0 & \lambda_{ub} T\omega_{jb} & 0 & \lambda_{lb} T\omega_{jl} & \lambda_{ub} T(r_{ubz} - r_{jbz}) & \lambda_{lb} T(r_{lbz} - r_{jlz}) \\ 0&1 & -\lambda_{ub} T\omega_{jb} & 0 & -\lambda_{lb} T\omega_{jl} & 0 & \lambda_{ub} T(r_{ubx} - r_{jbx}) & \lambda_{lb} T(r_{lbx} - r_{jlx}) \\ 0&0 & 1 & T\omega_{jb} & 0 & 0 & Tr_{ubz} & 0 \\ 0&0 & -T\omega_{jb} & 1 & 0 & 0 & -Tr_{ubx} & 0 \\ 0&0 & 0 & 0 & 1 & T\omega_{jl} & 0 & Tr_{lbz} \\ 0&0 & 0 & 0 & -T\omega_{jl} & 1 & 0 & -Tr_{lbx} \\ 0&0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0&0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.17}$$

The measurement matrix $\mathbf{H}$ of the first UKF variant with eight dimensions is also constant and thus does not need to be linearized at each filter iteration. The Jacobi

matrix of $h(\mathbf{x})$, is not an identity matrix as it only contains the rotational joints' angular velocity and patient position $\vec{r}_p$:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.18}$$

The second EKF's transition matrix $\mathbf{A}$ with six dimensions is determined accordingly:

$$\mathbf{A} = \begin{bmatrix} 1 & T\omega_{jb} & 0 & 0 & Tr_{ubz} & 0 \\ -T\omega_{jb} & 1 & 0 & 0 & -Tr_{ubx} & 0 \\ 0 & 0 & 1 & T\omega_{jl} & 0 & Tr_{lbz} \\ 0 & 0 & -T\omega_{jl} & 1 & 0 & -Tr_{lbx} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.19}$$

with a linearized observation matrix $\mathbf{H}$:

$$\mathbf{H} = \begin{bmatrix} \lambda_{ub} & 0 & \lambda_{lb} & 0 & 0 & 0 \\ 0 & \lambda_{ub} & 0 & \lambda_{lb} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.20}$$

4.4 Hybrid Check for Position Surveillance

For the features under examination here, a system is assumed to have a "load recognition system" (Chapter 2.6.4). Furthermore, patients in the supine (Fig. A.5) or the beach chair position (Fig. A.6) are of consideration here. This mainly influences the partial body movements (Appendix A.10.7, function 1) since the body parts moved by the OR table are not static. Moreover, *checking only the patient's position represented as CoG is sufficient to check the entire kinematic chain* and, thus, the entire OR table joint positions for plausibility. If one of these joint positions is determined incorrectly, it leads to deviations in the expected CoG trajectories. Thus, the working load, the patient CoG $\vec{r}_p$, and the joint velocities are the inputs for the *Position Surveillance* based on the *extended observer*, including dynamic checks (Fig. 4.7). This approach enables a context-aware alarm system combining patient models with medical device data as proposed by Lee et al. (Chapter 2.3.3). Additionally, it addresses the challenge of patient body dynamics for MCPS (Chapter 2.7.4).

Furthermore, in terms of security, it is an obstacle for an attacker to determine a plausible behavior of the physical properties of a CPS because the deformation in robotic systems (Appendix A.6.2) is already complicated for the manufacturer to consider, e.g., for control loops. One can take advantage of this by learning the expected behavior of a system and, thus, the natural deviations from the ideal behavior, e.g., the deformation of a system. Compared to the physical access to a vehicle, compromising the physical layer within an OR is unlikely (Chapter 4.1.3). Physical quantities are thus a reliable source to check for plausibility since sensors cannot be manipulated easily, as the OR access is limited to hospital employees (Chapter 4.1.3).

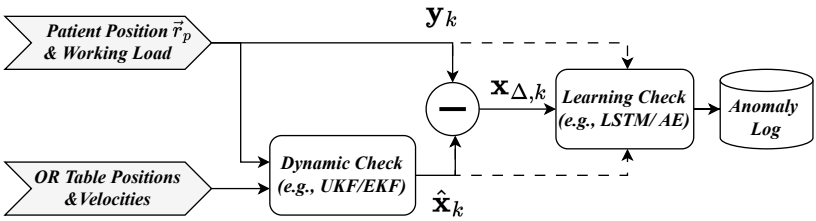


Figure 4.7: Hybrid check for patient position plausibility based on Fig. 3.11

A three-dimensional patient position $\vec{r}_p$ can be chosen as input sensor data, estimated by the dynamic check as part of $\hat{\mathbf{x}}_k$ (Chapter 4.3). Especially for heavy patients (Chapter 3.3), the trajectory deviates from the optimal one due to the deformation of the OR table and the patient's body (Fig. 4.8). Since the dynamic behavior needs to change over time, the input can be a time series of the difference of at least the patient position $\vec{r}_p$ (Fig. 4.7). The optimal length, therefore, needs to be determined, as well as the time distance between the sampling points.

Fig. 4.8 shows three scenarios for the dynamic check during this movement. The first scenario (a) shows the patient's position during a Trendelenburg movement (Chapters 4.3.1 & 4.3.2) with minor deviations from the optimal trajectory that is predicted by the whole body movement EKF (Chapter 4.3.2) or UKF (Chapter 4.3.1). It is estimated that the prediction made by the dynamic check will not completely fit the optimal or measured trajectory. In the second scenario (b), the

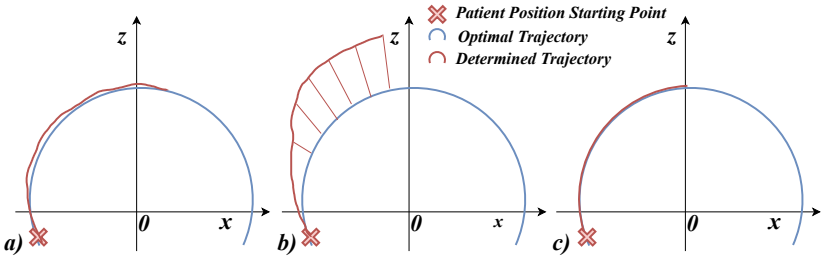


Figure 4.8: Scenarios for trajectory determination of the patient position during a Trendelenburg movement

determined and optimal trajectory deviates so that an error can be assumed, e.g., by the position sensors. In the third scenario (c), the prediction is fully consistent with the optimal trajectory. Since this is an unrealistic scenario, the deviation does not fit the expected behavior represented by the training data. This can happen if the signal or service (Chapter 2.5.3) containing the patient position is spoofed directly (Chapter 4.1.3), the dynamic check fails, or a sensor has a defect.

Finding the optimal machine learning algorithm is not targeted here and will defer between different devices for the proof of concept of the examined approach. Therefore, algorithms that have been defacto standard for anomaly detection problems are chosen based on requirements (Table 3.4). Furthermore, the performance of different algorithms needs to be examined to reduce false alarms and increase availability. In future applications, it can then be decided where optimization will likely be more efficient: Choosing the optimal machine learning algorithm, collecting more training data (“unreasonable effectiveness of data”, Chapter 2.3.2), or improving the model of the dynamic check.

In addition, architectural flexibility is required for a modular system such as OR tables to ensure accurate anomaly detection. Therefore, it is crucial to integrate modular services into the system (Chapters 3.4.2 & 3.4.4). As combined movements are condensed to a singular DoF (Chapter 3.4.2), incorporating known joint names into the algorithms is only required to identify the corresponding joint

services. When a joint is unavailable, movement in the corresponding directions will not be expected. If there is a failure or tampering, e.g., with the service discovery (Chapter 4.1.2), position estimations will become inconsistent with measurements, leading to the detection of an anomaly.

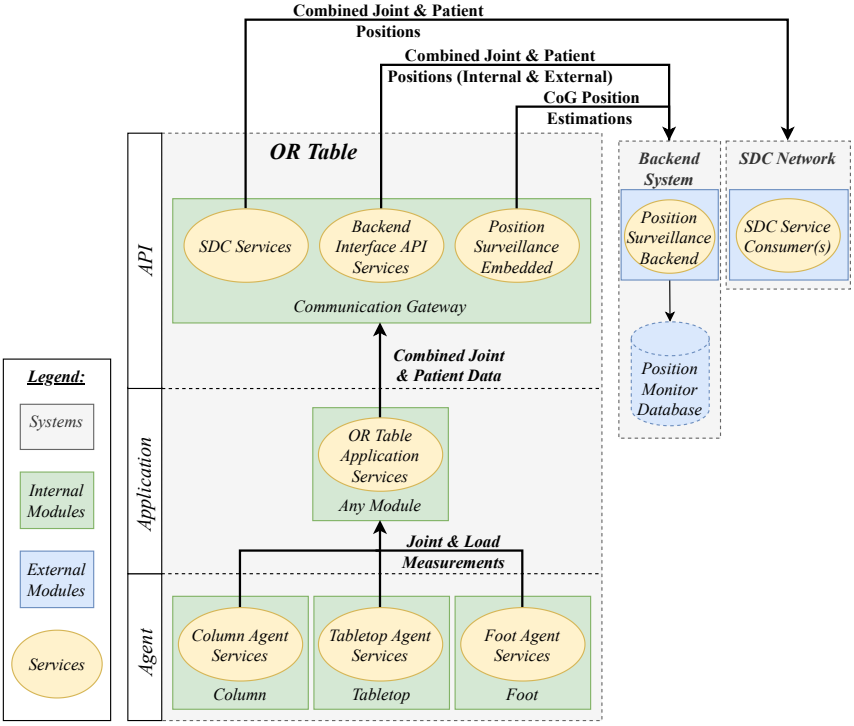


Figure 4.9: Data flow diagram for the *Position Surveillance* based on Figs. 3.21 & 3.20

Because the biggest threats result from potential attacks on the communication gateway (Chapter 4.1.4), it is insufficient to check the correctness of the internally determined positions. Furthermore, the positions provided via SDC need to be compared to the monitored positions. To fulfill that, the Position Surveillance consumes the position services provided by the SDC Interface API and the positions provided system internally (Fig. 3.21). The correct SDC transmission to

other devices is in the scope of the SDC standards. Thus, the focus here is on the correct data in MDIB communicated by the OR table under the premise that SDC takes care of a safe and secure transmission.

For single fault safety (Requirement 33), the Position Surveillance needs to be a separate software item (Chapter 45). This can be achieved by isolating it and, thus, its provided services from the others on the communication gateway or running it on a separate controller. Since the provided positions are checked on the output of the OR table, communication with other service consumers can be secured by encryption (Security) and Cyclic Redundancy Checks (CRCs) (Safety). In this way, the consumer can be guaranteed that the provided positions are valid.

Resource-intensive calculations can be executed in backend systems (Chapter 3.4.3). When combining the dynamic check as Position Surveillance in the embedded part with the learning check in the backend part, it is possible to include data such as the patient weight, which are challenging to include in a mathematical model based on expert knowledge. Therefore, this data is collected and transmitted in addition to the position data (Fig. 4.9). To facilitate a plausibility assessment of transmitted data, the *Backend Interface API* is designed to utilize data provisioning services, including those responsible for supplying positional information. By incorporating this supplementary data, verifying whether the communicated data aligns with the internal system data in the backend system becomes possible.

4.5 Mixed Architecture OR Table Design

Developing a dedicated SOA middleware for the E/E architecture for interoperable and modular OR tables (Chapter 3.4) is unnecessary, as solutions in other industries have proven suitable for safety-critical systems. According to a comparative analysis of the SOAs and corresponding middleware options conducted by Stoll [Sto21] (Table A.20), ROS2 (Chapter 2.1.4 & Appendix A.9.3) and thus DDS (Appendix A.9.4) is a suitable internal middleware for the implementation (Step 6 Fig. 2.4). In addition, with *MiroSurge* (Chapter 2.6.4 and Appendix A.9.4), DDS

has already been proven to cover the requirements of a medical robotic device. Also, to keep the development time and costs reasonable with rising regulations, the medical device industry must overcome proprietary solutions and build upon common technical standards and open-source projects platform-based (Chapter 3.4.2), as proposed for the automotive industry [Gui24].

ROS2 has emerged as the preferred choice over the previous version, ROS1, for the following reasons.

1. **Enhanced modularity and flexibility through plug-and-play capability:** Allows nodes to operate independently without relying on a central ROS master, which facilitates communication between nodes in ROS1
2. **Real-time capabilities through utilization of DDS and the support of microROS for microcontrollers:** Necessary in applications where precise timing is critical, such as in safety-critical scenarios
3. **Inherent robustness to lossy networks trough DDS:** Vital in applications where network connectivity can be intermittent or unreliable, as DDS ensures that data is delivered even in rough network conditions

4.5.1 Reconfigurable Position Surveillance

Fig. 4.10 depicts the decomposition (Chapter 3.4.2) into the *Position* and *Velocity* metrics of the *Trend* joint channel provided by the *Joint Tree* VMD (Chapter 3.4.1). The system's internal structure differs from the API MDIB as the services are provided by internal software items, which, in their sum, represent a *software system* (Chapter 2.1.2). The metrics comprise services provided by the OR table application software system. The *Position* and *Velocity* metrics use the *Move Trend To Position* service to control the movement. If an external device writes new values to these metrics via their corresponding SCOs, the *Move Trend* service is invoked. As the SDC structure is used to display the system state of a device, changing the values of different metrics can cause the same service to be called on the application layer.

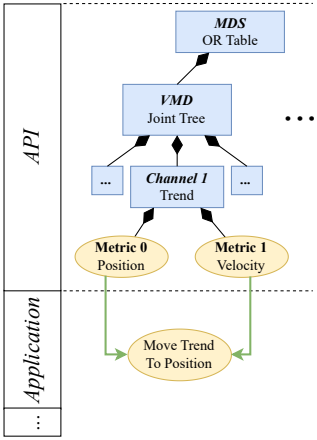


Figure 4.10: SDC API service decomposition example (Chapter 3.4.1)

The agent layer provides the services of exchangeable physical modules, whose capabilities are abstracted by the provided services. This enables a building block system based on modules as *building blocks* that provide their capabilities as service *interfaces* (Chapter 2.5.1). According to the current OR table configuration (mounted module *Tabletop 1* or *Tabletop 2*), the Trend movement provided on the application layer has a different composition. The *Move Trend to Position* service (Fig. 4.11) is either an invocation of the column service *Move Trend* in case *Tabletop 1* is mounted or a call to the service composed by the *Tabletop 2* service *Move Trend* and the column service *Move Trend*. Furthermore, a new module can introduce other joints with different movements.

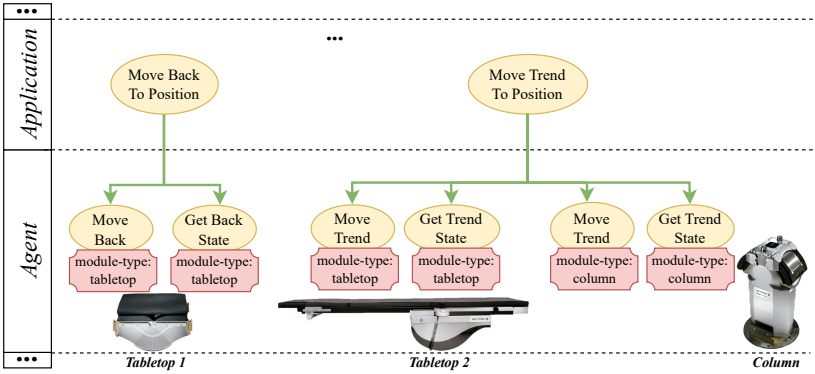


Figure 4.11: Decomposition of OR table services of corresponding modules on application layer

Also, the positions and velocities provided by the *Get Trend State* services are needed to control the combined movement. This results, for example, in increased Trend angles: In case *Tabletop 2* can move Trend to an angle of ±20 degrees and the column has the same range of motion, the overall angle is ±40 degrees. The properties ensure that only tabletop and column movement services can be combined or executed standalone and that the combination of, e.g., movements of two different columns can be prevented as it is implausible. Since this is abstracted by the application and the Joint Tree on the SDC level, other devices do not need to know the exact OR table configuration to know the whole range of motion. Although the services for movement at this level appear equivalent to those for movement at the agent layer, they involve the entire OR table structure and consider the positions of all OR table joints and links, enabling the actual control of the robotic mechanism (Chapter 2.4).

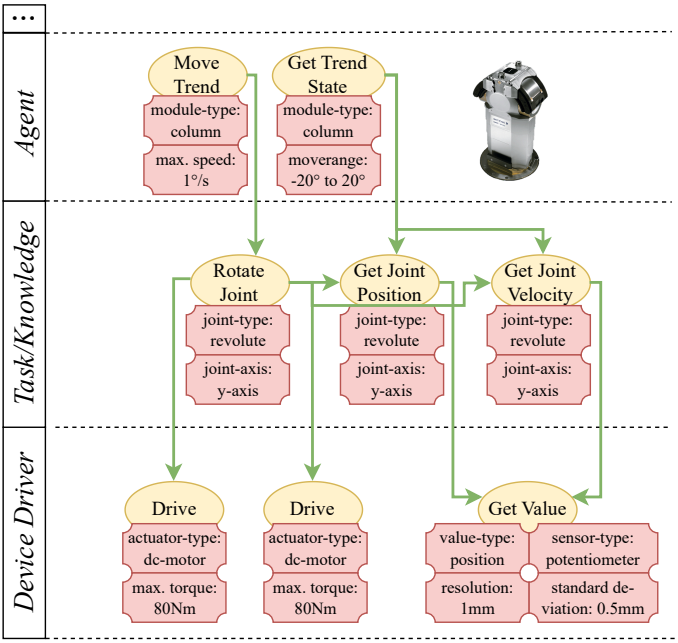


Figure 4.12: Example decomposition of column service *Move Trendelenburg* on agent layer

The *Move Trend* service (Fig. 4.12) on the agent layer consumes the service *Rotate Joint*, which is dependent on the *Drive* services from two actuators realizing the movement and the *Get Value* service of a corresponding *Position Sensor*. This framework also facilitates the interchangeability of physical system components at the task/knowledge or device driver level. Thus, components like sensors or actuators are replaceable, as they can be technically structured and decomposed.

4.5.2 Integration of Signal-based Legacy Modules

To integrate the legacy modules of OR tables (Chapter 3.4.4) in the anomaly detection, the seamless translation between the signal-based and service-oriented communication paradigms in different networks (Chapter 3.4.4, Fig. 3.22) must be ensured (Fig. 4.13). Therefore, the various modules and external devices can use different OSs and middlewares (SDC or ROS2), which must be connected using dedicated services such as the API service. This is a precondition to enable the reconfiguration based on the ontology-based service composition (Chapter 3.4.2) as the movement and joint functions of signal-based modules (Fig. 4.14) need to be uniquely identified.

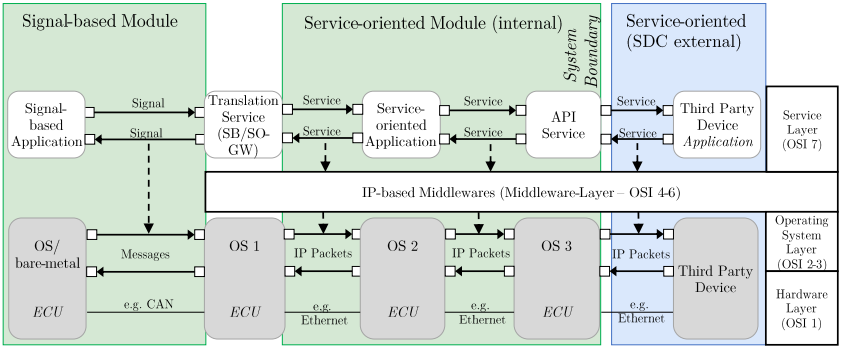


Figure 4.13: Service translation to and from signals for mixed architecture OR tables (based on Figs. 2.14 & 3.22)

Fig. 4.14 shows the exemplary translation of a movement service and its corresponding position service in the SDC of any joint channel in the VMD Joint Tree (Chapter 3.4.1). Joint position signals of the signal-based module transmitted as CAN messages (Appendix A.3.1) are translated into IP packets representing a service on a middleware layer. This first service is provided by the SB/SO-GW, representing a signal-based OR table module such as the column.

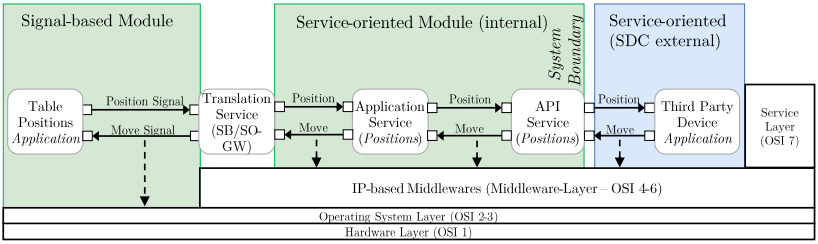


Figure 4.14: Service translation to signal-based communication for movements in a mixed architecture OR table (based on Fig. 4.13)

In the second stage, this service is provided on the application layer by any module-independent application (Chapter 3.4.2) that can use this service to create composed services. These services are internal to the OR table system and cannot be used by another device. Therefore, the application service needs to be translated to the external interface middleware, which is, in this case, SDC (Chapter 3.4.1). Thus, the communication gateway for SDC bridges the service-oriented middleware into SDC. Hence, the SDC service calls abstracts, which module is running the service, which communication paradigm, or which technology is used (Chapter 3.4.2). The movement is called in the same way as the position service in reverse order, starting at the interface SDC provided by the communication gateway for SDC.

4.5.3 OR Table Digital Twin

For the examined scenario of the CoGs of several parts of the OR table and the patient, generating "Ground Truth" data is intricate to achieve with real system measurements. Each component of the OR table and each part of the patient's body must be measured with a separate load recognition system. In addition, the z-axis value cannot be measured based on the gravitational force applied by this component, as it is only in the z-direction, which can only be estimated this way. Flexible elements and inhomogeneous mass distributions complicate it even further. Furthermore, using a real OR table for training data from today's perspective is connected with unreasonable effort. Determining the constants and placing the weight in several position possibilities is not feasible due to the effort involved. Hence, the proof of concept is carried out simulatively here, as the synthetic data is known.

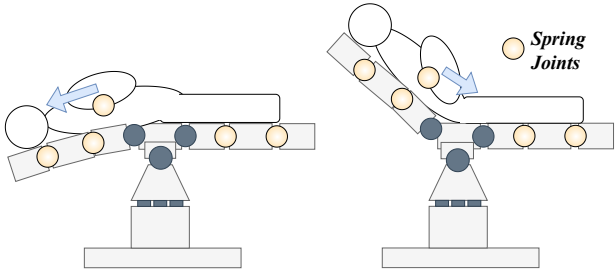


Figure 4.15: Patient and OR table models during a back movement

The OR table is modeled as a rigid body system, while *spring joints* distributed along the tabletop [Hal23] (Fig. 4.15) are used to realize the *finite segment method* (Appendix A.6.2) to model deformations of the OR table and a patient's deformable body parts like the abdomen (Chapter 3.3). Spring joints are virtual joints attached to the model and have a spring-like behavior. By adjusting the springs' stiffness and damping parameters, the model's flexibility can be controlled to approximate the real system behavior. A critical aspect of patient models is the representation of flexible elements, such as muscles and tissues. Therefore,

spring joints are also used here to model the flexibility of these elements, which are mainly dependent on the patient's weight. As a reference for the mass distribution of the human body, the IEC 60601-1 [IEC20] (Chapter 2.1.3) is used (Fig. A.7). Furthermore, the inhomogeneous mass distribution of the human body is neglected here, so the CoG of a body segment is assumed to be in its center.

Table 4.6: Extended simulation tool (Chapter 2.1.4) comparison based on [Lab22] (Table A.24), [Kin22] (Table A.25) [Käf17] [Hal23] (Rating for requirements in Table 4.7: 0 = not fulfilled, 1 = partially fulfilled, 2 = fulfilled)

Tool	Costs	Effort	Community	Versatility	Physics	Visualization	Σ
MATLAB	0	2	1	1	2	1	7
Blender	2	1	1	2	1	2	9
Gazebo	2	2	1	2	2	2	**11**

According to the evaluation of the simulation tools (Table 4.6), Gazebo (Chapter 2.1.4) meets the requirements (Table 4.7) for the proof of concept of data synthesis best for anomaly detection and service orientation in OR tables. Therefore, the simulation model will be built using Gazebo to enable accurate and efficient simulations, which is particularly relevant for creating physical data from simulations that feedback into the prototypical implementation of the mixed architecture OR table modules. In a subsequent step, the model simulates physical data for anomaly detection (Chapter 4.3) (see also [Shk23] & [Kin22]). In this instance, the simulation model represents a *digital twin* (Chapter 2.1.4) of an existing OR table, which is reverse-engineered to represent the behavior and dynamics of the system.

The digital twin of the OR table and the prototype of SOA can be used to synthesize physical data, including the position and velocity data of the OR table joints and patient's position. Therefore, issues resulting from using a state-of-the-art OR table to evaluate the anomaly detection on the example of *Position Surveillance* (Chapter 6.9 & Appendix A.12.4) can be bypassed with a digital twin.

Moreover, using sensor signals from a real system does not necessarily mean they are optimal, as evidenced by examples like On-Board Diagnostics (OBD) messages with variable sampling times [Web19].

Table 4.7: Simulation tool requirements

Requirement	Description
Costs	The cost of the tool can be an obstacle, especially if multiple users work with the tool and require a license.
Implementation Effort	The implementation effort can be reduced if a tool offers ready-to-use capabilities such as rigid body dynamics.
Community	If a tool has a large and lively community, it is more likely to get help, e.g., via community forums. The tool's longer support with updates can also be an effect.
Versatility	Tools can be combined with other third-party tools, extended with plugins, or directly by the user if they can write scripts to extend the functionality of a tool.
Physics	The physical accuracy of a simulation process affects the accuracy of anomaly detection. Tools that focus on 3D visualization reduce the calculation effort in favor of performance, and the focus is on visual appearance (Chapter 2.1.4).
3D Visualization	A 3D visualization of the simulation helps with setting up or troubleshooting the simulation, especially for motion-related topics with mechanics. It also offers the option of capturing images for documentation purposes.

4.6 SDC Network IDS

For the anomalies in the communication of medical devices in an SDC network (Chapter 3.3.3), the detection of three anomaly patterns (Table A.10) can be integrated with the mixed architecture OR table (Chapter 4.5.2) and the digital twin (Chapter 4.5.3). Although the underlying middleware is ROS2 and thus DDS and not SDC, the communication patterns examined are realizable with both, so the SDC design is similar. Therefore, simulations for a remote control, an angiography system, and an OR table are ROS2 nodes. At the same time, the

NIDS supervisor, also a ROS2 node, listens to their provided services and IDS sensors in the network (Fig. 4.16). Furthermore, the OR table and angiography system have a Gazebo model integration to simulate their positions (Fig. 4.17).

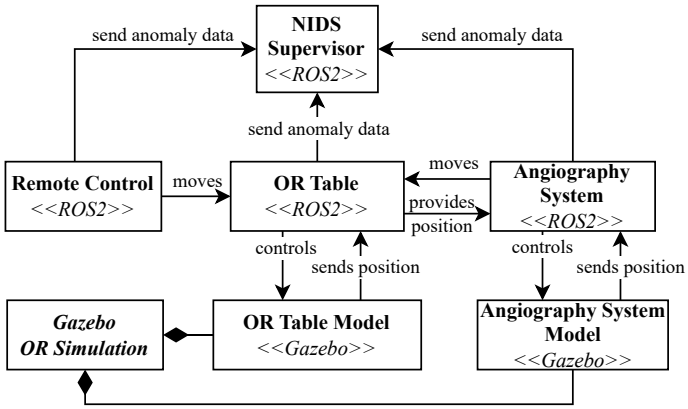


Figure 4.16: Object diagram for connected medical devices using ROS2 based on Fig. 3.12

Payload Plausibility Check for OR Table Joint Positions

Payloads of services can be analyzed to detect any anomalies in the time series (Chapter 2.3). For instance, the NIDS supervisor can also monitor the *Provide position* service (Chapter 3.3.3, Fig. 3.12) of the OR table. Whenever the position of a joint behaves unexpectedly (Chapter 4.3), an alarm is generated, which enables other devices in the OR to check the plausibility of the OR table.

Publish/Subscribe Causality for OR Table Movements

In a publish/subscribe communication, the relationship between changes in the payload of two correlating services can be determined. The NIDS supervisor can detect any unusual movement in an OR table by monitoring the *Move* service provided by the table. Anomalous behavior is detected when the provided positions indicate a moving joint that has not been requested to move by the remote

control. To achieve this, the timestamp of the last movement call is compared with the occurrence of joint movement to determine whether the remote control is responsible for the movement. Therefore, velocity is calculated by measuring the rate of change of the provided position over time.

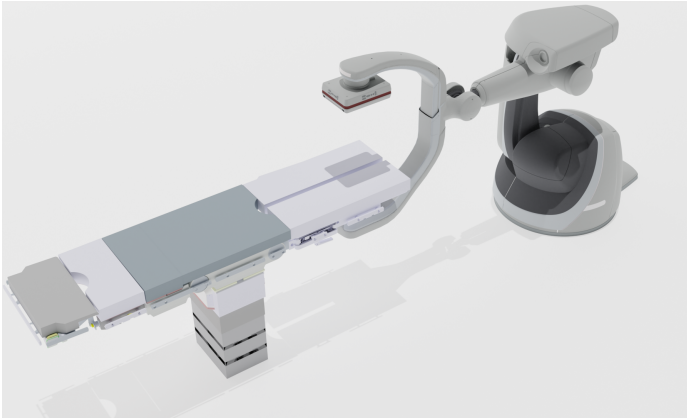


Figure 4.17: 3d model visualization of the OR scenario in Gazebo rendered in Blender (Chapter 2.1.4) based on [PRGS22]

Context Plausibility of a Locked OR Table

The state and payload of services of interoperating devices are combined to create an SoS context, which are checked for plausibility within a particular system context [GS22]. Thus, an anomaly is detected if the states and payloads are not plausible within a given context (Chapter 3.3). Here, the NIDS supervisor listens to a HIDS of the OR table, which provides the current lock state when it detects a call to the motion service. The motion service call should not occur while the OR table is locked, preventing movement. The HIDS provides information about locked and unlocked states as a service. If the position of the OR table changes while it is locked, an alarm is generated.

5 Prototypical Implementation of the OR Table Position Surveillance

5.1 Mixed Architecture OR Table Prototype

To enable the safe and secure integration of legacy modules (Chapter 3.4.4), an SB/SO-GW is implemented as ROS2 node [PVR+22] [Kin22]. Reading and storing the signal values in CAN messages and presenting them as services for the ROS2 components facilitates communication between signal-based and service-oriented modules. Since this implementation is purely virtual, it would be realized in a physical prototype as a dedicated hardware component (e.g., separate ECUs). The signal-based software runs in a virtual environment as well. Moreover, it is possible to separate the software nodes so that all components run in different execution environments, as in a real-world application. This involves running the proprietary software on dedicated hardware, connecting CAN to SocketCAN [Git22] with the SB/SO-GW, and enabling software separation.

Fig. 5.1 depicts the translation of a movement message via the remote control of the OR table to the back joint of the tabletop moving the digital twin in Gazebo. The data provided by services is stored in a separate node of the SB/SO-GW and published cyclically and event-based on the CAN bus. The service-oriented tabletop application controls the digital twin, whereby a movement is triggered using a simulated remote control that sends movement signals to the signal-based OR table software. The tabletop service receives messages containing the desired joint position and speed through ROS 2 topics. This invokes a *"Joint Trajectory*

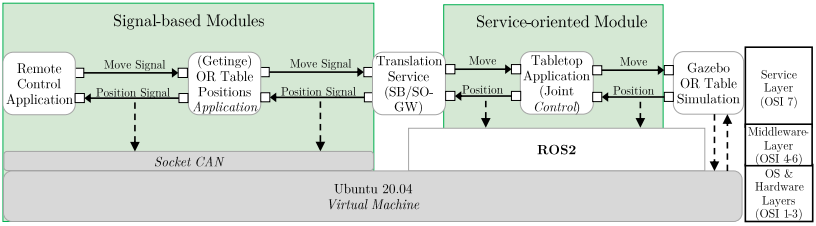


Figure 5.1: Translation from signals into services (sequence in Fig. 5.4) to trigger the movement of the back joint of the OR table digital twin (Fig. 5.2) based on Fig. 4.13 (simulation/digital twin: white, internal: green, Fig. 3.1)

Controller" [Rob18] integrated into the Gazebo model to control the movement and periodically publish the current joint states. These joint states are converted backward into CAN messages for each joint status containing the position signals. Another ROS 2 node can be used if it realizes the same interface to replace the exemplary tabletop module in this system. In the signal-based subsystem, the *OR Table Positions Application* ECU generates the actual movement command messages on the CAN bus. When a joint movement is triggered in this subsystem, the SB/SO-GW translates the command message into a service invocation of a move joint command in the service-oriented tabletop application. Using the same topic names is necessary to maintain compatibility with the SB/SO-GW (Chapter 5.1).

If signals cause actions, the concrete implementation must be considered. In the case of the joint movement, it is necessary to have a cyclic and continuous signal stream to enable continuous movement. The underlying safety concept allows for a movement only if the user commands it and stops as soon as the signal stream stops. Suppose one signal would trigger the movement while another is intended to halt it. The movement may not be appropriately halted if the stop signal is lost due to an error. This situation could result in patient harm. Hence, it is crucial to carefully design the translation process for each of them regarding safety-critical functions activated by signals or services. This, in turn, increases the effort required for risk measures and management (Chapter 2.1.3).

The OR table digital twin (Chapter 4.5.3) is created as a Gazebo model (Fig. 5.2) using the relationships for rigid body systems (Chapter 2.4) in URDF (Chapter 2.1.4).

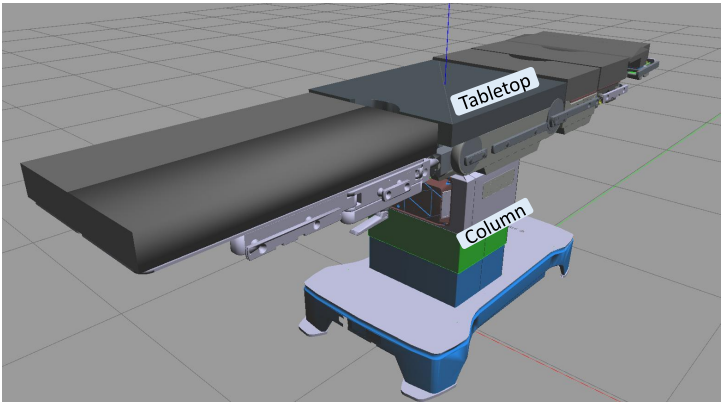


Figure 5.2: Visualization of the OR table digital twin in Gazebo

As the ROS2 integration for Gazebo does not support all features of the ROS integration, the OR table model is integrated into the ROS environment. This enables the use of "*Effort Controllers*" [Rob20] to control the speed or position of individual joints, which is necessary to evaluate controllability to simulate meaningful data for anomaly detection (Chapter 3.2.4). In the context of Gazebo, effort refers to the amount of force in prismatic joints or torque in revolute joints applied to a joint (Chapter 2.4.2). Furthermore, this requires the introduction of a ROS to ROS2 bridge [Rob23] on the service-oriented part (Fig. 5.1) to support the ROS environment (Fig. 5.3).

Including the Gazebo OR table digital twin (Chapter 4.5.3) and the SB/SO-GW, the concept for a mixed architecture OR table (Chapter 3.4.4) is realized with the integration of proprietary OR table software (Figs. 5.3 & 5.4). The concept is based on the scenario that an installed base of OR table columns is already in the field, and a new tabletop is being developed for these [PVR+22]. To realize this approach, the software embedded in a state-of-the-art OR table provided

by Getinge is running in a virtual environment within a virtual Ubuntu 20.04 machine (Fig. 5.1). This software represents the column and foot module's legacy signal-based ECUs (Chapter 3.4.4). Both are implemented as separate Linux applications, running in a proprietary Software-in-the-Loop (SiL) environment. They are linked via Linux SocketCAN, enabling the conventional broadcast of each CAN message. The service-oriented tabletop application implemented as ROS 2 node represents the newly introduced module connected via the SB/SO-GW to the signal-based modules. The virtual machine was chosen because of the improved portability of the entire setup. If more computing capacity is required, e.g., due to 3D graphics, the setup can also be installed on a physical computer. Ubuntu is preferred over macOS and Windows, as Gazebo has only been implemented for Windows on a trial basis. macOS offers less flexibility than Ubuntu, and setting it up as a virtual machine is also cumbersome [Kal22].

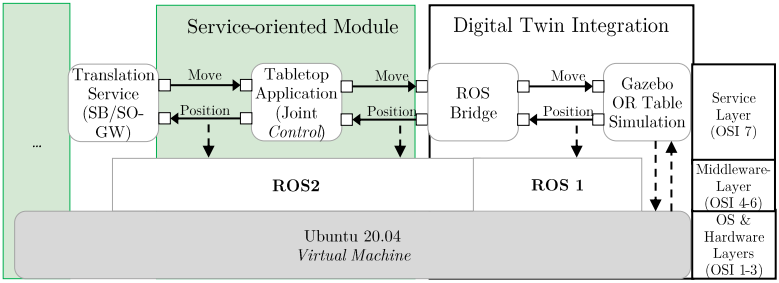


Figure 5.3: Adaption for position controllers in Gazebo ROS Integration (sequence in Fig. 5.4)

The service-oriented tabletop application realizes the *Move Joints* function (Appendix A.10.7, function 1) as a proof of concept, while the registration processes are out of scope here (Chapter 3.4.4) and "stubbed" with the signal-based pendant of the tabletop. Thus, the joint movements for the back and legs are translated into service invocations by the SB/SO-GW, leading to the corresponding joint movements of the digital twin in the Gazebo simulation (Fig. 5.3). The corresponding joint states are then translated from the Gazebo simulation over the SB/SO-GW into CAN signals again.

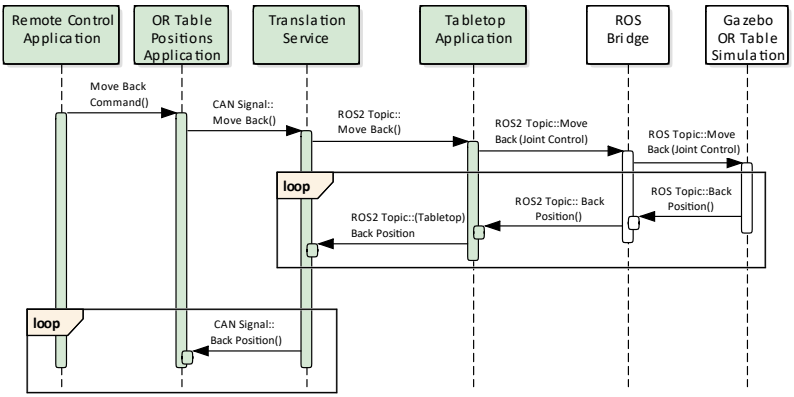


Figure 5.4: Sequence diagram for the control of the back joint (Figs. 5.3 & 5.1)

To integrate the modular OR table (Chapter 4.5.1) with the proposed mixed architecture into an SDC-based interoperable hospital environment, the ROS2-based plug-and-play approach by Stoll et al. (Appendix A.9.3) is used as an extension in the previous setup (Fig. 5.5) [PSS23]. As proof of concept for the dynamic service composition, the *Application* service *Move Trend to Position* (Fig. A.32) is implemented (Appendix A.10.7, function 2). In the considered scenario, two tabletop module applications, where one is capable of the *Move Trend* and *Get Trend State* service and one column module application with the same services, are realized as ROS2 nodes. Since the service composition of other service layers follows the same procedure, the lower service layers are substituted with the OR table digital twin.

Utilizing the plug-and-play approach by Stoll et al. entails individual YAML Ain't Markup Language (YAML) files for each module. In alternative implementations, formats like XML can also be considered for exchanging these capabilities. These files contain information about their capabilities that describe specific services and their dependencies, thereby representing the ontology. As an illustration, the column module includes the service named "Move Trend", categorized under

"moduletype:column", with a maximum position of 30 ° (Listing 5.1). By incorporating an additional tabletop module, which offers the same service with a maximum position of 10 °, the dynamically composed service *Move Trend to Position* on the application layer (Chapter 3.4.2) allows for a total Trend angle of 40 °. So, when replacing the tabletop module again without Trend support, the *Move Trend To Position Application* is only "composed" of the *"column"* type services. If the Position Surveillance relies on the Trend service, it is independent of the module(s) providing the service.

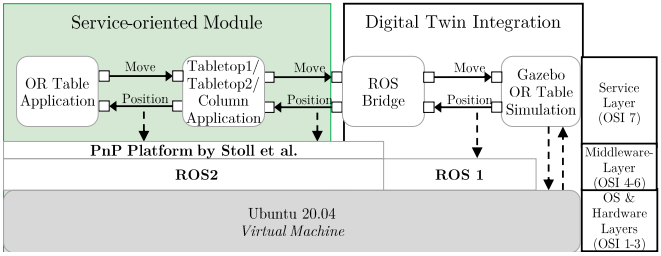


Figure 5.5: Service invocation of combined movements with two tabletops and one column

Listing 5.1: Example YAML content for service "Move Trend"

```
1     ros__parameters:
2     capabilities:
3     - '{"name":"move trend", "moduletype":"column", "category"
          :"agent/joint", "position_max":30, "position_min":-30,
           "parent-joint":"height"}'
4     dependencies:
5     - '{"name":"rotate joint", "category":"task/joint", "joint
          -type":"y-rotation", }'
```

5.2 Implementation of Dynamic Checks for Partial Body Movements

The UKF and EKF (Chapter 4.3) have been implemented (Step 6, Fig. 2.4) specifically for partial body movements to test the validity of the approach. Partial body movements introduce more uncertainty and non-linearity into the model than whole body movements, making them a more intricate yet realistic scenario for state estimation. The EKF and UKF implementations are structured as ROS2 nodes, seamlessly integrating with the digital twin through the dynamic service binding (Chapter 5.1). These nodes subscribe to the positions and velocities of the CoG derived from the Gazebo simulation. For the aim of evaluating and training data-based models, crucial data, such as the estimated CoG positions and table velocities, are logged into a Comma-Separated Values (CSV) file. This archival approach facilitates the assessment of algorithm performance and sets the groundwork for future model refinement and development. The recorded data can be leveraged for comprehensive analysis (Chapter 2.2.4), including examining influences such as measurement noise.

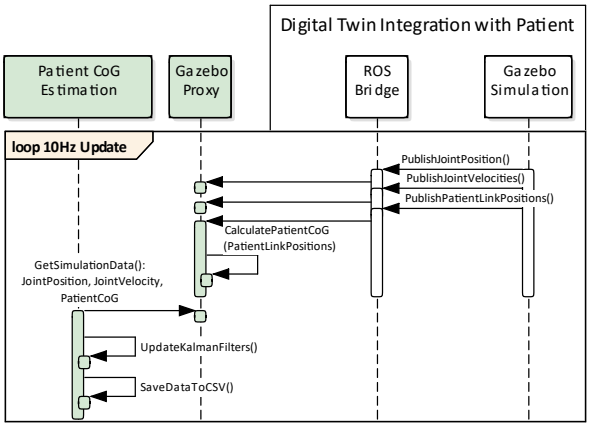


Figure 5.6: Sequence diagram for state estimation with KFs and simulation using the *OR table digital twin integration (Fig. 5.3)*

The UKF and EKF are implemented as ROS2 Python nodes using the package *filterpy* [Lab22] and enable the system to estimate and update the state of the simulated environment. The simulation data are transmitted through the ROS1 Bridge and intercepted by a dedicated class *Gazebo Proxy* (Fig. 5.6). The Gazebo Proxy is an intermediary between the Gazebo simulator and the dynamic check node *Patient Position Kalman Filter*. Its primary function is to capture the simulated data and facilitate communication between the Gazebo environment and the dynamic checks. The update frequency of the filters is 10Hz, providing a balance between real-time responsiveness and computational efficiency. This periodic update ensures that the state estimates remain current and accurate, reflecting the dynamic nature of the simulated environment. Furthermore, the 100ms interval aligns with the demands of real-time applications and the internal communication latencies of an OR table [PVR+22], allowing for timely adjustments to the estimated states based on the incoming sensor data.

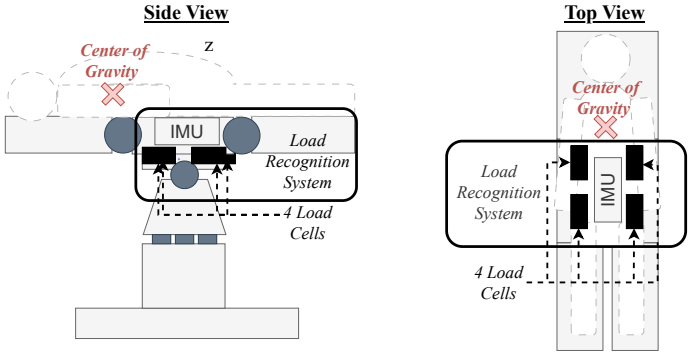


Figure 5.7: Load recognition system [DSG+22]

The Gazebo Proxy simulates a *load recognition system* (Chapter 2.6.4) that measures the CoG of the load/patient on the OR table. The actual load recognition system consists of an Inertial Measurement Unit (IMU) and four load cells mounted between the tabletop and the column. As the individual sensor measurements are not required for the CoG monitoring, they are summarized as a single sensor measurement with the patient CoG as direct output (Fig. 5.7).

5.3 Distributed Position Surveillance

A distributed system (Fig. 5.8) realizes the Position Surveillance (Chapter 3.4.3), while the prototypical implementation omits the SDC supervision as the approach is considered technology- and protocol-independent. Therefore, simulated positions are not distinguished between internal OR table signals or services and externally provided position data over SDC services. The dynamic check as ROS2 node (Chapter 5.2) realizes the patient position estimation. The implementation realizes the distributed hybrid check since the standalone dynamic/learning check variants are reduced alternatives without backend interaction. First, the available

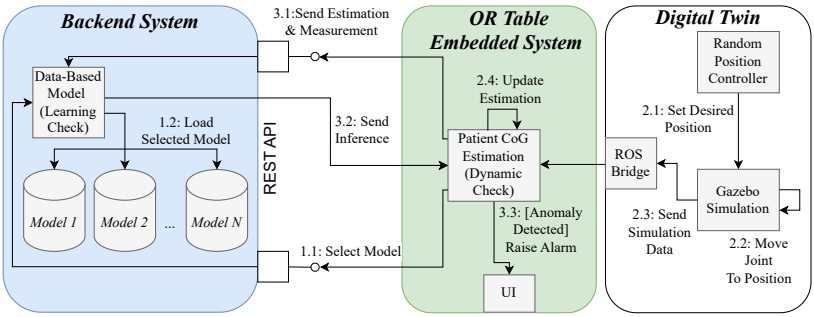


Figure 5.8: Communication diagram for distributed hybrid anomaly detection Implementation with interfaces between backend system and OR table based on Fig. 4.9 (API service layer only, external: blue, internal: green, simulation/digital twin: white) based on [PZSS24]

data-based models are selected over a REpresentational State Transfer (REST) (Appendix A.9.4) interface (*1.1 Select Model*) based on the current determined patient weight and preferred data-based model (LSTM, IF, or AE) and loaded into the *Data-Based Model Learning Check* (*1.2 Load Selected Model*). With another ROS2 node, random positions are commanded (*2.1 Set Desired Position*) to the Gazebo OR table digital twin (Chapter 4.5.3) that executes the desired joint movement (*2.2 Move Joint To Position*), which is sent then from Gazebo (*2.3 Send Simulation Data*) to the *Patient CoG Estimation Dynamic Check* (*2.4 Update Estimation*) using the ROS Bridge (Chapter 5.2).

The dynamic check (Chapter 5.2) estimates each patient CoG that is then sent to the *backend system* (*3.1 Send Estimation & Measurement*), which evaluates and sends the inference back to the node for the patient estimation (*3.2 Send Inference*). If the estimated value is an anomaly, the User Interface (UI) raises an alarm triggered by the dynamic check (*3.3 Raise Alarm*).

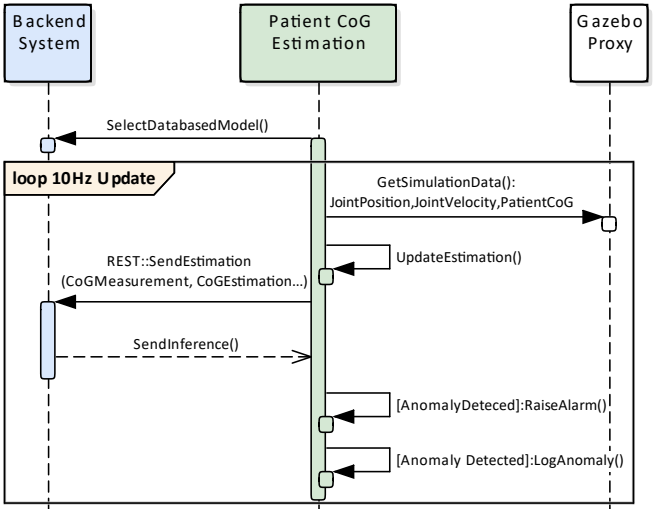


Figure 5.9: Sequence diagram for state estimation with inference evaluation over a REST service call on backend system

In the test setup, a controller service repeatedly targets a position in random time steps to simulate the user's interaction with the OR table. The *Patient CoG Estimation* updates all dynamic filters cyclically every 100ms (Fig. 5.9) based on the Gazebo Proxy data (Chapter 5.2), checks for an anomaly on the *backend system* and logs the current simulation and check data for later evaluation.

If new models are available, they can be uploaded after the training (Chapter 5.4) to the backend system so they can be invoked over a provided service by another external system (Chapter 5.3). A more flexible learning approach, which sends the collected data directly during simulation to the backend system to learn from data

after the collection or enable online learning (Chapter 3.4.3), is not realized here. As the data is collected before a simulation, all training data is available during the initial training of the models. Therefore, online learning during the simulation does not add any additional insight to the evaluation. In addition, online learning is already state of the art, so there is no need to prove its technical feasibility.

5.4 Implementation and Training of Learning and Hybrid Check for Partial Body Movements

LSTMs, IFs, and AEs are trained as hybrid check and pure learning check (Chapter 3.3.2), whereby the selection of features determines whether the model is either a hybrid or a pure data-based model. All chosen algorithms may use the same inputs: a time series of window-length n or a single time step at iteration k (time point). In addition, as the context of the variables and correlation is considered mandatory to check for plausibility, only multivariate variables are used. The variables of the dynamic checks (Table 4.5) and the position and velocity measurements are considered features for the learning checks. Moreover, only the measurements are input for the pure data-based models. The difference between each measurement variable and the corresponding estimation is also considered a possible feature for hybrid checks (Chapter 3.3.2).

The same setup for the dynamic check implementation is used to create the data for training the learning checks (Chapter 5.2), whereby the training is done offline using the data collected during simulation. The data collected in CSV files contain the different inputs for the specific data-based models. Since deviations are more likely to occur during motion, the acquired data is upsampled to ensure that windows of tensors, which are statistically underrepresented, occur with the same frequency as overrepresented windows. This process shapes the data distribution from a generally Gaussian one to a uniform one, so different upsampling metrics are needed for different algorithm outputs. The AE will reconstruct a window

based on a given input window. Therefore, the data of a window must be interpreted in the form of a metric usable to determine the frequency of the data. Thus, l_k norms (Chapter 2.3.1) are applicable. Here, the l_1 norm is used (MAE) for the difference of the measurement to the estimation of the x value. For non-time-variat outputs like for the LSTM or IF design here, the absolute error between a single measurement in x and estimation in x is used. This is the procedure of choice if the difference is not part of the feature set, e.g., for the pure learning checks.

5.4.1 Long Short-Term Memory (LSTM) Network

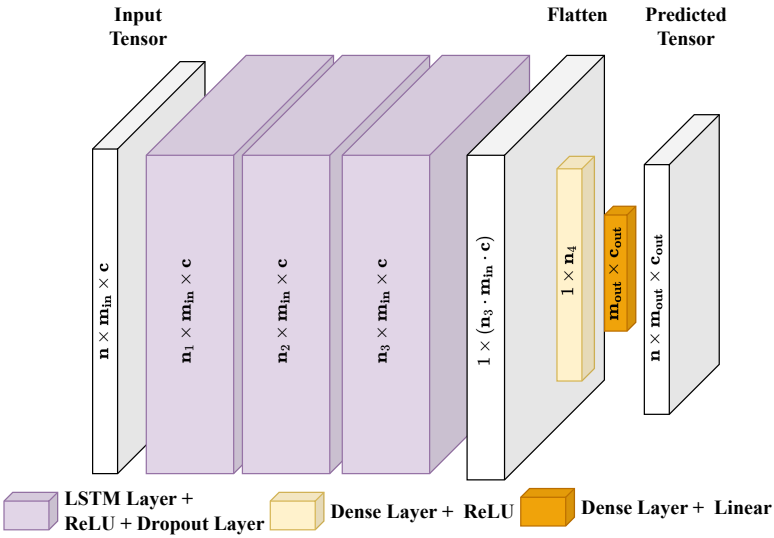


Figure 5.10: Implemented LSTM architecture [PZSS24] (n: number of windows, m_{in}: window size, c: number of channels/features of input, n_i: neurons number of layer i, m_{out} window size of predicted output tensor, c_{out} number of channels/features of output)

The LSTM is implemented in Python using the library TensorFlow [Ten23] starting with three LSTM layers, whereby a dropout layer follows each to avoid overfitting (Fig. 5.10). The LSTM is appropriate for several setups for features to combine with the dynamic check output. Since there are four measurement values

(CoG x and z, joint velocity back and leg) as well as six estimation features from the KFs (upper body CoG x & z, lower body CoG x & z, joint velocities back and leg), these can be combined arbitrarily including a variable time series (Chapter 6).

As activation functions, each LSTM layer contains a Rectified Linear Unit (ReLU) activation function. The first Dense layer uses a ReLU activation function, and another Dense layer at the output uses a linear activation function. Furthermore, other activation functions that allow modeling of non-linear behavior, such as *Swish* instead of ReLU in the LSTM layers, are also possible.

5.4.2 Isolation Forest (IF)

The IF is implemented in Python with the library Scikit-learn [CGV+23] and its dependencies. As inputs, either the differences or the measurements themselves can be used. The hybrid variant uses a two-dimensional input of the difference of the measurement to the estimation in the x and z directions, whereby the standalone variant is only trained on the position measurements. A single time step is chosen to keep the IF implementation minimal to evaluate the low-resource variant of the hybrid checks. As the IF does not need a predefined threshold and only classifies into inlier and outlier data, the training data primarily influences the evaluation metrics such as FPR and FNR, so the manual influence for subsequent fine-tuning is limited in inference time.

Another essential aspect of implementing an IF is the choice of hyperparameters. The number of trees in the forest, set to 100, and the maximum depth of each tree are two key hyperparameters that affect the algorithm's performance. The number of samples provided to the IF indirectly determines the maximum tree depth, which here is the maximum number of samples. In addition, the subsampling size, which determines the number of data points sampled to create each tree, can also impact the results. Finding the optimal set of hyperparameters involves experimentation and evaluation on a validation set, which is not the aim here.

5.4.3 Autoencoder (AE)

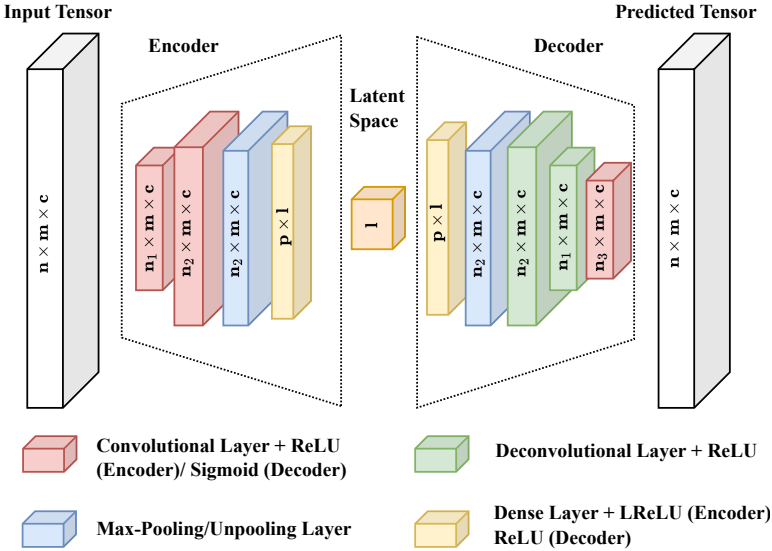


Figure 5.11: Implemented AE architecture [PZSS24] (n: number of windows, m: window size, c: number of channels/features, n_1: neurons number of 1st conv. and 2nd deconv. layer, n_2: neurons number of 2nd conv. and 1st deconv. layer, n_3: neurons number of 3rd conv. layer, p: pooling layer size, l: latent space dimension.)

The AE is implemented in Python using the PyTorch library [PGM+19] to reconstruct a desired time series of the provided estimations and measurements. Since 1D Convolutional Neural Network (1DCNN) AEs are a popular choice to analyze time series (Chapter 2.3.2), it is also considered a suitable design for this application. Two convolutional layers followed by a max-pooling and dense layer comprise the encoder. In comparison, the decoder has a mirrored structure based on an unpooling and dense layer followed by two deconvolutional layers and completed by another convolutional layer (Fig. 5.11). The convolutional, deconvolutional, and dense layers use ReLUs, Leaky Rectified Linear Units (LReLUs), or sigmoid as activation functions. The hybrid variant uses the position difference as input of a specified window size, similar to the other data-based models. As a result of the prediction, the AE reconstructs a window so that anomalies cannot be reconstructed since they were not part of the training data (Chapter 2.3.2).

6 Evaluation and Discussion

The evaluation process entails comparing the presented dynamic and learning checks and exploring hybrid checks formed through their combination (Chapter 3.3) [PZSS24]. The overarching objective is identifying a strategy for achieving the desired outcomes. Furthermore, the aim is to detect anomalies while reducing the False Positive Rate (FPR) rather than to optimize the actual state estimation as examined by Koch [Koc23] or to identify each anomaly data point. Based on the evaluation (safety and security-related verification and validation, Fig. 2.4), it is possible to determine the next improvement steps. Selecting the most promising "Adjusting Screw" among three crucial elements is vital: the learning check, increased data volume, and enhancements to the dynamic check.

The algorithms are evaluated on the deviation of each signal's estimation, respectively, reconstruction to the measurements. The load recognition sensor is a summary of a sensor system consisting of load cells and an Inertial Measurement Unit (IMU) (Chapter 5.2, Fig. 5.7), so the *measurements* in this context are simulation results from Gazebo. Each data point is evaluated to determine whether it is an anomaly to retain comparability throughout the different algorithms. Optimization of the evaluation of the difference is considered necessary for the actual product rather than for evaluating the models and algorithms examined here, so finding the ideal parameter and hyperparameter set is not targeted.

Furthermore, it is anticipated that the measurement is close to the ground truth in non-anomaly scenarios and only overlaid with white noise. Therefore, white noise is added to the ground truth of the Center of Gravity (CoG) positions to simulate the measurements. In anomaly cases, the measurement contains deviations to the ground truth caused by system external triggers (Chapter 2.3). The quality of measurement data depends on the quality of the sensors and the signal processing,

which is not examined here. The noise is added to investigate the assumption that anomaly detection using pure data-based models is inadequate for data with sensor noise (Chapter 2.3.1). Therefore, the focus is on the qualitative rather than the quantitative influence and on obtaining a more realistic scenario than using ground-truth data, which is unavailable in a real system during runtime.

The Isolation Forest (IF) uses path lengths to group data, automatically determining the anomaly detection threshold (Chapter 2.1.3), whereby the Long Short-Term Memory (LSTM) network and the Autoencoder (AE) require a predetermined threshold to categorize anomalies. This threshold is based on either the reconstruction error of the AE variants, the LSTM prediction of the difference between the dynamic check output and the measurement, or the absolute error of the pure LSTM position prediction to the measurements.

Each feature, such as the CoG in the x and z direction, is evaluated separately using a dedicated threshold. Hence, a data point is classified as an anomaly if the model's output exceeds the threshold value in either of these signals. The 99th percentile of the prediction, respectively, reconstruction errors based on the training data, has been chosen, similar to the approach used in [RSZ21]. This method has been preferred over a manual threshold due to its ability to adapt to new data and create objective comparability between different approaches during evaluation. Although the percentile choice can be changed, it is essential to maintain a negligible FPR for applications in hospitals, as false alarms [Cli07] may lead to alarm fatigue, causing critical alarms to be missed [SF13] and reducing the quality of care [LSC+12] (Chapter 2.1.3). Consequently, a percentile above 90 corresponds to this target, which must be achieved, as over 70% of current alarms are false alarms (Chapter 2.1.3). The measures based on the detection of anomalies must be carefully selected. Accepting false positives to detect all true positives can lead to lower system availability and may even be worse than a possible cyberattack [PRGS22].

If a lower percentile is selected, the number of false positive results increases accordingly by definition, as the corresponding threshold value is only based on normal data. Theoretically, using a 90th percentile results in an overall FPR of

10% since the most deviating 10% of the training data is considered an anomaly. Therefore, a percentile near 100 is required to achieve an FPR of 0%, but then edge cases of normal behavior leading to significant deviations drastically increase the threshold. As an example, when covering 100% of the deviations within the normal data for the Unscented Kalman Filter (UKF) (Chapter 6.1), the threshold is doubled in comparison to the 99th percentile from ∼4.35cm to ∼8.59cm in x direction and from ∼5.91cm to ∼12.90cm in z direction. Thus, minor anomalies are not detectable anymore (Chapter 6.1). Defining an appropriate threshold based on a percentile for medical devices depends on the use of the device. If the edge cases of deviation from normal behavior only occur in special applications of the device, a percentage threshold value is appropriate. The borderline cases can then be covered by special treatment. This requires statistics on the usage behavior of a medical device. The selection of the 99th percentile filters out these edge cases to enable an objective evaluation of the algorithms.

Table 6.1: Position anomaly scenarios (Fig. 6.2) based on Table A.9

Anomaly	**x-axis**	**z-axis**	**Potential Scenario**
Positive/Negative Step Plateau	(1) Jump to $x = -0.3\,\text{m}$ between 100-103s (3 sec.)	(2) Jump to $z = 1.2\,\text{m}$ between 140-143s (3 sec.)	Indicates a sensor defect or a potential attack where certain positions are set.
Plateau	(4) x stuck at a value between 245-250s (5 sec.)	(3) z stuck at a value between 163-168s (5 sec.)	A defect or manipulation attempt that keeps the position value stuck at the current value while recovering the actual value after the anomaly.
Positive/Negative Ramp Plateau	(5) x increase from 300-310s (10 sec.), decrease from 320 to 330s (10 sec.) with $\approx 1.33\,\text{cm/s}$	(6) z increase between 400-410s (10 sec.), decrease from 410-420s (10 sec.) with $\approx 1.33\,\text{cm/s}$	Indicates a possible sensor error or an attempt by an attacker to manipulate the position to a desired value gradually.
Positive/Negative Ramp with Jump Back	(7) x increase between 500-510s (10 sec.) with $\approx 1.33\,\text{cm/s}$	(8) z increase between 550-560s (10 sec.) with $\approx 1.33\,\text{cm/s}$	Similar to Positive/Negative Ramp Plateau, but ending with a jump back to the actual value.

Eight anomalies from four anomaly types (Table 6.1), each in the x and z position (Fig. 6.2), are used to evaluate the different checks (see Table A.9). These anomalies have been chosen for an interpretable comparison. The anomaly scenarios are derived from the signal anomaly types examined by Weber [Web19] and based on ISO 26262-5:2018 (Table A.9). Furthermore, the speed of ramps and the amount of deviation are aligned with plausible movement speed and position range of the CoG during a motion of the back joint (Fig. 4.6).

The patient's weight is a representative parameter for the anthropometric data, as it has a significant influence on the nonlinearities that are not modeled in the dynamic model (Chapter 3.3.2). Based on the Body Mass Index (BMI) distribution in Germany, the categories described in Table 6.2 represent the German population applied to the standard patient from IEC60601 (Fig. A.7).

BMI Range	Description
BMI $<$ 25	Approximately 47.3% of the German population falls into this category, with an illustrative weight of around 67 kg at a BMI of 18.5.
25 $<$ BMI $<$ 30	The average BMI for Germans is around 26, constituting approximately 35.9% of the population, equivalent to an average weight of approximately 94 kg.
BMI $\geq$ 30	About 16.8% of Germans fall into this category, with an indicative weight of approximately 126 kg at a BMI of 35.
BMI ca. 70	Representing an extreme case weighing 250 kg. This weight serves as a comparison point, considering it as the maximum patient load for a middle-class Operating Room Table (OR table) (e.g., Getinge Maquet Meera [Get23a]).

Table 6.2: Distribution of BMI categories in the German population applied to standard patient (Fig. A.7)

As the target is also to be able to evaluate scenarios with patients outside the norm, BMI cases $\geq$30 are more of a concern, as they would result in more false positives with traditional approaches (Chapter 3.3). With weights of 250 kg, it

is expected that the deviations resulting from nonlinearities and deformation are larger compared to the case of 126kg (BMI $\geq$30), but the procedure and the results are similar, and, therefore, not explicitly simulated and evaluated.

6.1 Unscented Kalman Filter (UKF) Evaluation

Applying the UKF (Chapter 2.3.1) for CoG estimation using measurements of a load recognition system (Chapter 5.2, Fig. 5.7) demonstrates effectiveness, particularly during motion, where it excels in smoothing measurement noise (Fig. 6.1). Furthermore, challenges arise during static phases, manifesting as drift and an overshoot in position signals when movement stops, which necessitates further attention to improve the filter's accuracy. Additionally, the overshoot is attributed to the omission of acceleration/deceleration modeling (Chapter 4.3.3) and the impact of patient body and OR table deformations. As a metric for evaluation, the absolute difference of the estimated CoG to the measured one is calculated for each time step (Fig. 6.2).

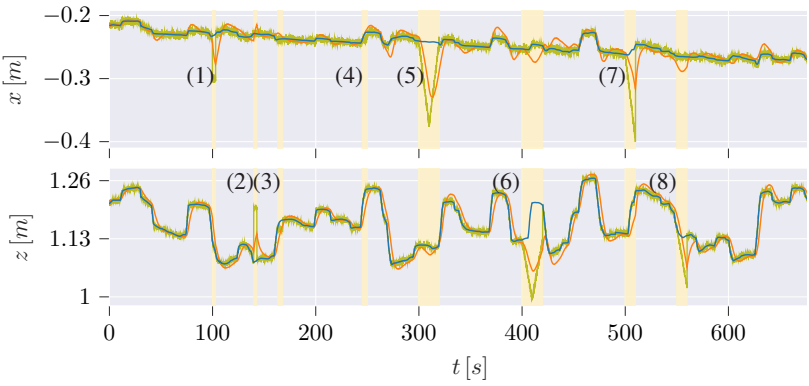


Figure 6.1: CoG estimation (*orange*) of the partial body movement UKF compared to measurements (*green*) and ground truth (*blue*) with anomalies (background marked *yellow*, Table 6.1)

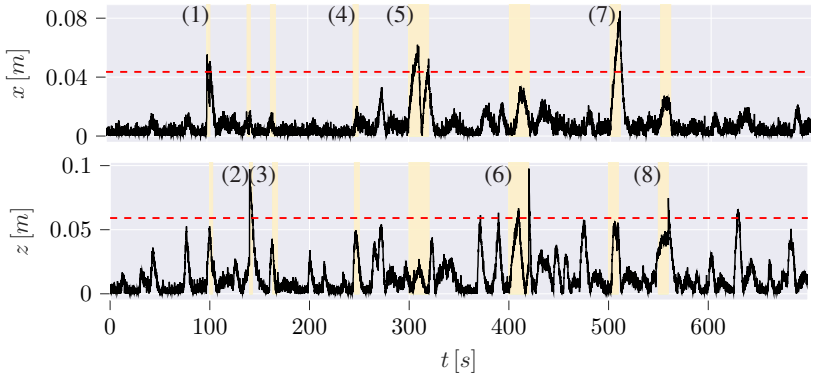


Figure 6.2: Absolute error of the estimated CoG by the UKF to the measurement data

The recovery time after, e.g., jump anomalies increases the FPR, but each time correlates to actual anomalies. That effect is beneficial if evaluated based on anomaly events rather than by each time step (Chapter 6.8). Especially in the z-direction, the absolute error between the measurement and the estimation does not distinguish between actual anomalies and deviations resulting from measurement errors. Furthermore, it can be stated for the examined scenarios that any anomaly will significantly increase the differential error. Moreover, the plateau x/z anomalies (3 & 4) do not lead to crossing the threshold, and in three non-anomaly-related events, the absolute estimation error crosses the threshold.

6.2 Extended Kalman Filter (EKF) Evaluation

The Extended Kalman Filter (EKF) (Chapter 2.3.1) estimation yields similar results (Figs. 6.3 & 6.4) as the UKF, especially since the model's nonlinearities are not significant (Chapter 2.3.1, Table 2.1). Consequently, EKF becomes a viable choice for resource-constrained embedded applications in specific scenarios to minimize resource consumption. If additional nonlinear factors be incorporated into the model, a reassessment of this choice is necessary, including a comparison with Kalman Filters (KFs) beyond both EKF and UKF.

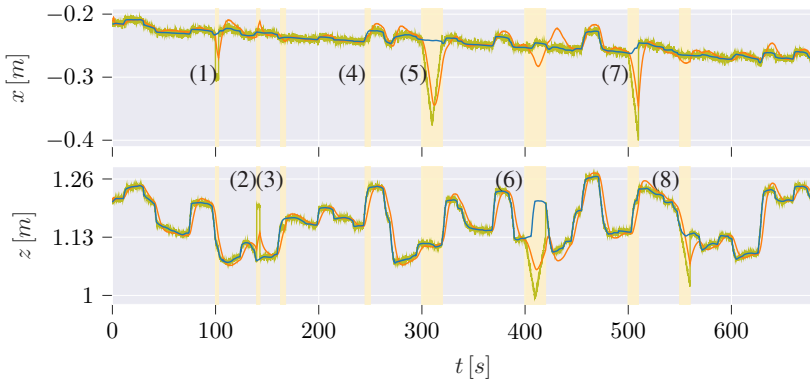


Figure 6.3: CoG Estimation of the EKF with anomalies (background marked *yellow*, Table 6.1)

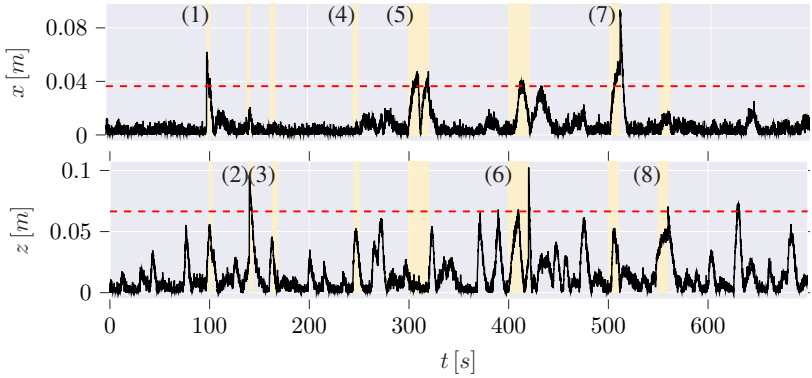


Figure 6.4: Absolute error of the estimated CoG by the EKF to the measurement data

The absolute error of the EKF (Fig. 6.4) shows significant deviations correlating with anomalies. Three events cross the threshold, and the plateau x/z anomalies (3 & 4) do not lead to crossing the threshold either. It can be stated here, too, that any anomaly will significantly increase the differential error for the examined scenarios.

6.3 Dynamic Model Comparison

For the UKF and EKF, both dynamic checks show a similar 99th percentile, so the deviations in evaluating normal behavior do not differ significantly (Table A.28). Only the absolute errors for the anomaly (8) (Figs. 6.5 & 6.6) deviate between both plots as the UKF has a more significant error in the x direction, although the anomaly is on the z-axis. This indicates that the UKF better captures the correlation between the two features. In addition, the z-axis 99th percentile of the EKF is 12% higher, and the x-axis 99th percentile is nearly 16% lower than those of the UKF, which indicates that the behavior of the UKF during regular system behavior is more accurate.

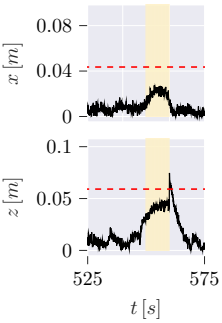


Figure 6.5: Anomaly 8 UKF (Fig. 6.2)

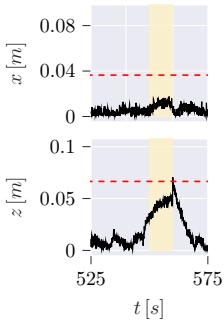


Figure 6.6: Anomaly 8 EKF (Fig. 6.4)

While the EKF correctly classifies 18 data points more as an anomaly than the UKF, it has five more false positives (Table 6.3). Hence, the differences between the models are not significant. Thus, when using KFs for anomaly detection, selecting the most appropriate filter depends more on the chosen parameterization and model properties than on the type of KF for the scenario examined here. The effectiveness of anomaly detection depends on adequate consideration of these factors, which underlines the need for a customized approach adapted to the specific characteristics and intricacies of the system under investigation.

Since the implemented UKF and EKF accuracies are similar, the evaluation for anomaly detection is done only for UKF, which involves the hybrid checks, too (Chapter 3.3.2). The choice of UKF over EKF is mainly motivated by the FPR,

which is 0.1% lower (FPR = 1.0%) than the FPR of the EKF (FPR = 1.1%). Based on their similar performance, an educated statement on which combination cannot be made since differences can only be due to statistical variance or slightly different parameter influences. No additional insight is expected from examining these two filters, as they are not optimized either.

Table 6.3: Confusion matrices (Appendix A.5.1) of dynamic checks in comparison (best-performing variant marked in bold)

Model	**UKF**		**EKF**	
	Predicted Label			
True Label	False	True	False	True
False	**6169**	**65**	6164	70
TNR / FPR	99.0%	1.0%	98.9%	1.1%
True	594	172	**576**	**190**
FNR / TPR	77.5%	22.5 %	75.2%	24.8%

6.4 Long Short-Term Memory (LSTM) Network Evaluation

The pure LSTM (Chapter 2.3.2 & Appendix A.5.2) network has emerged as a viable deep learning alternative to KFs in predicting a system's subsequent state. Thus, all position and velocity measurements are used as input tensors. Furthermore, the LSTM does have a disadvantage compared to the UKF as it is not iteratively updated based on all previous measurements. Therefore, an adequate time window that has to be exploratively determined is needed, which is why a time window of 30 values, corresponding to 3 seconds, is chosen. When the window size is increased, the resources required to execute the model also increase.

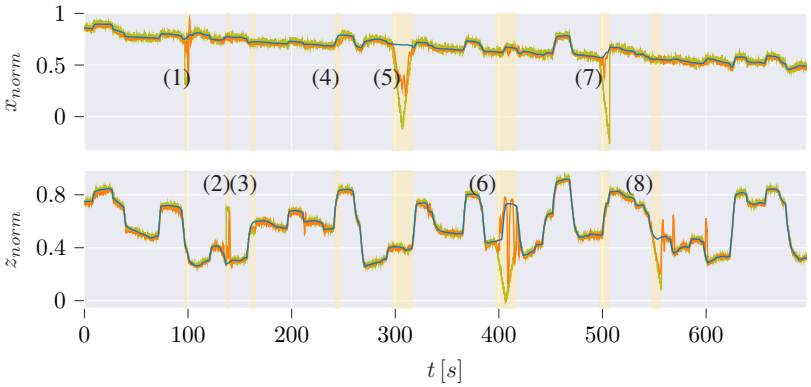


Figure 6.7: Normalized prediction of LSTM network (measured: green, estimated: orange, ground truth: blue) with anomalies (background marked *yellow*, Table 6.1)

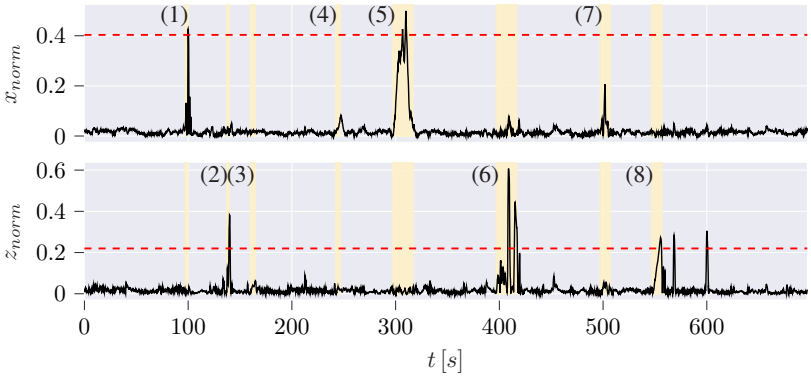


Figure 6.8: Absolute normalized difference between prediction of the LSTM network to measurements

The cost of increased resource consumption (Chapter 2.2.4) is rewarded with significantly improved FPR and False Negative Rate (FNR) (Table 6.4) compared to the dynamic checks (Chapter 6.3). Here, one event passes the threshold that is not correlated with an anomaly. Furthermore, the LSTM is trained to project the position and joint velocity measurements to the ground truth of position and thus to predict these (Fig. 6.7). In practice, it is not guaranteed to have ground truth data for training (Chapter 6.9).

Table 6.4: Confusion matrices (Appendix A.5.1) of LSTMs in comparison (best-performing variant marked in bold)

Model	**LSTM**		**Hybrid LSTM**	
	Predicted Label			
True Label	False	True	False	True
False	6167	37	**6213**	**1**
TNR / FPR	99.4%	0.6%	99.98%	0.02%
True	674	92	**571**	**195**
FNR / TPR	88.0%	12.0 %	74.5%	25.5%

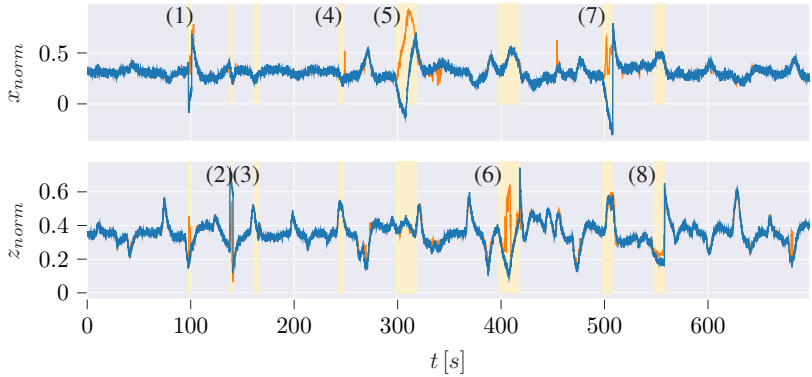


Figure 6.9: UKF error ($\hat{\mathbf{x}}_{\mathbf{UKF}} - \mathbf{y}$, Table A.3) predicted by the LSTM (prediction to measurement: orange, prediction to ground truth: blue) with anomalies (background marked *yellow*, Table 6.1)

The hybrid LSTM predicts the estimated deviation of the following UKF iteration based on a time window of 20 values corresponding to 2 seconds, which is a reduction of 33% compared to the pure variant. When reducing the window size, the FPR and non-anomaly-related peaks are reduced, while the FNR also decreases (Table A.26, Fig. A.56). The hybrid LSTM predicts the difference of the UKF estimation to the measurement directly (Figs. 6.9 & 6.10). Subsequently, the absolute difference between the prediction and measurement is used to determine anomalies by comparing it to the 99th percentile.

The up-sampling for the windows for the hybrid LSTM is based on the difference between the measured and the estimated x value. As this information is not available for the pure LSTM, the angular velocity of the back joint $\vec{\omega}_{jb}$ (Table 4.5) is used for upsampling since deviations are expected during movement. In addition, more extended window sizes >2 seconds lead to an increase of false positives.

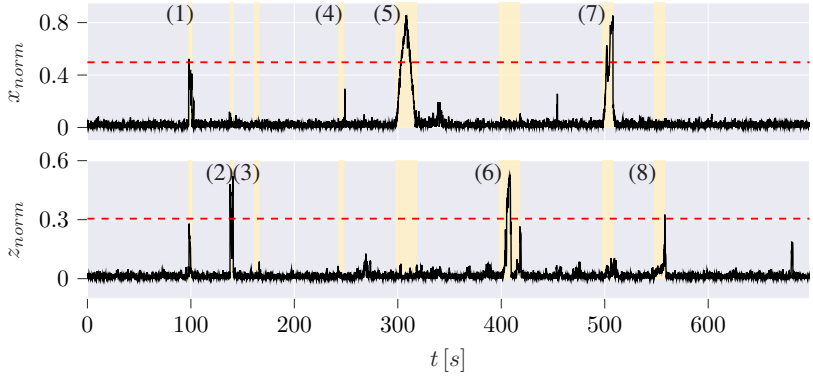


Figure 6.10: Absolute difference of hybrid LSTM prediction to measurement (background marked *yellow*, Table 6.1)

The hybrid LSTM results in better outcomes in terms of FPR, as it decreases the FPR of the pure LSTM by a factor of 30 (from ∼0.6% to ∼0.02%). At the same time, it decreases the FNR by ∼13.5%, leading to improved precision and specificity. Furthermore, both LSTM variants exhibit a threshold of ∼0.4 to ∼0.5

for x_{norm} and ∼0.2 to ∼0.25 for z_{norm} based on the 99th percentile (Table A.27) compared to the average non-anomaly data or peaks in the non-anomaly data (Figs. 6.8 & 6.9).

Comparing both LSTM variants to the UKF, the FPR can be improved, while the pure LSTM nearly halves the FPR (1.1% to 0.6%). In comparison to the hybrid variant, the pure LSTM has a ∼10.5% increased FNR (77.5%, Table 6.3). In addition, the hybrid LSTM outperforms the UKF and LSTM in terms of errors during normal behavior. Two non-anomaly-related peaks cross the threshold in the z-axis, while none in the hybrid variant exists.

6.5 Isolation Forest (IF) Evaluation

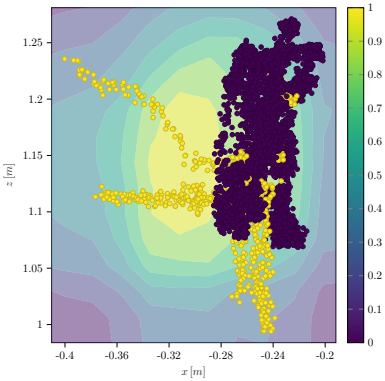


Figure 6.11: IF classification for position measurement x and z as inputs (normal: black, real anomaly: yellow) depending on the average path length

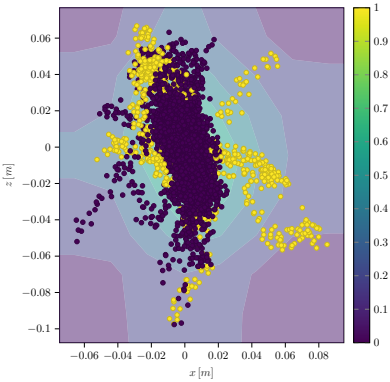


Figure 6.12: Hybrid IF classification of difference between UKF estimation to measurement of positions (normal: black, real anomaly: yellow) depending on the average path length

The IF (Chapter 2.3.2 & Appendix A.5.2) is trained on the position measurement data, while *no* window is used for evaluation. In this comparison, the IF is considered a “low-end” variant for the minimum achievable performance of learning

checks and profits from the reduced resource consumption (Chapter 2.3.2). Nevertheless, the pure IF is not capable of determining meaningful thresholds (Fig. 6.11), leading to an FPR of ∼40% and an FNR of ∼55.5%. Approximately half of the anomaly data points are detected; therefore, the number of false positives is 2488, which is unacceptable for medical applications (Table 6.5).

Table 6.5: Confusion matrices (Appendix A.5.1) of IFs in comparison (best-performing variant marked in bold)

Model	**IF**		**Hybrid IF**		**Hybrid IF Filtered**	
	Predicted Label					
True Label	False	True	False	True	False	True
False	3746	2488	**5424**	**810**	5273	961
TNR / FPR	60.1%	39.9%	87%	13%	84.6%	15.4%
True	425	341	**144**	**622**	141	625
FNR / TPR	55.5%	44.5 %	18.8%	81.2%	18.4%	81.6%

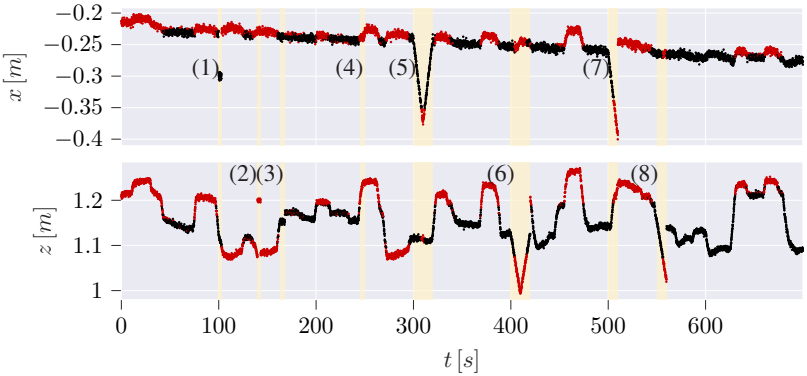


Figure 6.13: Anomalies detected by pure IF (red), shown in time series of position measurements with actual anomalies (background marked *yellow*, Table 6.1)

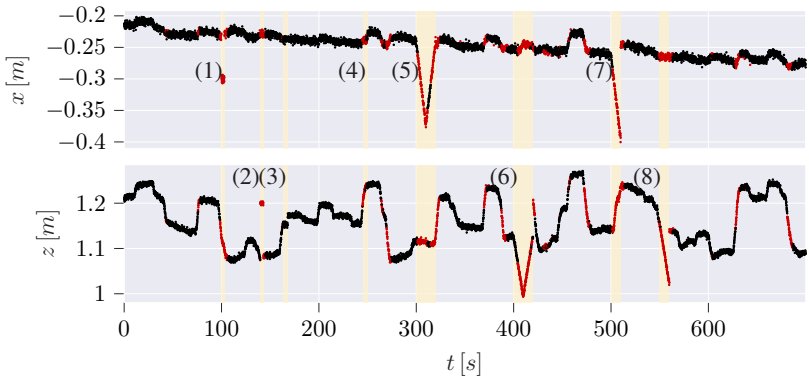


Figure 6.14: Anomalies detected by hybrid IF (red), shown in time series of position measurements with actual anomalies (background marked *yellow*, Table 6.1)

The hybrid approach uses a two-dimensional input based on the difference between the measurement and estimation in the x and z CoG direction. The normal data is centered in between thresholds, resulting in a significant reduction in both FPR (∼39.9% to ∼ 13%) and FNR (∼55.5% to ∼18.8%) (Table 6.5) compared to the pure IF. The hybrid IF surpasses the pure IF and the pure UKF in FNR, finding twice as many anomalies as the IF and four times more than the UKF. As a result, it has a three times lower FNR than the pure model. It is twofold in terms of false positives as it has a third of the false positives of the pure IF but ∼13 times more than the UKF (1% to 13%). Therefore, the hybrid IF improves the anomaly detection regarding the FNR (Table 6.5) *but not for improving the FPR*.

Furthermore, there is an effect to the hybrid approach regarding anomaly classification, which is revealed here. Because the classification relies on manipulating measurement data without considering that the dynamic model, in this case, the UKF, may need time to recover from abrupt changes in measurements back to normal states (Chapter 6.1). In jump scenarios, where the dynamic model experiences a short-term delay in returning to normality, the anomalies are not classified as such, negatively impacting the FNR across all examined models. In comparison to the hybrid variant (Fig. 6.14), it is noticeable in the time series of the pure IF (Fig. 6.13) that most of the detected anomalies do not correlate with anomalies.

Table 6.6: Confusion matrix (Appendix A.5.1) of pure and hybrid IF for position without measurement noise (best-performing variant marked in bold)

Model	**IF**		**Hybrid IF**	
	Predicted Label			
True Label	False	True	False	True
False	4083	2151	**5273**	**961**
TNR / FPR	65.5%	34.5%	84.6%	15.4%
True	580	186	**141**	**625**
FNR / TPR	75.7%	24.3 %	18.4%	81.6%

When reducing the measurement noise for the pure IF, the results cannot be improved but worsened (Table 6.6), leading to an FPR of $\sim$ 34.5% and an FNR of $\sim$ 75.7%. The result does not improve further, even for the hybrid variant, when efficient data filtering removes the noise from the measurement.

6.6 Autoencoder (AE) Evaluation

As each window is reconstructed entirely instead of a single value compared to LSTM or IF, a complete window must be evaluated for the pure and hybrid variants at each time step. Therefore, the AE (Chapter 2.3.2 & Appendix A.5.2) threshold evaluation is based on the reconstruction error for each axis. It is determined as the Mean Absolute Error (MAE), which calculates the average of all reconstructed values to the measurement for one window:

$$MAE_{window} = \frac{1}{n} \sum_{i=1}^{n} |\hat{\mathbf{x}}_i - \mathbf{x}_i| \tag{6.1}$$

With the number of data points n in the window, the reconstructed value $\hat{\mathbf{x}}_i$ for the i-th data point and the measured value $\mathbf{x}_i$ for the i-th data point.

Table 6.7: Confusion matrices (Appendix A.5.1) of AEs in comparison (best-performing variant marked in bold)

Model	**AE**		**Ia) Hybrid AE**		**Ib) Hybrid AE**	
			Predicted Label			
True Label	False	True	False	True	False	True
False	6064	141	6040	165	**6107**	**124**
TNR / FPR	97.7%	2.3%	97.3%	2.7%	98%	2%
True	502	264	673	93	**469**	**297**
FNR / TPR	65.5%	34.5 %	87.9%	12.1%	61.2%	38.8%

Based on this metric, the 99th percentile can be generated as a threshold performance evaluation (Table 6.7). The AE uses the CoG position as input with a time window of 30 (3 seconds) and does show significant errors in the reconstruction during anomalies (Fig. 6.15). Furthermore, the reconstruction error during plateau anomalies does not show a significant increase either at the z-value or the x-value. Two events cross the threshold without an anomaly as the trigger. Also, a drift anomaly in the x direction (5) remains barely detectable.

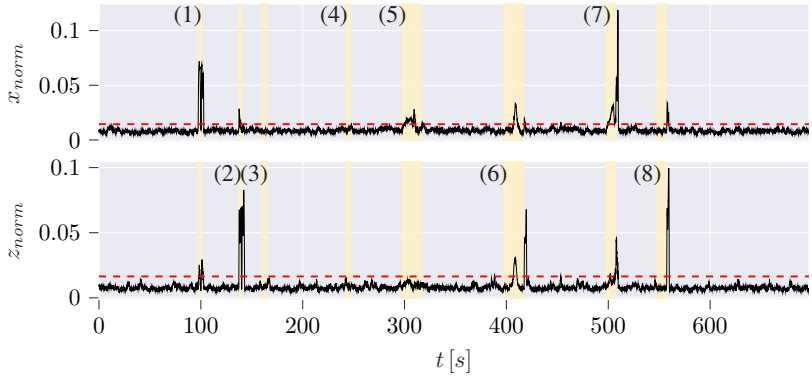


Figure 6.15: AE window reconstruction MAE per feature with anomalies (background marked *yellow*, Table 6.1)

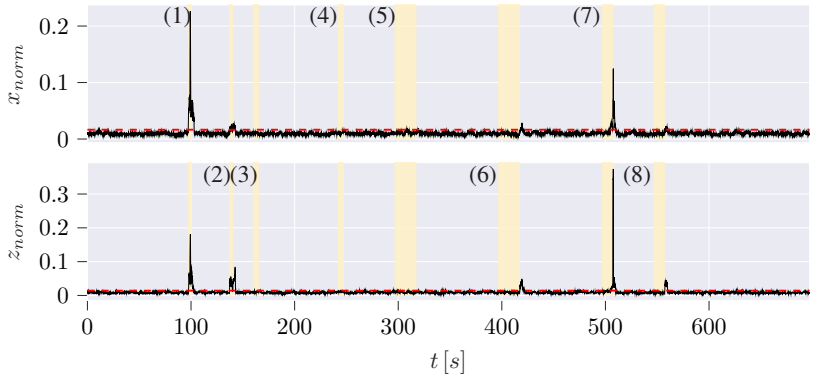


Figure 6.16: Hybrid AE Ia window reconstruction MAE per feature with anomalies (background marked *yellow*, Table 6.1)

For the first hybrid AE variant (Ia), a time window of 30 (3 seconds) with two input features is chosen as well (Fig. 6.16). Reducing window size induces other issues as the reconstruction error is noisier (Table 6.8, & Fig. A.54). Similar to the pure AE, the MAE of a window between the reconstructed signal to the difference between estimation and measurement is built for each of the Patient's CoG x and z position (Fig. 6.16). The input features are chosen as the difference between the estimated UKF value and the measurement. Although this is similar to the LSTM (Chapter 6.4), the AE cannot be trained to a desired value like the ground truth but only to the input tensor, and therefore, the noise has a negative impact.

A smaller window size increases the reconstruction error during anomalies but increases the overall noise, which makes it harder to identify anomalies (Table 6.8, & Fig. A.55). The reconstruction error for plateau anomalies shows no significant increase as with the pure AE. In addition, a drift anomaly in the x-direction (5) remains barely recognizable. Overall, the reconstruction error is lower for normal data points. Still, the anomalies found are almost a third less, and there are 24 more false positives than detected by the pure AE (Table 6.7).

While the AE has an increased FPR (2.3%) and FNR (65.5%) at smaller window sizes, a different approach for the hybrid AE shows better performance than the difference between dynamic model estimation and measurement (FPR: 2.7%;

Table 6.8: Confusion matrices of AEs in comparison for window size 10 (best-performing variant marked in bold)

Model	**AE**		**Ia) Hybrid AE**	
	Predicted Label			
True Label	**False**	**True**	**False**	**True**
False	**6133**	**92**	6092	133
TNR / FPR	98.5%	1.5%	97.9%	2.1%
True	**624**	**142**	669	97
FNR / TPR	81.5%	18.5%	87.3%	12.7%

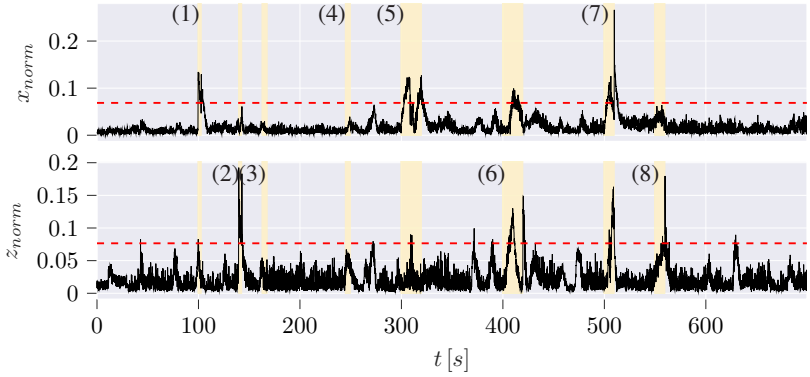


Figure 6.17: Hybrid AE Ib MAE of measurements with window size 4 reconstructing UKF estimations and measurements with anomalies (background marked *yellow*, Table 6.1)

FNR: 87.9%): Using the measurements and estimates as separate values, a time window of 4 (0.4 seconds) leads to the improvement of FPR and FNR, while increasing the window size worsens the results. In addition, the latent space (Appendix A.5.2) can be reduced from 8 to 2 while improving performance. This approach (Fig. 6.17) shows that the training effort and the window size can be reduced. Moreover, the first hybrid AE (Ia), similar to the pure one, shows that the FPR and FNR cannot be reduced for the investigated scenario. This makes the second variant (Ib) more attractive for multidimensional tasks. Although it increases the FPR by 1%, it decreases the FNR by 16.3% compared to UKF.

The threshold is crossed in the z-axis five times without being correlated to an anomaly. Similar to the LSTM (Chapter 6.4), the up-sampling for the windows is based only on the x value, so the normal data in the z-direction could have been negatively impacted as the training data then does not adequately represent the actual normal behavior in the z-axis.

6.7 Discussion on Anomaly Detection Algorithms

By comparison between the pure data-based models (Table 6.9), the LSTM has the lowest FPR and the IF the lowest FNR. While the LSTM can improve the current state estimation (Chapter 2.3.1 & [Koc23]), algorithms like the AE or the IF classify the data into normal or anomaly data, respectively, creating a reconstruction/anomaly score. Although the pure IF performs best for false negative and true negative data, the difference between correctly and incorrectly classified cases is so insignificant that the data point has only about a 58.4% chance of being correctly classified (accuracy). Therefore, the practical relevance of this approach is questionable. Furthermore, the pure LSTM network performs best for true negatives and false positives but is closely followed by the UKF.

Table 6.9: Confusion matrices (Appendix A.5.1) of pure learning checks and dynamic checks in comparison (best-performing variant marked in bold)

Model	**AE**		**IF**		**LSTM**		**UKF**	
	Predicted Label							
True Label	False	True	False	True	False	True	False	True
False	6064	141	3746	2488	**6167**	**37**	6169	65
TNR / FPR	97.7%	2.3%	60.1%	39.9%	99.4%	0.6%	99.0%	1.0%
True	502	264	**425**	**341**	674	92	594	172
FNR / TPR	65.5%	34.5 %	55.5%	44.5%	88.0%	12%	77.5%	22.5%

Generally, the time window and input features for the AE (Ia & Ib) and LSTM (III) hybrid checks are considerably smaller. When comparing the confusion matrices of different hybrid checks (Table 6.10), the hybrid LSTM (III) network performs best when it comes to true negatives and false positives. On the other hand, the hybrid IF (II) performs best in the case of false negatives and true negatives and is the best choice if a balanced FPR and FNR are desired with focus on recall and F1 Score. Since FPR is prioritized over FNR, the hybrid IF (II) is unsuitable in this form, although it has the best overall performance. Nonetheless, this lightweight approach is promising, particularly in resource-constrained environments. The hybrid AEs (Ia & Ib) offer a solution that lies between the performance of the other two, with the second hybrid variant (Ib) showing more promising results. Furthermore, only the LSTM variants (pure & hybrid III) surpass the pure dynamic check implemented as UKF in terms of FPR, but all learning and hybrid checks except for the pure LSTM surpass the UKF dynamic check in terms of FNR and recall. An increase in the window size is ineffective as the LSTM performance even decreases.

Table 6.10: Confusion matrices (Appendix A.5.1) of hybrid checks using the UKF (Chapter 6.1) in comparison (best-performing variant marked in bold)

Hybrid Model with UKF	**Ia) & Ib) AE**		**II) IF**		**III) LSTM**	
	Predicted Label					
True Label	False	True	False	True	False	True
False	6040/6107	165/124	5424	810	**6213**	**1**
TNR / FPR	97.3/98.0%	2.7/2%	87%	13%	99.98%	0.02%
True	673/469	93/297	**144**	**622**	571	195
FNR / TPR	87.9/61.2%	12.1/38.8%	18.8%	81.2%	74.5%	25.5%

The hybrid IF (II) has the most balanced accuracy among all the algorithms (Table 6.11) regarding FNR, even though the IF is the data-based model requiring the least resources here. The hybrid checks (Ia, Ib, II & III, Table 6.11) have proven efficient in identifying anomalies, but the dynamic checks affect their overall performance. The UKF filter performed worst in FNR except for the pure LSTM,

Table 6.11: Metrics (Appendix A.7) in comparison all anomaly scenario and normal behavior data (best-performing variant marked in bold)

Metric	AE		IF		LSTM		UKF
	Pure	Hybrid	Pure	Hybrid	Pure	Hybrid	
Accuracy	0.908	0.880/0.915	0.584	0.864	0.898	**0.918**	0.906
Precision	0.652	0.360/0.705	0.121	0.434	0.713	**0.995**	0.726
Recall	0.344	0.121/0.388	0.445	**0.812**	0.120	0.255	0.225
F1 Score	0.451	0.182/0.500	0.190	**0.565**	0.205	0.405	0.343
ROC AUC Score	0.661	0.547/0.684	0.523	**0.841**	0.557	0.627	0.607
False Positive Rate	0.023	0.027/0.020	0.400	0.130	0.006	**0.0002**	0.010
False Negative Rate	0.655	0.879/0.612	0.555	**0.188**	0.880	0.745	0.775

suggesting that the underlying models and parameterization must be improved for a performance increase. Hence, the next step to improve the performance of the hybrid checks is to enhance the dynamic check.

Since KFs can estimate values from sensor fusion that are not measured, they can be used as another input for learning checks. On the one hand, this reduces the effort to train the machine learning algorithm and, on the other hand, help reduce the deviations in estimation with the hybrid check to minimize FPR. Furthermore, these estimated values can be used for static checks. For example, $\vec{r}_{ub}$ and $\vec{r}_{lb}$ (Chapter 4.3.3, Table 4.5) that are not measured directly can be used as features for hybrid checks. This can further reduce the needed window size: e.g., in combination with an LSTM (Fig. A.57), the window size is reduced to a single time step while decreasing the FNR to $\sim$ 57.4% but increasing the FPR to $\sim$ 0.2%.

Except for the hybrid IF (II), which is not evaluated based on the 99th percentile (Table A.27), all other models have a significant FNR. Thus, it can be assumed that the selected anomalies are challenging to detect due to minor deviations (e.g., <2cm for Anomaly (4), Fig. 6.1) in the measurement from the true values, which are additionally obscured by noise. Furthermore, the jump anomalies (1 & 2) are

detected by all models. In comparison, the plateau anomalies (3 & 4) are only detected in the z-direction (3) by the pure AE, the first hybrid AE variant (Ia), and the hybrid IF (II). Since not all anomalies are reliably detectable by all models (Chapter 2.3), an ideal anomaly detection system would include an ensemble of models (Chapter 2.3.3), e.g., requiring an anomaly to be detected in at least two checks. In addition, the used performance metrics have disadvantages when comparing the algorithms regarding significant anomalies due to misleading FNR since an algorithm must not detect all anomaly data points to detect the anomaly.

6.8 Relevance to the Operating Room (OR)

The types of anomalies considered are a cross-section of those specified in ISO 26262-5:2018 (Chapter A.9). As these types are combinations or special cases of each other, the examined anomalies can be considered representative. Furthermore, the value range of the deviation caused by the anomalies (Table 6.1) must be considered, especially when dealing with plateau anomalies. These are not detected here because they do not deviate significantly from the simulated measurement noise. As a result, they cannot be clearly distinguished from the system's normal behavior. In the context of the Operating Room (OR), this is an advantage rather than a disadvantage since anomalies that do not deviate significantly - e.g., less than the 99th percentile - from the system's normal behavior should be neglected in any case to avoid further disruption to the surgery. Therefore, an essential feature of anomaly detection is not only to decrease the FPR but also to ignore anomalies that have an insignificant impact on the patient's safety so that the availability of the medical device is maintained.

In the OR context, the hybrid LSTM (III) is practically relevant, as the false alarm rate is such that only one data point is a false positive. Furthermore, this false positive is still correlated with the anomaly (8) and can therefore be neglected. In this case, no false alarm would interfere with the surgical procedure. All other variants will detect at least one anomaly resulting from normal behavior. As the time frame examined is 700s, this would imply one false alarm for every 700s

of OR table movement. As this time frame consists of 32 up and downwards movements of the patient's upper body (Fig. 6.1), one false positive means a false alarm every 32 movements. Since a state-of-the-art OR table such as Maquet Corin is allowed to move for 2 minutes until it has to cool down again for 8 minutes [Get24], this extrapolates to nearly one hour of surgery in the worst case, while adhering to the instructions for use. Furthermore, based on expert knowledge, the beach chair position (Fig. 4.6) considered in the scenario here corresponds to 10% of the overall surgeries, which are estimated to be 13000 in a mobile OR table's life cycle, in which the back joint is moved 1300 times in total, meaning once per surgery in beach chair position. In that case, a false positive every 32 movements implies a false alarm every 32 surgeries in beach chair position. Including all patient positions, a mobile OR table's overall amount of back joint movement (Chapter 2.6, Fig. 2.24) during its life cycle is estimated to be ∼5000, whereby one movement is considered a cycle of upwards (>90°) and downwards (<-90°) movement resulting in a total angle of >360°.

For the hybrid LSTM (III), *no* false alarms are expected from the data examined. Even in the other cases, the current estimate of at least 70% false alarms caused by medical devices (Chapter 2.3) is undershot. The LSTM network, due to its memory cells, can model the internal states of a system, such as its velocity over multiple time stamps, unlike other data-based models (Appendix A.5.2). This property is valuable when monitoring the time-dependent behavior of dynamic systems and an advantage in hybrid checks. Data-based models with similar properties that have not been investigated may have a similar or better FPR/FNR in combination with a dynamic model. Therefore, the LSTM network may be a suitable choice in the scenarios investigated, but may not be the best choice, and hybrid checks may even achieve better results. This aligns with the "no free lunch theorem"[1], as it cannot be stated beforehand which algorithm performs best in a hybrid check.

[1] The No-Free Lunch (NFL) theorem is a fundamental principle in optimization and machine learning. It asserts that no single method or algorithm can consistently outperform all others across all types of problems. In practical terms, this means that an algorithm's effectiveness depends on the specific problem to which it is applied, and there is no universal "best" approach [WM96].

As the previous metrics (Table 6.9) show an application-independent evaluation of the performance of the different algorithms, Tables 6.12 and 6.13 consider the context of an OR, where only one event is decisive and not every step. The pure IF is not considered functional and is therefore not taken into account in the comparison. An event here is either the occurrence of a true anomaly or the detection of an anomaly (false or true). Therefore, true negatives are not considered here, as they cannot be meaningfully defined as an event, so the metrics for this evaluation cannot be calculated.

Table 6.12: Confusion matrices (Appendix A.5.1) of pure learning checks and dynamic checks in comparison for event-based evaluation (best-performing variant marked in bold)

Model	**AE**		**LSTM**		**UKF**	
	Predicted Label					
True Label	False	True	False	True	False	True
False	/	6	/	**2**	/	3
True	**1**	**7**	2	6	2	6

Table 6.13: Confusion matrices (Appendix A.5.1) of pure learning checks and dynamic checks in comparison for event-based evaluation (best-performing variant marked in bold)

Hybrid Model	**Ib) AE**		**II) IF**		**III) LSTM**	
	Predicted Label					
True Label	False	True	False	True	False	True
False	/	5	/	9	/	**0**
True	2	6	**1**	**7**	2	6

For pure learning checks without a dynamic model, it is unlikely that sufficient data can be collected to achieve the desired performance. Thus, hybrid checks, with their reduced amount of inputs compared to learning checks, are likely to outperform these even further when the dimensions of the system increase. As the movement scenario in beach chair position corresponds to an estimated 10% of all surgeries with a mobile OR table, it is a subset of the capabilities of the OR table (Chapter 2.6) and the possibilities of patient positioning (Chapter 4.3.3) and anthropometry (Chapter 3.3.2). Therefore, the training data collected via simulation of $\sim$5 hours ($\sim$850 movements) of driving the back joint in a static scenario is enough to yield results on the performance of the examined algorithms. Data collection will be a key challenge in the OR (Chapter 2.3.3), even if the issue of data protection is disregarded. While in the automotive industry, manufacturers reach out to the data of millions of cars increasing year by year, in the medical device industry, such as for OR table, manufacturers will have only the data available of 1000 to 2000 each year - and not all of these OR table is used for all patient types, positions, and surgical procedures [PVR+22]. In addition, more features will result in less distinguishable anomalies (Chapter 2.3.2) due to their distance in multidimensional space. Therefore, reducing input features and the window size of the hybrid checks are likely to outperform pure learning checks (Chapter 6.7) here as well.

Hybrid checks with low calculation effort (e.g., EKF with IF) can be used locally with a low threshold, leading to an increase in the FPR. When running a dynamic check on a state-of-the-art OR table, calculation times of <10ms are achieved (Appendix A.12.4) with a loss of accuracy. By leveraging the calculation performance provided, e.g., by the clinical Information Technology (IT) systems or an OR integration system (Chapter 2.7) for a hybrid check, a calculation time of <2ms is achievable as one prediction with the LSTM of the hybrid check is executed on a conventional workstation laptop[2] within $\sim$1.62ms with batch size of one. Latencies due to local networks need to be considered. In the case of the design

[2] The system used here is a Dell Precision 7740 [Del20] with Intel(R) Core(TM) i7-9850H CPU 2.60GHz 2.59 GHz and AMD Radeon (TM) Pro WX 3200 Graphics.

of the hybrid LSTM (III), three values (CoG x, y & z) need to be transmitted every 100 ms. If they have a resolution of a 64-bit float, that means a necessary bandwidth of 1.92 kbit/s, which falls under the required data throughput of control data realizable with classic Controller Area Network (CAN) (Table A.21). Thus, when using, e.g., Ethernet, latencies of <1s are feasible, which facilitates a direct notification of an incident within <2s if the response needs to be sent over the network as well.

When integrating anomaly detection in ORs, the hybrid LSTM (III) should be selected so that the surgical procedures are not disturbed. In this way, an alarm is likely to be a safety or security event since the accuracy is close to 100% with at least a 99th percentile as a threshold (Table 6.11). If an alarm is raised, clinical staff must decide whether to shut down the network in the OR and go into a manual, disconnected mode to prevent potential patient harm. This includes shutting down tasks that perform automated movements, e.g., with angiography systems. A reaction time of 2s makes technical measures, e.g., to stop motions to prevent collisions, possible. Since the movement speeds of OR tables are <2cm/s for prismatic and <6.5°/s for revolute joints (Chapter 2.4.2), severe patient harm is preventable within that time frame. But as incidence are reported that the prevention of motions within a Hybrid Operating Room (HOR) due to technical measures has led to severe patient harm, the expertise of the clinical staff, especially in an emergency, must not be ignored. Furthermore, detecting security events will make it necessary to prepare and train the clinical staff to handle these situations during surgery.

If the local IT infrastructure for direct measures should not be used, suspicious events can be extracted, which all the examined hybrid checks are capable of, and sent to a backend system with the “context” (Chapters 3.3.3 & 4.6) data for further investigation in the backend. In a further step outside the clinical context (e.g., Security Information and Event Management (SIEM), Chapter 3.4), where a false alarm does not threaten patient safety, a less conservative approach with a lower percentile than the 99th percentile as a threshold allows the manufacturer to analyze the detected events and improve the system’s security through protection with new updates.

6.9 Hybrid Anomaly Detection with Real Data

Given that the hybrid LSTM demonstrated the best performance among the variants examined (Chapter 6.7), it is employed on real system data from Getinge Corin [Get23c], collected via internal CAN communication. The setup is configured as in the simulated scenario, with the patient approximated using sandbags distributed along the tabletop in accordance with IEC60601 (Fig. A.7). Additionally, the same anomalies (Table 6.1) are applied to a scenario involving the movement of the back joint.

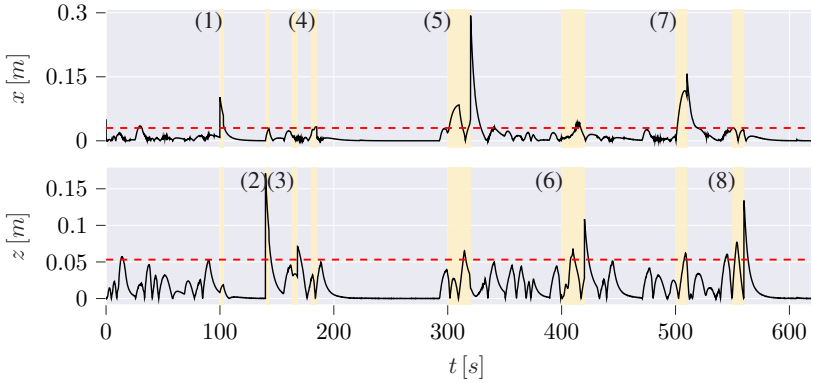


Figure 6.18: Absolute error of the estimated CoG by the UKF (Fig. A.58) to the measurement using real data with anomalies (background marked *yellow*, Table 6.1)

The UKF estimation (Figs. 6.18 & A.58) exhibits a FPR of 7.2% and a FNR of 42.2% (Table 6.14), while the pure LSTM (Figs. 6.19 & A.59) shows a FPR of 2.8% and a FNR of 47.3%. The hybrid LSTM (Figs. A.60 & 6.20) reduces the FNR compared to both, although it has a higher FPR than the pure dynamic or pure learning check variants. When applying an event-based evaluation (Table 6.15), the hybrid LSTM achieves ∼67% fewer false positives. In addition, the pure LSTM fails to detect the fourth and eighth anomalies, while both the hybrid LSTM and UKF successfully detect all the anomalies examined.

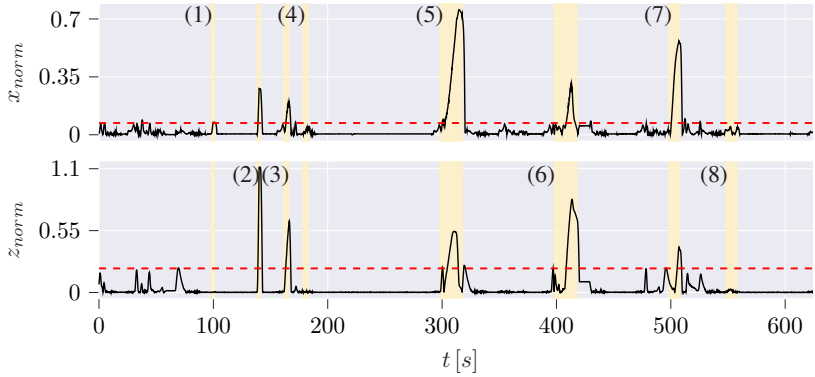


Figure 6.19: Absolute normalized difference between prediction of the LSTM (Fig. A.59) network to measurements using real data with anomalies (background marked *yellow*, Table 6.1)

Compared to the simulation data, the acquisition of data using the load recognition system (Chapter 5.7) in this test environment has limitations that negatively impact the performance and the evaluation results especially for the UKF and thus the hybrid LSTM. Yet, this requires an update of the actual design and implementation of the load recognition system, which is outside the scope of this dissertation:

Table 6.14: Confusion matrices (Appendix A.5.1) of LSTMs and UKF in comparison for real system data (best-performing variant marked in bold)

Model	**LSTM**		**Hybrid LSTM**		**UKF**	
			Predicted Label			
True Label	False	True	False	True	False	True
False	**5399**	**157**	5116	440	5173	403
TNR / FPR	97.2%	2.8%	92.1%	7.9%	92.8%	7.2%
True	362	404	**239**	**527**	323	443
FNR / TPR	47.3%	52.7%	31.2%	68.8%	42.2%	57.8%

1. Missing Ground Truth: The CAN traces consist of filtered measurements that do not represent the *ground truth* and lack measurement noise. Consequently, training must be performed using these CAN traces, thereby incorporating potential measurement errors and filter behavior into the training.

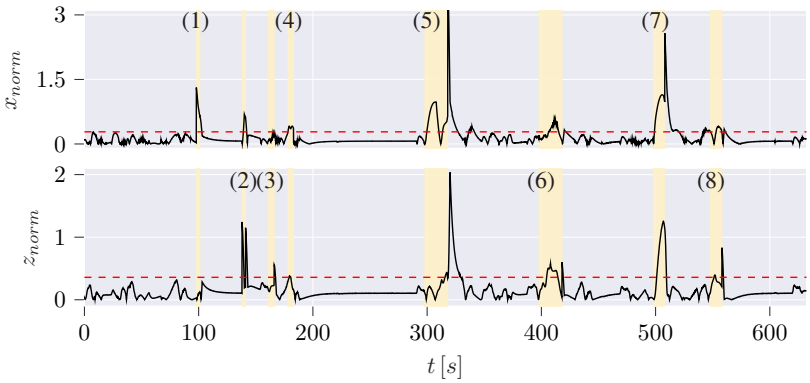


Figure 6.20: Absolute difference of hybrid LSTM prediction (Fig. A.60) to measurement using real data with anomalies (background marked *yellow*, Table 6.1)

2. Worst-Case Assumption of the z-axis: The z-axis of the CoG measurement is based on worst-case assumptions for tipping scenarios. Therefore, in the considered beach chair position, it will not fall below the theoretical horizontal patient position (Fig. A.58). Since this is not modeled in the dynamic check, it negatively affects the performance of the UKF and, consequently, the hybrid LSTM.

Table 6.15: Event-based confusion matrices (Appendix A.5.1) of LSTMs in comparison for real system data (best-performing variant marked in bold)

Model	**LSTM**		**Hybrid LSTM**		**UKF**	
			Predicted Label			
True Label	False	True	False	True	False	True
False	/	6	/	**2**	/	6
True	2	6	0	**8**	0	**8**

3. Sandbags as Patient: The patient is approximated using sandbags placed across the tabletop, but the deformability of the human body cannot be replicated with these. Additionally, to prevent them from falling, they must be tied tightly to the table, which further affects the approximation of the patient's deformability, making the scenario more rigid.

7 Conclusion and Future Work

7.1 Conclusion and Scientific Contribution

This dissertation contributes to a safe and secure medical device network in ORs by presenting a hybrid anomaly detection approach that targets the challenge of false alarm reduction. In addition, it covers both the System of Systems (SoS) of medical devices as a whole and a single medical device, for which an OR table is taken as an example (RQ 2, Chapter 1.3). Furthermore, this approach accommodates the necessary architectural changes, resulting in a mixed Electric/Electronic architecture (E/E architecture) that supports signal-based and service-oriented modules (RQ 3, Chapter 1.3). Together, the anomaly detection and the hybrid architecture fulfill the requirements (RQ 1, Chapter 1.3) of the future challenges of increasing interoperability, connectivity, and automation within the OR for an interoperable and modular OR table (Chapter 1.1). While solutions from the automotive industry served as inspiration, they cannot be applied to medical devices without further adaptation due to their special requirements and constraints (Chapter 3.2). In conclusion, the following results represent the scientific contribution of this dissertation:

1. Reference Requirement Specification
The requirements analysis (Chapter 3.2) regarding future trends and challenges reveals that architectural flexibility in the OR, especially for OR tables, is mandatory (RQ1). Traditional signal-based architectures and proprietary interfaces cannot provide this flexibility, as medical devices in the OR will operate in a networked environment. Moreover, flexibility is essential for modular medical devices such as OR tables as it is also required at the system/subsystem level (Chapter 3.2.3).

Thus, the derived functional and non-functional requirements (Chapter 3.2) result in a reference requirement specification for interoperable and modular OR tables as well as similar medical devices (Appendix A.10 & A.11). Additionally, the requirement specification resulting from analyzing demands for interoperability in the OR for the modular OR table gives a novel guidance for similar devices. Here, safety and security requirements from the automotive industry that resulted from connectivity challenges were leveraged. This procedure can serve as a blueprint for the future, as the medical device industry will not be the first to adopt new technological solutions due to reasonable safety and security concerns. Therefore, security measures can be implemented proactively to prevent patient harm.

2. Threat Analysis and Risk Assessment (TARA) for Medical Devices
Security processes should be adapted from other safety-related domains, such as the automotive industry. This has been demonstrated by the adaptation and application of HEAling Vulnerabilities to ENhance Software Security and Safety (HEAVENS) 2.0 Threat Analysis and Risk Assessment (TARA) (Chapter 3.1). Using proven security processes will accelerate securing medical devices in the OR (RQ2). In addition, adopting HEAVENS 2.0 as TARA is the first concrete proposal to fulfill the combination of risk analysis and threat modeling required by medical device standards (Chapter 3.1). Moreover, as HEAVENS 2.0 originated in the automotive industry, it is a more suitable process for medical devices based on embedded systems than the proposals of current medical device standards and guidelines that predominantly refer to IT processes.

3. Anomaly Detection
Anomaly detection will play a central role in the OR regarding security (RQ2). The novel anomaly detection approach (Chapter 3.3) uses a hybrid check consisting of a dynamic and a learning check and contributes to reducing the false alarm rates in the OR. As a proof of concept, it has been realized as a *position surveillance* for an OR table by monitoring the CoG of the patient. With this example, the assumption that a hybrid check can reduce FPR and FNR (Chapter 3.3.2, Fig. 4.8) could be verified in all examined cases (Chapter 6.7). Furthermore, the hybrid LSTM (III) is proposed as a practical solution to the false alarm rates in the OR (Chapters 6.8 & 6.9).

The inspiration also taken from the automotive field is the anomaly detection approach for physical signals based on the *Automotive Observer* (Chapter 2.3.3). Additionally, the solution pursued here is not subject to the resource constraints of embedded systems, as backend systems/external servers are included in the architecture with distributed anomaly detection (Chapter 3.4.3). Moreover, while including other devices with Service-oriented Device Connectivity (SDC), this is the first holistic proposal for an Intrusion Detection System (IDS) within an OR supervising multiple devices. In addition, by checking the plausibility of individual physical signals, safety is improved. Thus, interoperability should not only be considered an efficiency increase in the OR that bears the security challenge (Chapter 1). Furthermore, it significantly contributes to the safety of the patients.

Lastly, the hybrid checks are a domain-independent anomaly detection approach for a system to monitor, which can partially be described mathematically using expert knowledge as a dynamic check (Chapter 3.3.1). Therefore, the *position surveillance* for the OR table is a novel approach for checking the patient and OR table position plausibility, but the hybrid checks are not restricted to this application. Hybrid checks outperform dynamic and pure learning checks and are partially explainable due to the properties of the involved dynamic check. Explainability is mandatory for verification, validation, and the approval process, so it is considered the first step in introducing reliable machine learning algorithms in the OR. Considering OR applications (Chapter 6.8), false alarm rates of over 70% (Chapter 2.3) resulting from state-of-the-art solutions can be significantly reduced with hybrid checks to decrease the burden on the clinical staff.

Moreover, the execution environment of services is not confined to their designated initial locations, allowing for dynamic deployment that harnesses the capabilities of backend systems (Chapter 3.4.2). This dynamic service deployment enhances system reliability and availability and introduces inherent redundancy. The system adapts and seamlessly switches between services by utilizing dynamic service discovery with service orchestration, enhancing reliability and availability.

4. Mixed Software and E/E Architecture
The mixed software and E/E architecture OR table (Chapter 3.4) allows medical devices to be flexible while integrating legacy modules (RQ3). This will facilitate the gradual adaptation of medical devices to future challenges in interoperability, connectivity, and automation. In addition, the architectural approach is aligned with a proposal for an OR table SDC interface (Chapter 3.4.1), considering modularity.

Furthermore, the architecture is the fourth inspiration drawn from the automotive industry. As there is currently no medical architecture standard like AUTomotive Open System ARchitecture (AUTOSAR) to build upon, the proposal for the medical E/E architecture contributes to adapting current OR table architectures and the first proposal for a standardized medical device software and E/E architecture for robotic medical devices similar to OR tables (Chapter 3.4.2). Specifically, the proposed software and E/E architecture incorporates a standardized SDC interface.

5. Challenges of Medical Cyber-Physical Systems (MCPS)
In terms of Medical Cyber-Physical Systems (MCPS), these are contributions to the challenges of MCPS (Chapter 2.7.4).

- The novel anomaly detection approach (Chapter 3.3) and the OR table *position (CoG) surveillance* (Chapter 4) contribute to *Reliable Software, Interoperability, Context Awareness, security and privacy*.
- Modular E/E architecture (Chapter 3.4) contributes to *Reliable Software, Interoperability, and Certifiability*.
- Detecting anomalies enables automated safety and security measures and contributes to the *Autonomy* of medical devices to make decisions (Chapter 6.8).

7.2 Future Work

To improve the position surveillance for the OR table (Contribution 3. Anomaly Detection), real training data from actual surgeries, which is not available yet, is necessary in addition to data generated in test environments and simulations (Chapter 6). Pre-training the data-based models with data from the digital twin simulation and real-world test environments of all key scenarios builds the foundation for training with real data to increasingly adapt to actual behavior. Thus, the collected data can be incrementally used to improve the digital twin and the hybrid check models. Furthermore, correcting the current estimate to the probable value is a possible approach to increase the availability of the medical device after the detection of an anomaly. This can be achieved by using the LSTM network's predicted values or the reconstructed window from the AE. In this way, not only can the anomalies be compensated for, but the availability is also increased in the event of such incidents, thereby increasing safety.

The hybrid check for the position surveillance of the CoG could be further improved by several extensions:

- **External Information:** E.g., use position data provided via SDC by other systems or check plausibility against their dynamic behavior
- **Additional Sensors:** Integration of sensors such as IMUs (Chapter 5.2, Fig. 5.7) that provide acceleration and velocity measurements
- **Adaption to Scenarios:** E.g., use an Interacting Multiple Model (IMM) filter to model the likelihood of different scenarios, such as patient positions, while using different dynamic checks
- **Ensembles:** Combining multiple data-based models improves robustness and accuracy in different scenarios (Chapter 6.7)

Furthermore, the SDC network anomaly detection can be further elaborated. Since anomaly detection for a single system applies to interoperable medical devices to take a whole OR into account, the different devices may observe each other, creating a robust SoS that would need to be compromised as a whole to cause

any harm but also improves safety by leveraging monitoring capabilities. Here, *context-aware fleet security* approaches for cars, as proposed by Grimm et al., may prove to be a valuable starting point [GS22]. An ensemble for the medical devices in an SDC network can be derived from the setup for the mixed E/E architecture OR table (Fig. 5.3). Here, the medical devices are connected over SDC and thus an SDC network anomaly detection (Network Intrusion Detection System (NIDS) Chapter 4.6). With the fusion of the sensor data of several medical devices in the OR and the Electronic Health Record (EHR), patient health states could be determined more comprehensively to improve monitoring and create a digital twin of the patient. These approaches address the MCPS challenge of context awareness for the patient's health status (Chapter 2.7.4). Hence, with the hybrid and distributed anomaly detection, an initial step is taken, laying the foundation for a context-based anomaly monitoring system in the OR.

A Appendix

A.1 Medical Devices

Figure A.1: Mobile OR table Maquet Yuno II with mounted accessories

Figure A.2: Back plate board accessory

Figure A.3: Head rest accessory

Figure A.4: Motorized joint module

Figure A.5: Supine position [Get23b]

Figure A.6: Beach chair position [Get23b]

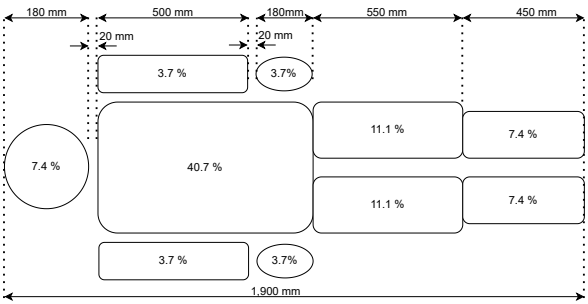


Figure A.7: Patient mass distribution according to [IEC20]

A.1.1 Life Cycle and Classification of Medical Devices

The sequence of phases in a product's life, from initial conception to final disposition, is called the life cycle [ISO21a]. According to ISO81001-1 [ISO21a], the life cycle for health software and health IT systems with a focus on safety and security also includes the *implementation* phase of the medical device in, e.g., hospitals and the *clinical use* phase apart from the product *design and development* phase.

Table A.1: Medical device classification according to Food and Drug Administration (FDA) and Medical Device Regulation (MDR) based on [PHS23]

Risk	FDA Class	MDR Class	Example
Low	Class I	Class I	Bandages
Moderate	Class II	Class IIa	X-Ray-Machines
Moderate to High	Class II/III	Class IIb	Defibrillators
High	Class III	Class III	Pacemakers

A medical device is classified [Eur17] based on its intended use (Table A.1). If the software of a medical device controls another one with a higher safety classification, it automatically inherits the classification of the controlled device [Eur17]. Conversely, this means that if it only reads the values of a higher-rated device, it will retain its rating.

A.1.2 Hybrid Operating Room (HOR)

The HOR enables imaging-guided surgery by connecting an OR table with an angiography system, a C-shaped medical device used for inter-operative imaging with X-ray technology (Fig. A.8). Angiography is the examination and visualization of blood vessels and vascular networks through introducing and detecting a contrast agent [CSL12]. Before the invention of the HOR, it was necessary to transport the patient between the hospital's imaging room and the OR, which was time-consuming and posed more risk to the patient.

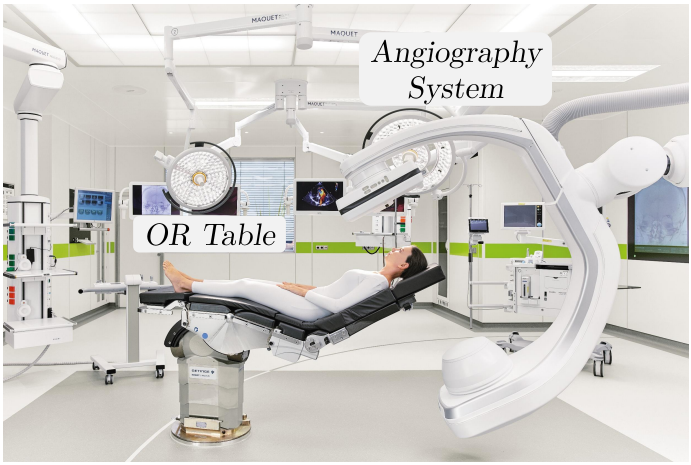


Figure A.8: OR table integration with an angiography system in an HOR

HORs [NHF+12] typically have system OR tables (Chapter 2.6.3) with a static, non-mobile column. These static columns are embedded in the floor, allowing accurate relative position determination to the angiography system. Hence, the OR table and the angiography system must interoperate to enable imaging in the HOR. Therefore, the current rotation of the column and other position information must be measured and communicated to the angiography system.

A.1.3 Integration with Surgical Robots

The past two decades' rapid technological advancements and digitization, an aging population, and a shortage of healthcare personnel, have led to a significant growth market for medical robotics [BKSU23]. Robotic surgery means procedures that are based upon telemanipulation principles (Fig. A.9) [TEMH+20]. Among the most successful surgical robots is Intuitive's da Vinci system, but other companies are trying to get a share of the market as well [RGRS+18].

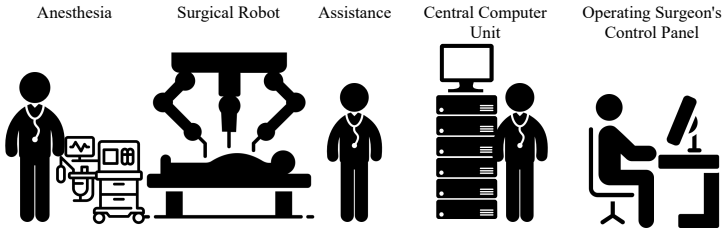


Figure A.9: Surgical robots in the OR based on [St.22] (robotic surgery icons designed by smashing-stocks, anesthesiology icons designed by bsd, doctor icons designed by Freepik, computer icons designed by Prosymbols Premium from Flaticon)

The DLR developed a multi-arm system called MiroSurge. Competence from the MiroSurge robot ultimately led to the Hugo system by Medtronic in 2021 [BKSU23]. *MiroSurge* [DLR21] [HKT+10] is based on Data Distribution Service (DDS) Connext (Chapter A.9.4) that was used to realize the communication of three MIRO robots, the endoscope, the surgeon's robotic controls, and the surgeon and technician user interfaces. Today, integrations for OR tables with surgical robots exist. The Trumpf TruSystem™ 7000dV OR table, for example, has integration with Intuitive's da Vinci, allowing the synchronization of the robot movements in alignment with the OR table by using a virtual pivot point for the surgical instruments [Hil22b]. Systems such as the Freehand endoscope holder or the Flex System from Medrobotics can be attached to the OR table's slide rail. The MiroSurge system is even integrated with the OR table. An overview of the different systems can be found in [PAK+18].

A.1.4 Digital Imaging and Communications in Medicine (DICOM)

Digital Imaging and Communications in Medicine (DICOM) is a de-facto standard published in 1993 [Rod09] for the generation, transmission, processing, and storage of image data in the medical area [Sch08] [TEMH+20]. It transfers image and video data from a server to a medical device (Fig. A.10). Data is encoded binary because it was standardized before the invention of web technologies such as Extensible Markup Language (XML). This leads to the supplementary standards Web Access to DICOM Persistent Objects (WADO), which defines a web-based service for accessing and displaying persistent objects such as images or medical imaging reports [Nat11] and DICOM Structured Reporting (DICOM-SR) used for transmission and storing of clinical documents [DIC04][Rod09].

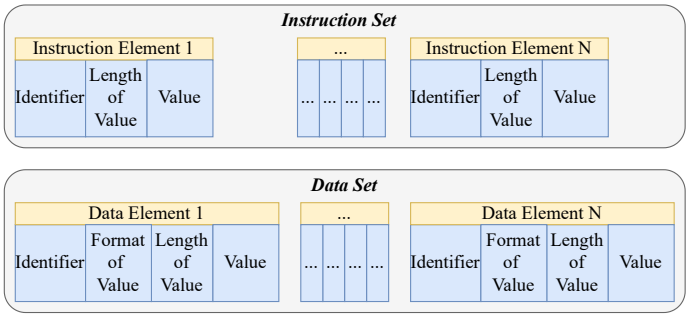


Figure A.10: DICOM instruction set and data set structures [Sch08]

Although OR tables do not directly provide image data, the standard must be applied when communicating with other image-generating devices that rely on the standardized data format, such as angiography systems. Furthermore, clinicians rely on the unified representation of the data, and therefore, the OR table displays the patient position data on the remote control.

A.1.5 Health Level Seven (HL7)

The Health Level Seven (HL7) is a popular set of not-for-profit standards for hospital information systems, which is located on application layer 7 of the Open Systems Interconnection (OSI) reference model (Fig. 2.14) and was first established in 1987 [BS13]. Compared to DICOM, it is focused on communicating clinical and administrative data through detailed work instructions [Sch08] excluding image data (Fig. A.11). The standards are elaborated by the HL7 standards group, an American national standards institute today, and already widely spread in several countries such as the United States, Australia, Canada, Germany or Japan [Rod09]. The successor HL7 v2 followed in 1989 to integrate primarily administrative and clinical systems in the hospital and thus not in the OR.

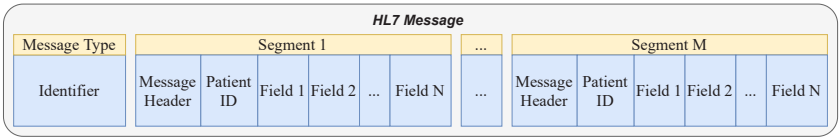


Figure A.11: HL7 message structure according to [Sch08]

Because of several drawbacks of HL7 v2, the development of HL7 v3 began in 1995 with new concepts that led to incompatibility with the previous version [BS13]. Thus, interoperability between both versions was only possible with elaborate translation software. HL7 v3 is based on an object-oriented model called Reference Information Model (RIM). In 2011, the development of HL7 Fast Healthcare Interoperability Resources (FHIR) began to streamline and accelerate the adoption of the standards by using open Internet standards where possible. For this purpose, RESTful (from REpresentational State Transfer (REST)) web services, which are Service Oriented Architecture (SOA)-based, were chosen for implementation [BS13]. Furthermore, HL7 is the base of the SOA-compatible Virtual Medical Record (vMR) [HL724].

A.1.6 Circulation Principle in the OR

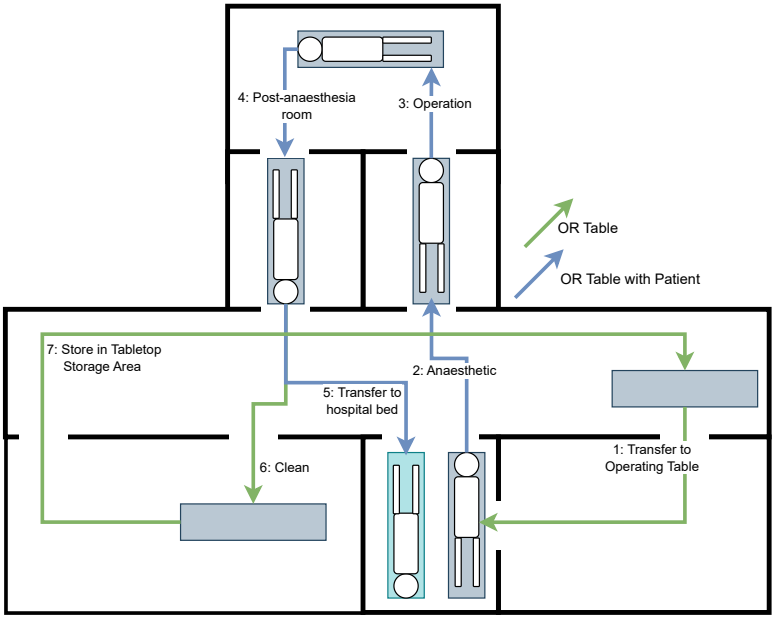


Figure A.12: Circulation principle in the OR with a system OR table [KAKA06]

A.2 Software and E/E Architecture

There are five layers of abstraction within the E/E architecture (Fig. A.13) [ST12]:

Functionality View: The functionality view contains all functions that the customer can directly experience. These functions are composed of features that "represent abstract functional characteristics of a system that end-users and other stakeholders can understand" [Int15].

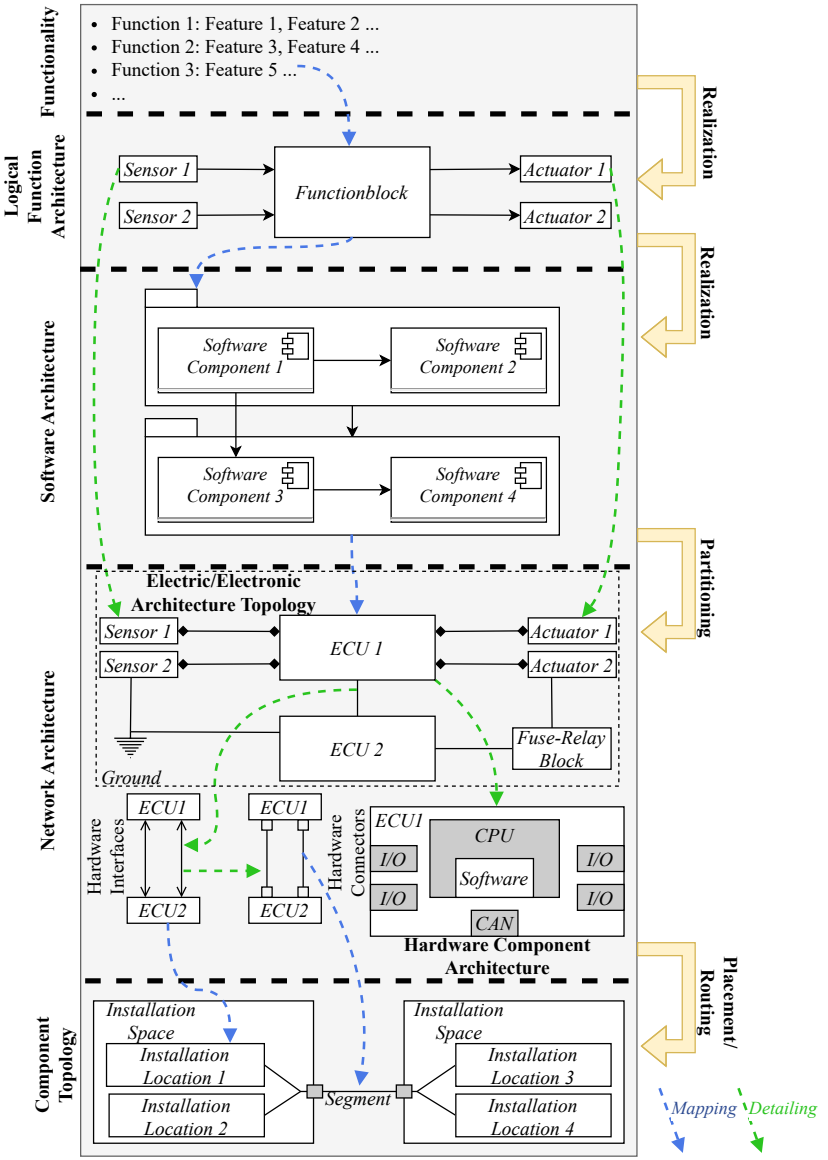


Figure A.13: Artifacts such as Electronic Control Units (ECUs), sensors and actuators, as well as layers of an E/E architecture based on [ST12] and [Obe21]

Logical Function Architecture View: This view describes the previously defined features as logical functions comparable to mathematical functions. These are part of an application that receives an input value and converts it into an output value, allowing users to perform their tasks [Int08].

Software Architecture View: The concrete implementation in code is not relevant on this abstraction level. Instead, the focus is on the software components that realize the logical functions, their corresponding interfaces, and their relationships with each other.

Network Architecture View: The network architecture describes the communication and power supply structure of the ECUs, sensors, and actuators. This layer is further divided into the communication structure layer, which describes the network composition and the deployed network technologies, and the power supply layer, which illustrates how the different components are supplied with energy.

Component Topology View: The component topology describes the lowest abstraction layer and specifies where the individual components are placed in the construction spaces. In addition, the placement of the wiring harness is defined.

A.2.1 Software Architecture

The main task of the software architecture design is to find a construction path leading to a complete solution in which functional and non-functional requirements are fulfilled. This path is not linear but is based on a mutual approach between the requirements engineer and the software architect. In addition, the effects of requirements can only be realistically assessed if a rough architecture has already been defined [Gha20]. The goal of software architecture itself is describable following the architecture in the civil engineering sector as an *artfully balanced triad* of usefulness, firmness, and beauty [Gha20]:

- **Usefulness**: The software fulfills the functional and non-functional requirements of the user/customer.

- **Firmness**: The software is stable and durable according to the required quality characteristics, so it can be extended without rebuilding the system.
- **Beauty**: The software is intuitive for the user (externally well-structured) and straightforward for the developer to understand to support maintenance and development (internally well-structured).

A.2.2 Software Quality

ISO/IEC 25010:2011 defines *software quality* as the entirety of characteristics and characteristic values of a software product to fulfill specified requirements [Gha20]. The standard also defines a set of main characteristics for *internal and external quality* [ISO11a][EG18], which can be further broken down into sub-characteristics (Table A.2).

The ISO/IEC 25010:2011 standard also defines a *quality-in-use* model of five characteristics [ISO11a][NDA12][EG18]:

- **Effectiveness**: To what extent does the system support the user in completing an intended task *completely*, *accurate* and without errors?
- **Efficiency**: To what extent does the system support the user in completing an intended task in a reasonable *investment of resources* such as time?
- **Satisfaction**: Is the system *useful* to the user and grants him *pleasure*, *comfort* and *trust*?
- **Freedom from Risk**: To what extent does the system mitigate the *environmental*, *economic*, *health and safety* risk (Chapter 2.1.3)?
- **Context Coverage**: Can the system be used in *all specified contexts* of use and in contexts *beyond those originally specified* in the requirements, while maintaining the other quality-in-use characteristics.

Quality characteristics may influence each other, so it is impossible to provide all to the same extent. A comprehensive overview of their dependencies is given in [Bal11]. For example, reliability in terms of availability can be very restrictive

Table A.2: Quality characteristics and their sub-characteristics [ISO11a]

Quality Characteristic	Sub-Characteristic Examples	Description
Functional Suitability	Functional Completeness, Functional Correctness, Functional Appropriateness	Do the functions of the software fulfill the requirements?
Reliability	Maturity, Availability, Fault Tolerance, Recoverability	Does the software keep its performance under defined constraints over a defined time?
Usability	Appropriateness Recognizability, Learnability, Operability, User Error Protection, User Interface Aesthetics, Accessibility	Is the software easy for the user to learn and operate?
Performance Efficiency	Time Behaviour, Resource Utilization, Capacity	Does the software save resources, runtime and storage?
Security	Confidentiality, Integrity, Non-Repudiation, Accountability, Authenticity	Are other persons or systems authorized to read or change data (Chapter 2.2)?
Compatibility	Co-Existence, Interoperability	Is the system able to exchange information with other systems?
Maintainability	Modularity, Reusability, Analysability, Modifiability, Testability	How big is the effort to eliminate errors or to make adjustments and changes?
Portability	Adaptability, Installability, Replacability	Is the software also executable on other systems?

for safety and security measures. As time is a critical factor in the medical area, the hospital staff cannot always enter a password when using a medical device. Therefore, it is necessary to compromise these characteristics to achieve an adequate solution for the device. In addition, it is advisable to monitor the quality characteristics throughout the life cycle of a system, as compliance with the processes is not a guarantee of the quality of a system [And13].

A.3 Communication Networks

CAN and Ethernet are widely used communication standards within the medical device area [PDDL15]. Other interfaces, such as Universal Serial Bus (USB) and Serial Peripheral Interface (SPI), are also commonly used, but these technologies are not applied to ECU networks of OR tables. CAN and Ethernet are both on layers 1 (physical layer) and 2 (data link layer) in the International Standards Organization (ISO)/OSI reference model (Chapter 2.5.1).

A.3.1 Controller Area Network (CAN)

CAN was first developed by Bosch in the 1980s to connect the rising number of safety-critical ECUs within cars. Later, it was also standardized in ISO11898-1, which started the ISO 11898 standard family describing the data link and physical layer of CAN [AHJ+18]. Since the mid-1990s, CAN has been the most widely used protocol in the automotive industry [Law13] and is only slowly being replaced by Automotive Ethernet (Chapter A.3.2). It is a multi-master serial bus where the messages contain up to 8 bytes of payload data. Each message has its unique identifier (Fig. A.14). The data frame type, which is indicated by the Identifier Extension Flag (IDE), CAN 2.0A uses 11-bit identifiers, while CAN 2.0B uses 29-bit identifiers [Law13]. The prioritization of the different messages is directly coded within the identifier so that the message with the lowest identifier value prevails in case of collision. All participants can generally read all messages on the connected bus due to its broadcast characteristic.

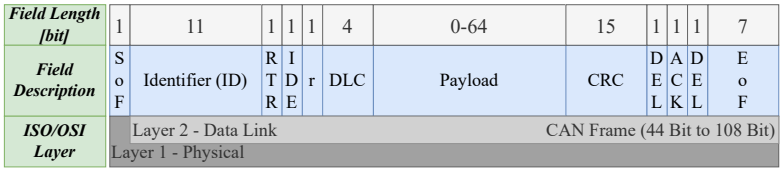


Figure A.14: CAN-Classic frame [GPS20]

The CAN standard defines high-speed CAN with up to 1 MBit/s and low-speed CAN with up to 125 kBit/s [ZS14]. Since the introduction of CAN with Flexible Data Rate (CAN-FD) in 2012 and its standardization as ISO11898-2:2016 in 2016 [AHJ+18], transmission rates of up to 64 Bytes per CAN message and a data rate of 5 MBit/s are also possible, while CAN-FD controllers still can participate in classic communication [Law13]. A comparison of CAN and CAN-FD can be found in [AHJ+18]. The bandwidth and bus length limitation of CAN are closely related to its advantage, determinacy, through the non-destructive arbitration mechanism for media access control. To achieve this, the propagation delay between any two nodes must be less than half of a Bit time [Law13].

Currently, the CAN-XL standard is being developed to close the gap between Ethernet 10BaseT1 and CAN [DG20, Rob21]. The third generation of CAN supports data field lengths of up to 2048 bytes and thus is designed to be integrated into Transfer Control Protocol (TCP)/Internet Protocol (IP) network systems [CAN22]. Furthermore, CAN XL is backward compatible with CAN-FD networks [Rob21]. Therefore, it is possible to mix signal-based and service-oriented communication on the same network. An example of a mixed communication approach can be found in [PVR+22]. Furthermore, the standard allows up to 256 logical networks to be used on a single physical network segment with the introduction of virtual CAN network ID (VCID) [CAN22]. While the 11-bit or 29-bit headers from CAN and CAN-FD are used for arbitration as well as addressing purposes, CAN XL separates these into an 11-bit Priority ID followed by an 32-bit Acceptance Field.

Because CAN was developed with a focus on safety applications almost 40 years ago, security concerns due to increased connectivity were outside the scope. The majority of attacks where the researchers gained access to the vehicle's internal communication were due to sending or suppressing CAN messages [Web19]. It is designed to withstand harsh and noisy environments without the slightest premonition that the car may be hacked in the future [Yos15]. Thus, it is considered the main vulnerability in automotive security in several publications [GPS20]. The broadcast characteristic, for example, allows unauthorized CAN nodes to read all transferred data, called *Frame Sniffing*. These data can afterward be used to decode the meaning of the messages and signals.

Furthermore, CAN provides no device authentication, so the message's source is unknown. Thus, an attacker can send spoofed messages (Spoofing) and inject messages into the communication called *Frame Injection*. Another weakness of CAN is the lack of encryption, which is necessary to protect the communication from being *sniffed*. In addition, the 8 bytes of data for classic CAN do not suffice to provide AES-based encryption according to the Rijndael algorithm that needs 128-bit blocks [Yos15]. The data length for modern encryption algorithms is too short, so CAN-FD with 64 bytes does not make a difference. The CAN XL standard provides an optional data-link-layer-security protocol called CADsec [CAN22]. Thus, it tackles one of the central vulnerabilities of the previous generations by introducing a 16-byte authentication tag, while the extended data frame length of up to 2048 Byte will allow the use of state-of-the-art encryption algorithms.

A.3.2 Ethernet

Ethernet defines a hardware and software standard for computer communication on the data link and physical layer (Fig. A.15). It was invented by Robert Metcalfe at the Xerox Palo Alto Research Center in May 1973 but was not officially called Ethernet until the IEEE 802.3-2012 [IEE16] standard in 2016 [Ste18]. It is the most widely used communication technology for wired local area networks [Rob17][ILW+23] due to its universal applicability [Ste18].

In comparison to CAN, modern Ethernet-based communication is a switched collision-free point-to-point connection. Therefore, participants only communicate with their respective switches and cannot read all messages. With standards like the IEEE802.3cg [IEE19], which has a lower data rate of 10 MBit/s, a bus topology without the need for switches similar to CAN can be achieved. The more advanced standards like 100BASE-T1 (IEEE802.3bw [LAN16]) and 1000BASE-T1 (IEEE802.3bp [IEE16]) are capable of 100 Mbit/s respectively 1 GBit/s but need a switch. For external communication with vehicles (V2X, Internet), in the automotive domain, there is the wireless protocol IEEE 802.11 definition for the physical layer [BE17].

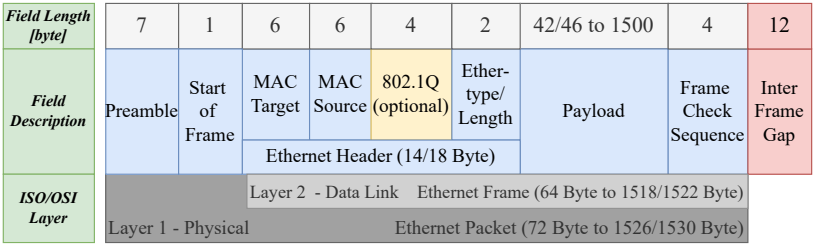


Figure A.15: Standard Ethernet frame format [Ste18]

Although the physical layer defers between the different domains, most domains like automotive or IT rely on Internet Protocol (IP) in the network layer as well as Transfer Control Protocol (TCP) and User Datagram Protocol (UDP) in the transport layer (ISO/OSI Fig. 2.14). On top of these, different middleware technologies enable service-oriented communication (Chapter A.9.4) [GPS20].

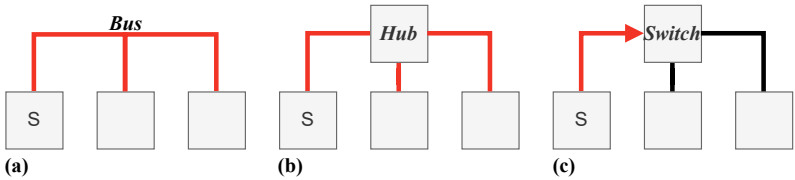


Figure A.16: Ethernet collision domains on sending (sender: S) in comparison: bus (a), hub (b) and switch (c) according to [Ste18]

Classic Ethernet requires collision handling similar to CAN, since each participant is allowed to use the medium simultaneously. The switched Ethernet, uses switches to create isolated point-to-point connections between two nodes to avoid collisions. This also enables to increase data rates and longer cable lengths compared to the hubs of classic Ethernet (Fig. A.16).

Unlike CAN, Ethernet initially was not designed for systems requiring hard real-time in harsh environments. The advantages of flexibility, scalability, bandwidth, and cost-effectiveness have led to various adapted Real-Time Ethernet (RTE)

standards in different sectors such as automotive, industrial, and avionics (see [Ste18] for an overview). In industrial applications, Ethernet-based protocols are slowly replacing other industrial bus communication protocols [Dec05].

With its ability to segment into virtual networks using Virtual Local Area Network (VLAN) (Fig. A.17) and the sufficient bandwidth that enables encryption and authentication algorithms, Ethernet offers opportunities to improve security [Ste18].

	Optional 802.1Q VLAN-Tag (4 Byte)			
Field Length [bit]	16	3	1	12
Field Description	Tag Protocol Identifier (TPID)	Tag Control Information (TCI)	Drop Eligible Indicator (DEI)	Virtual LAN Identifier (VLAN-ID)

Figure A.17: Optional VLAN-Tag structure according to IEEE 802.1Q [IEE]

Another advantage over classic CAN and CAN-FD is that each Ethernet frame contains its source and target address, a precondition for authenticity and authorization. Also, by using switches to establish point-to-point connections, not every node can read all the data, hindering attackers to conduct sniffing or spoofing attacks. Due to its limitation to layers 1 and 2 of the ISO/OSI layer reference architecture, top layers such as TCP/IP need to provide additional security measures. Thus, already established and proven secure protocols from the IT sector can be used [Ste18]. Especially the overhead incurred by these protocols requires a larger bandwidth than is available from classic CAN, which implicitly requires a performance increase of the ECUs. An elaborate discussion on the security challenges and potentials of Ethernet for automotive systems can be found in [CSKP16].

A.4 Kalman Filters

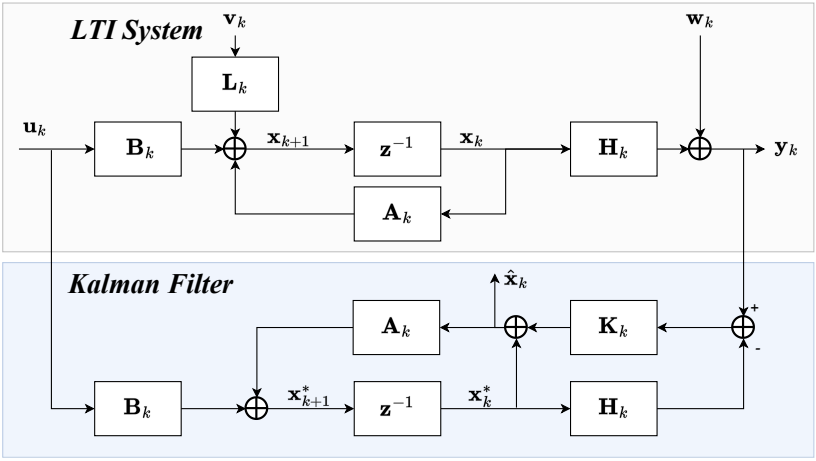


Figure A.18: Signal process with stochastic disturbances of an observed system with measurable input variables for a time-discrete Linear Time Invariant (LTI) system [FH19] with KF based on [PB17][Wen11][PZSS24]

A.4.1 Extended Kalman Filter (EKF)

Since the classical KF (Chapter 2.3.1) is not designed for nonlinear processes or measurement relationships, one approach to make this possible is the EKF. It linearizes the model around the current mean and covariance in each step k [WB06]. Although the overall process (Fig. A.19) stays similar to the KF, the prediction and the measurement are based on nonlinear functions. Thus, the prediction of the estimated state $\mathbf{x}^*$ is based on the nonlinear state transition function $f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, 0)$. Moreover, the projection of the estimated state $\mathbf{x}^*$ onto the observation $\mathbf{y}$ is based on the nonlinear observation function $h(\mathbf{x}^*_k, 0)$.

Table A.3: KF & EKF variable overview with dimensions m, l and n [PB17]

Variable	Description
$\mathbf{x}^{*(m)}$	Predicted system state vector
$\hat{\mathbf{x}}^{(m)}$	Estimated system state vector after new observation $\mathbf{y}$
$\mathbf{u}^{(l)}$	System input vector
$\mathbf{y}^{(n)}$	New observation/measurement vector
$\mathbf{B}^{(m\times l)}$	Dynamics of the system input $\mathbf{u}$ and projection on system vector $\mathbf{x}$
$\mathbf{K}^{(m\times n)}$	Kalman-Gain-Matrix to project the residuals onto the correction of the system state
$\mathbf{A}^{(m\times m)}$	Transition-Matrix to propagate the system state to next time point
$\mathbf{P}^{*(m\times m)}$	A-priori covariance matrix of the predicted system state before new observation $\mathbf{y}$
$\hat{\mathbf{P}}^{(m\times m)}$	A-posteriori covariance matrix of the estimated system state after new observation $\mathbf{y}$
$\mathbf{Q}^{(m\times m)}$	Process noise covariance matrix to introduce uncertainties caused by modeling errors or changing boundaries
$\mathbf{H}^{(n\times m)}$	Observation matrix projecting the system states on the measurement
$\mathbf{R}^{(n\times n)}$	Covariance matrix of the measurement noise
$\mathbf{L}_k$	Input matrix projecting the input noise $\mathbf{v}_k$ on the system state $\mathbf{x}_k^*$

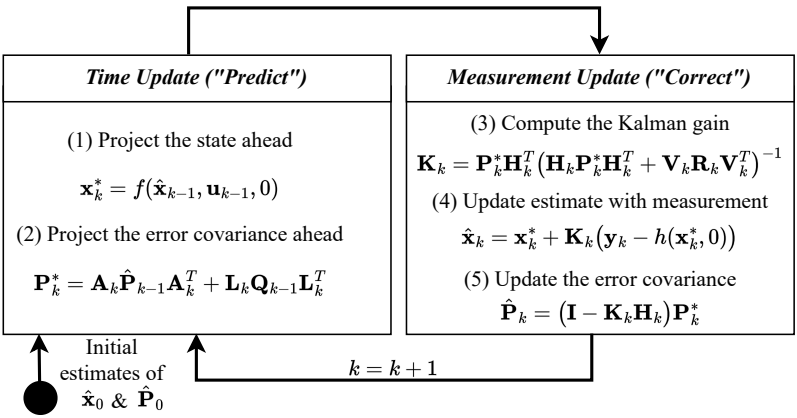


Figure A.19: EKF operation for each iteration k [WB06] (Table A.3). The measurement noise matrix $\mathbf{V}_k$ can be neglected if the noise of the measurements is expected to be white [Wen11]

The nonlinear functions are used to derive the system matrix $\mathbf{A}_k$ from the Jacobian matrix of the partial derivatives of f concerning $\hat{\mathbf{x}}$:

$$\mathbf{A}_{i,j} = \frac{\partial f_i}{\partial \mathbf{x}_j}(\hat{\mathbf{x}}_\mathbf{k}, \mathbf{u}_k, 0) \tag{A.1}$$

and $\mathbf{H}_k$ from the Jacobian matrix of h with respect to $\mathbf{x}^*$ at each time step k:

$$\mathbf{H}_{i,j} = \frac{\partial h_i}{\partial \mathbf{x}_j}(\mathbf{x}^*_\mathbf{k}, 0) \tag{A.2}$$

A.4.2 Unscented Kalman Filter (UKF)

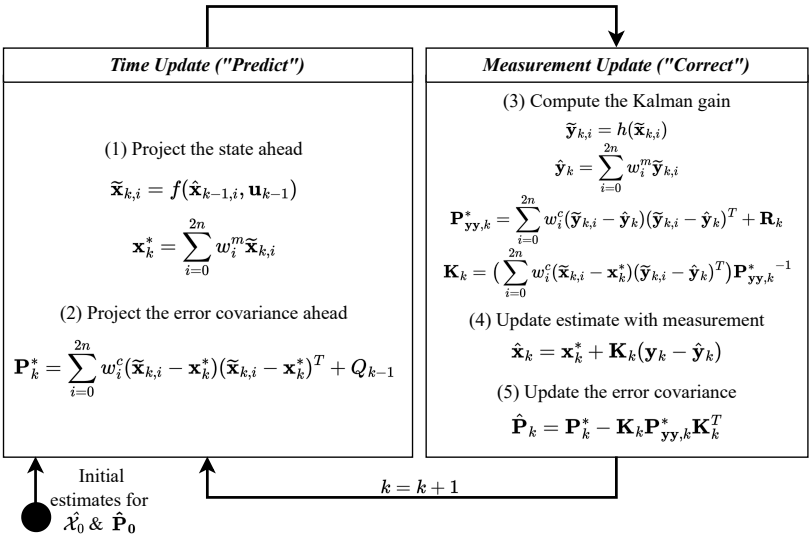


Figure A.20: UKF operation for each iteration k [Lab20] (Tables A.3 & A.4)

The UKF belongs to Sigma-Point-Kalman-Filters [Wen11], an alternative for EKF (Table 2.1). It is a more precise option for dealing with nonlinear processes than the EKF [van04] and is especially suitable for nonlinear scenarios (Table 2.1) [Lab20]. This is achieved using sigma points with weights to transform

the probability distribution of states through the nonlinear function (Fig. A.20). Afterward, it calculates new mean and variance values based on these transformed points [ATGSK21].

Table A.4: UKF variable overview addendum to Table A.3

Variable	Description
$\widetilde{\mathbf{x}}_i$	i-th sigma point of the predicted state
$\widetilde{\mathbf{y}}_i$	i-th sigma point of the predicted measurement
$\mathcal{X}$	Set of i sigma points $\widetilde{\mathbf{x}}_i$ of the predicted state
$\hat{\mathbf{y}}_k$	Predicted measurement
$\mathbf{w}_i^m \& \mathbf{w}_i^c$	Sigma point weights for mean and covariance calculation

The set of sigma points $\mathcal{X}$ consisting of $\widetilde{\mathbf{x}}_\mathbf{i}$ is calculated by a *sigma function* $s(\hat{\mathbf{x}}_i, \hat{\mathbf{P}})$. For the calculation of sigma points, the *Van der Merwe* scaled sigma points have been mainly used in industry and research, as they represent a balanced compromise between accuracy and performance [Lab20]:

$$\mathcal{X}_i = \begin{cases} \mu & \text{for } i = 0 \\ \mu + [\sqrt{(n+\lambda)\Sigma}]_i & \text{for } i = 1..n \\ \mu - [\sqrt{(n+\lambda)\Sigma}]_{i-n} & \text{for } i = (n+1)..2n \end{cases} \tag{A.3}$$

while the according weights can be determined with the following equations:

$$W_0^m = \frac{\lambda}{n+\lambda} \tag{A.4}$$

$$W_0^c = \frac{\lambda}{n+\lambda} + 1 - \alpha^2 + \beta \tag{A.5}$$

$$W_i^m = W_i^c = \frac{1}{2(n+\lambda)} \text{ for } i = 1..2n \tag{A.6}$$

A.5 Anomaly Detection and Machine Learning

If a set of rules can be specified for a problem to provide a desired output based on a given input [Alp20], the procedure corresponds to traditional programming. In contrast, machine learning is preferable when the traditionally written algorithm would result in a long list of complicated rules, e.g., for spam filters.

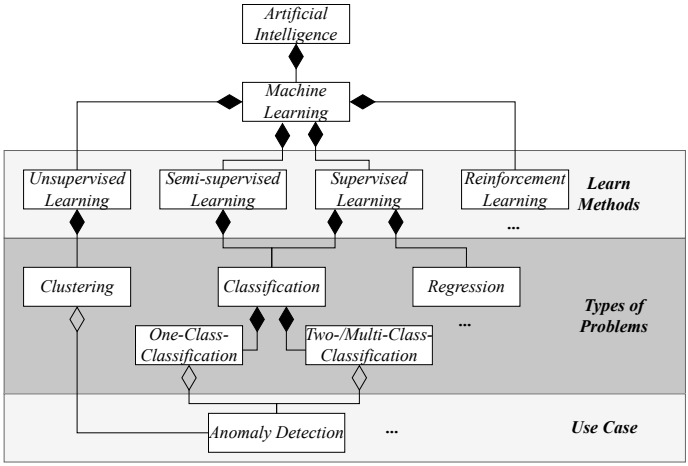


Figure A.21: Overview of Artificial Intelligence (AI) and machine learning [Web19]

Data is a collection of data points that contain information, while objects, instances, or observations are called data points. In addition, the information in a data point, also called a feature vector, is a combination of properties, attributes, and features. Data points can be unordered if they represent single independent observations or ordered if they can relate to each other. Examples of relations are temporal when a physical value like velocity is sampled at a particular sampling rate, and spatial, such as the position of different objects in a room. When an order relation exists ($x > y$) between the different features, they are called categorical; otherwise, they have continuous values. The value ranges of the individual attributes form the basis for the mathematical feature space built upon all data

points' features. Data with a single property are considered unidimensional or univariate, while data with multiple features are considered multidimensional or multivariate. Attributes are data types, and features are the data type and its value. Further basic terms in machine learning are summarized in Table A.5.

Name	Description	Function
Input (Features)	The perceptron takes multiple input features $(x_1, x_2, \ldots, x_n)$ from the input layer.	The perceptron computes a weighted sum of these input features.
Weights $(W_1 \ldots W_n)$	Each input feature is associated with a weight, which signifies the importance of that feature.	The weighted sum is calculated by multiplying each input by its corresponding weight.
Bias (b)	The bias is an additional parameter that captures an offset or a baseline.	The bias is added to the weighted sum to produce the final input to the activation function.
Weighted Sum	The result of multiplying each input by its weight, summing these products, and adding the bias.	Weighted Sum $= \sum_{i=1}^{n}(W_i \cdot x_i) + b$
Activation Function	Introduces non-linearity to capture relationships in the data.	Output $= f(\text{Weighted Sum})$
Output	The result of applying the activation function to the weighted sum.	Output $= f(\text{Weighted Sum})$

Table A.5: Basic terms and components in machine learning

A.5.1 Metrics for Anomaly Detection

A *confusion matrix* is a table used to describe a classification model's performance on a test data set. It contains information about the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions (Table A.6).

		Predicted Label	
		True	False
True Label	True	TP	FP
	False	FN	TN

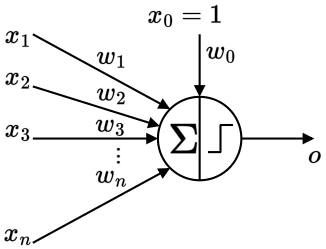


Figure A.22: Single layer perceptron with one neuron [Web19]

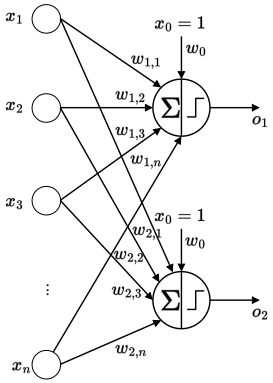


Figure A.23: Single layer perceptron with two neurons [Web19]

True Positive	Instances correctly predicted as positive (correctly classified as anomaly
True Negative	Instances correctly predicted as negative (correctly classified as normal)
False Positive	Instances incorrectly predicted as positive (incorrectly classified as anomaly)
False Negative	Instances incorrectly predicted as negative (incorrectly classified as normal)

Table A.6: Confusion matrix

A.5.2 Machine Learning

Autoencoder (AE)

An AE is an artificial Neural Network (NN) used for unsupervised learning. It aims to learn a compact representation (encoding) of input data and then reconstruct the original data from this representation (decoding) [GBC16]. The network consists of an *encoder*, which compresses the input into a lower-dimensional representation, the so-called *latent space* or *bottleneck* layer, and a *decoder*, which reconstructs the input from this representation (Fig. A.24). Both the encoder and the decoder may consist of one or more layers, whereby the encoder gradually reduces the dimensionality of the input.

Metric	Description	Formula
Accuracy:	Measures the overall correctness of the model. It is calculated as the ratio of correctly predicted instances (both true positives and true negatives) to the total number of instances.	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision:	Measures the accuracy of positive predictions. It is calculated as the ratio of correctly predicted positive observations (TP) to the total predicted positives (TP + FP)	$\frac{TP}{TP+FP}$
Recall	Measures the ability of the model to capture all relevant positive instances (ratio of correctly predicted positive observations to the total actual positives).	$\frac{TP}{TP+FN}$
F1 Score:	The harmonic mean of precision and recall. Provides a balance between precision and recall.	$\frac{2\cdot(\text{Precision}\cdot\text{Recall})}{\text{Precision}+\text{Recall}}$
ROC AUC Score:	Area Under the Receiver Operating Characteristic curve. Measures the area under the curve representing the trade-off between true positive rate and false positive rate.	/
False Positive Rate (FPR):	Proportion of actual negatives that are incorrectly identified as positives. It is calculated as:	$\frac{FP}{TN+FP}$
False Negative Rate (FNR):	Proportion of actual positives that are incorrectly identified as negatives. It is calculated as:	$\frac{FN}{TP+FN}$

Table A.7: Anomaly detection metrics [Gér19] [CTJ21]

Mathematically, the encoder compresses the input vector $\mathbf{x}_i \in \mathcal{X}$ of dimension n to a latent variable $\mathbf{z}_i$, which is part of the latent space $\mathcal{Z}$ of dimension c, where $\mathbb{R}^c < \mathbb{R}^n$:

$$\mathcal{E}_{\Theta_E} : \mathcal{X} \to \mathcal{Z} \tag{A.7}$$

Afterward, the decoder reconstructs $\hat{\mathbf{x}}_i \in \hat{\mathcal{X}}$ of the original variable $\mathbf{x}_i$:

$$\mathcal{D}_{\Theta_D} : \mathcal{Z} \to \hat{\mathcal{X}} \tag{A.8}$$

An AE can also be implemented using 1D convolutional layers, which are particularly useful for sequence data or signals and are so-called 1D Convolutional

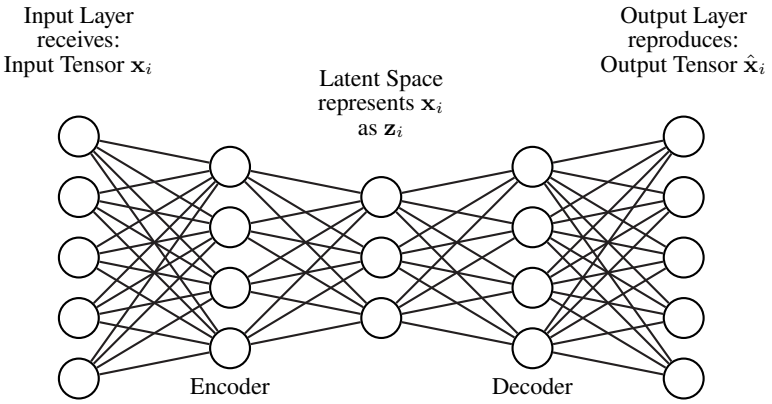


Figure A.24: Functional principle of an AE based on [PZSS24]

Neural Network (1DCNN) AEs. They are a common approach for anomaly detection, especially when dealing with sequential or time-series data [KAA+21]. The architecture of 1DCNN AEs allows them to capture local patterns and temporal dependencies in the data, making them valuable for identifying anomalies in sequences. For applications such as intrusion detection in network traffic, fault detection in machinery, or anomaly detection in physiological signals, 1DCNN AEs have shown effectiveness in capturing spatial and temporal aspects of the data.

Long Short-Term Memory (LSTM) Networks

LSTM networks are a specialized type of Recurrent Neural Networks (RNN), which are NNs where neuron outputs are propagated backward. Their architecture effectively tackles the vanishing gradient problem in traditional RNNs [HS97]. This problem arises due to the limitation of these traditional RNNs, which can experience vanishing gradients, rendering optimization via gradient descent impossible [BSF94]. LSTM models are well-suited at processing and predicting data sequences. Their key strength lies in their capacity to identify long-term dependencies in sequential data, which they achieve by integrating memory cells and

various gates to regulate information flow through the network based on the current cell state c_t, the hidden state h_t and the cell input x_t [HS97]. The components of an LSTM architecture comprise (Fig. A.25):

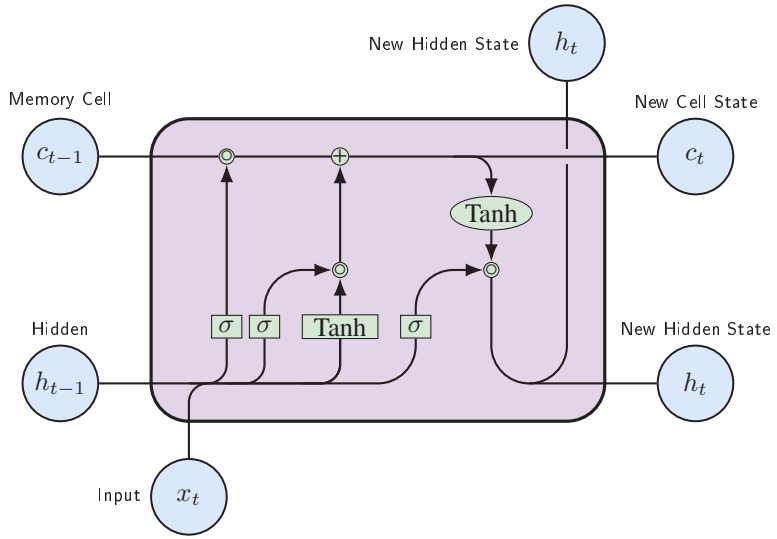


Figure A.25: Structure and states of an LSTM-Cell based on [PZSS24][HS97]

1. **Memory Cell:** Allows the network to store information over long sequences in the cell state c_t.
2. **Input Gate:** Regulates the flow of information into the memory cell, determining which information from the current input x_t should be stored.
3. **Forget Gate:** Controls removing information from the cell state c_t, deciding what information is no longer relevant.
4. **Output Gate:** Determines the output, which is also the hidden state h_t, of the LSTM based on the current input x_t and the cell state c_t.

Isolation Forest (IF)

The IF algorithm is a machine learning technique that excels in detecting anomalies in unsupervised learning. It is particularly effective in identifying outliers or anomalies in datasets where most instances are regular. The algorithm employs a forest of random decision trees to isolate anomalies [LTZ08]. Each tree is recursively grown by randomly selecting features and dividing data points until individual instances are isolated or the tree reaches its maximum depth. During training, the IF builds multiple trees using a subset of the data, and during testing, it identifies instances with low isolation scores as potential anomalies. This is due to anomalies requiring fewer splits to be isolated than regular instances, resulting in a lower average path length. The main components of the IF include:

1. **Isolation Trees:** Individual trees in the forest isolate instances based on random feature splits. Isolation means "separating an instance from the rest of the instances" [LTZ08]. They are proper binary trees, as each node has zero or two daughter nodes.
2. **Isolation Score:** A measure of how clearly a data point can be isolated, calculated based on the average path length in the trees.
3. **Ensemble of Trees:** Multiple isolation trees are combined to form an ensemble, and the overall anomaly score for a data point is determined by aggregating the scores from individual trees.

Unsupervised Learning	Process unlabeled data
Resource Intensive	Run on resource-constrained runtime environments
Multidimensional Feature Set	Handle multidimensional feature sets
Amount of Hyperparameters	Few hyperparameters that need to be trained
Successful Adoption	Successfully applied in other examples available in the literature

Table A.8: Requirements for anomaly detection algorithms based on [Koc22]

The IF is more resource-efficient than the AE and LSTM networks since it uses less memory, has a linear time complexity, and has lower computational cost due to not requiring inter-distances and densities calculation [LTZ08]. Thus, they are also executable on embedded systems using microcontrollers [APVA23].

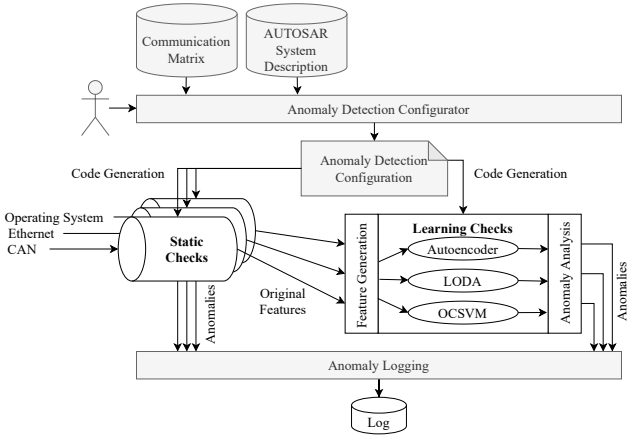


Figure A.26: Overall concept for the automotive observer based on [WKSZ18].

Table A.9: Anomaly types in signals examined by Weber [Web19] based on ISO 26262-5:2018

Anomaly Type	Description
Plateau	Freezes a signal value for a configurable duration (Stuck in range)
Positive Step Plateau	Positive value jump followed by constant signal (Short Circuit to Ground/V-bat)
Negative Step Plateau	Negative value jump followed by constant signal (Short Circuit to Ground/Vbat)
Positive Peak	Triangular shape with configurable sharpness and duration (Power spikes)
Negative Peak	Triangular shape with configurable sharpness and duration, opposite polarity to Positive Peak (Power spikes)
Positive Ramp Plateau	Signal value rises linearly, then remains constant before returning to original value (Disrupted feedback loop)
Negative Ramp Plateau	Signal value falls linearly, then remains constant before returning to original value (Disrupted feedback loop)
Noise	Original signal overlaid with white noise of configurable amplitude (Open Circuit)
Sine	Overlays original signal with a sine wave with a configurable number of cycles (n = D/10+1) (Oscillation)
Positive Step Offset	Sudden positive shift in signal value over a configurable duration (Offsets)
Negative Step Offset	Sudden negative shift in signal value over a configurable duration (Offsets)
Positive Ramp Offset	Signal value rises linearly over a configurable duration, then returns to original value with an added offset (Drift)
Negative Ramp Offset	Signal value falls linearly over a configurable duration, then returns to original value with an added offset (Drift)

Table A.10: Anomaly patterns [PRGS22]

Pattern	Description
Monitor Stationary States	If all participants have started and all services are set up, the context has reached a stationary state. Therefore, some state or payload values should remain in a specific range.
Monitor Number of Active Connections	While the number of connections for Request/Response (R/R) is defined during the development phase, this is not applicable for stream and publish/subscribe service types. An upper and lower threshold for an interval of possible active connections can be defined. If the number of connections during runtime is outside such an interval, it represents a deviation from expected behavior.
Service ID-Assignment Check	At the host level, exclusive features can only be observed with a host-based IDS. For example, each application is assigned to a specific service ID, which specifies the use of request/response IDs. The middleware can observe these assignments to verify if other combinations exist during runtime.
Payload Plausibility	Interception based on the service ID and requested parameter (API) requires decoding and plausibility checking of the contained data. Focusing on the payload of the services, these can be analyzed for anomalies in time series, for example, as presented by Weber et al. [Web19]. Assuming future architectures will increasingly consider the zero trust principle, payload plausibility checks can only be performed at the host level since connections are end-to-end encrypted.
Effect Chains for R/R	The invocation of R/Rs may result in an effect chain because the orchestration of services is based upon other services. Thus, a specific service will result in other services being executed and thus can be trained, e.g., according to [RLF+20]. Another approach is to consider the timing of the specific services invoked within an effect chain. The order of occurrence for R/Rs is predictable by the specification within an automotive or medical SOA network, as these are usually fixed for primary services/functions.
Publish / Subscribe Causality	For publish/subscribe patterns, it is determinable which change within the payload of a publish/subscribe service results in a change of the payload of another service or vice-versa. A machine learning algorithm trained with correlating data can determine this causal dependency.
Stream Causality	For streams, the expected behavior is more challenging to determine. In this case, checking the causality according to an exclusion procedure is more manageable. In case a stream-service *A* is missing, then a stream-service *B* might also be missing because it is based upon the stream-service *A*.
Context Plausibility	The state of different devices and the payload of different services can be fused to create a context of a device ensemble (e.g., the OR). Thus, it is possible to make a plausibility check if a device state is possible within a similar system context, as proposed by Grimm et al. [GS22]. If the states and payloads are not plausible within a given overall context, an anomaly is detected.

A.6 Robotics

A.6.1 Connectivity Graph

A typical method for modeling the rigid body mechanisms of a robot is to use a connectivity graph that represents the links and joints of the robot. It is a graphical representation of the robot's structure that shows how the links are connected through joints. Each node of a connectivity graph represents a rigid body, and the connection of these is modeled with arcs representing joints. For mobile robots, one body of the graph is a *floating base*, with a fictitious 6 Degrees of Freedom (DoF) joint between the base and the floating base. Arcs that connect a node with itself, meaning that a rigid body is connected with itself via a joint, are not allowed. Kinematic loops, which are closed paths in the graph that do not cross any arc more than once, are allowed. Robot mechanisms without kinematic loops, which are closed chains formed by connecting the last link to the first in a sequence of joints and links, are called kinematic trees.

Furthermore, if a body of a kinematic tree has at least two child bodies, the kinematic tree is called branched. As in this dissertation, OR tables are always branched kinematic trees (Fig. A.27), and therefore, loops are not considered here any further. This has the consequence that each joint only has one parent.

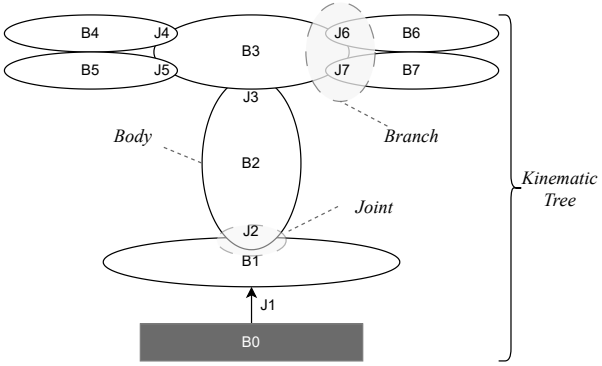


Figure A.27: Connectivity graph of an OR table with $N_B = 7$ and $N_J = 7$

The moving bodies B_i with a total amount of bodies N_B of a system are numbered using the *regular numbering scheme*, starting with the fixed base B_0 and all other bodies B_i in any order, with each child body numbered higher than its parent. The joints J_i with the total amount of joints N_J are numbered so that joint i links body i to its parent. Each joint J_i needs a direction to describe its velocities and forces, mostly from parent to child. The velocity of a joint is then defined as the relative velocity of the predecessor body p(i) to the successor body s(i) of the joint J_i. Moreover, the joint force is the force that is applied to the successor body [SK16].

A.6.2 Flexible and Deformable Elements

Rigid bodies are an idealized representation that only applies to slow motion and small interacting forces. This representation is not sufficiently accurate with increasing speeds and forces, such as friction, compliance, and deformation. This results in the erroneous behavior of the robot when the kinematics, controllers, and dynamics are developed while neglecting these effects. Flexibility can be modeled by distributing it along a link or concentrating in a joint. For flexible joints, the elasticity at a joint i can be modeled with a spring. The corresponding spring stiffness K_i that is either torsional, e.g., for revolute joints, or linear, e.g., for prismatic joints, can be used together with the motor position ϑ to calculate the torque τ_{J_i} at joint J_i [SK16]:

$$\tau_{J_i} = K_i(\vartheta - q_i) \tag{A.9}$$

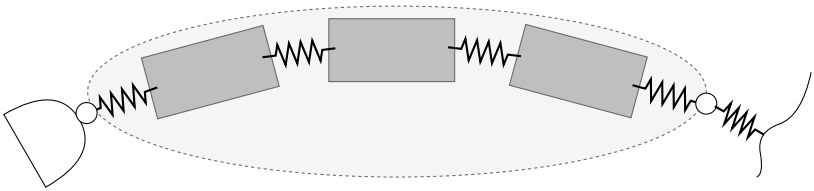


Figure A.28: Finite segment method [Ebe23]

Despite the Finite Element Method (FEM), which is the typical numerical method of choice for solid mechanics [LLP22] but very computationally intensive, the Finite Segment Method (FSM) [JR94a] [JR94b] is an approach to model flexible links with a set of rigid bodies connected by springs and/or dampers (Fig. A.28). It is used for slender objects such as flexible cables [AWW18].

A.7 Security

A.7.1 Defense-In-Depth Security Layers

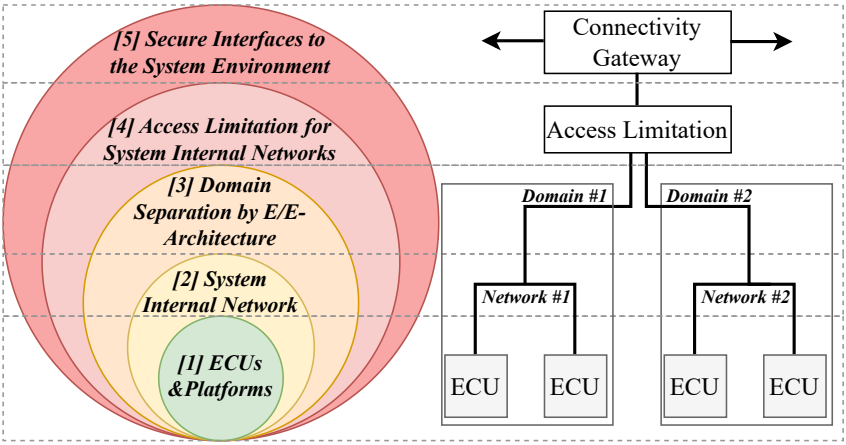


Figure A.29: Security layers [NWL+15] [Web19]

Layer 1 - ECUs & Platforms

On the first layer, trust anchors embedded in the individual ECUs ensure the integrity of the ECU software and data. Special hardware keeps the encryption keys secure. Furthermore, measures such as firewalls (Chapter A.7.3) and IDS (Chapter 2.2.4) to detect misbehavior and manipulation are deployed here.

Layer 2 - In-Vehicle network: Access limitation for system internal network
Communication within a domain of the components is secured, while the main task at this layer is to maintain the integrity of the in-vehicle signals. One approach is to secure messages with cryptographic measures such as Message Authentication Code (MAC). Since these algorithms are generally computationally intense, the payload is rarely encrypted at this level.

Layer 3 - E/E architecture Domains: Separation and protection of domains
This layer is designed to protect the different areas of a system by separating them into physically distinct networks. If a data exchange between the domains is necessary, gateways handle the communication and monitor the traffic for anomalies and unauthorized access.

Layer 4 - Access limitation for system internal networks
If an attacker can access a control unit and send messages over the system networks, this should be limited to only the necessary networks. For a vehicle, this could mean that an infotainment ECU is not allowed to send a message to a safety-critical domain such as brakes or motor control. For an OR table, where domains are separated on spatial distribution (Chapter 2.5.2), this could mean that a tabletop ECU is not allowed to send a move command to the column ECU.

Layer 5 - External System Connections: Secure interfaces to the system environment
The last layer's task is to protect the system from malicious activities on external IT interfaces. Therefore, the system must only handle required and relevant data. Unused protocols and functions should be disabled to minimize the attack surface to its minimum. Furthermore, cryptographic algorithms and secure protocols for data transmission, such as Transport Layer Security (TLS), are mandatory for communication with external systems.

A.7.2 STRIDE

STRIDE is divided into the following six parts [Mic16]:
***S**poofing*: attacker impersonates a legitimate user (*freshness & authentication*)
***T**ampering*: attacker modifies stored data/message content (*integrity*)
***R**epudiation*: attackers' actions cannot be traced back (*freshness & non-repudiation*)
***I**nformation Disclosure*: attacker reads stored/sent data (*confidentiality & privacy*)
***D**enial of Service (DoS)*: attacker disrupts operation of a system (*availability*)
***E**levation of Privilege*: attacker performs actions without *authorization*

A.7.3 Firewalls

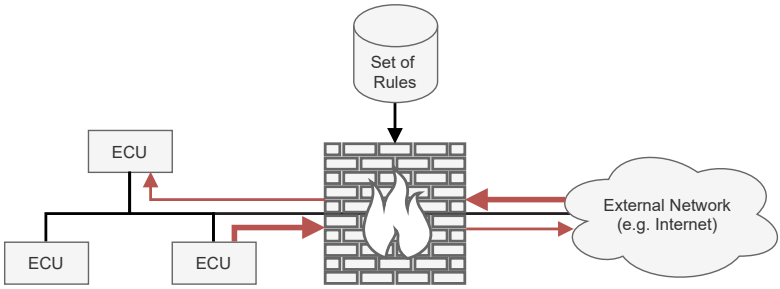


Figure A.30: Functional principle of a firewall according to [Web19]

Access control automatically prevents unauthorized access to resources [RGKS20]. One way to restrict access is to deploy firewalls, which are widely used in the IT industry [SH09]. In addition, firewalls are proactive measures because they preemptively minimize a system's potential attack surface through predefined rules [RGKS20]. They can be applied on different layers of the ISO/OSI model, namely 3, 4, and 7. An overview can be found in [RGKS20]. Fig. A.30 shows the functional principle of firewalls. A set of rules is applied to each message passing the firewall. If a message violates a rule, it will not be forwarded. Regarding the defense-in-depth layers (Fig. A.29), they are used in layers one, two, and five [Web19].

A.7.4 Identity- and Access Management (IAM)

Another way of restricting access control is Identity and Access Management (IAM), as it prevents unauthorized access to resources. Unlike firewalls, IAM ensures a rule-compliant use of resources by granting authorized access to specific resources based on an *access control model* [RGKS20]. A major weakness on the application level (Chapter 2.5.1) of vehicles was a weak IAM to ensure authorization, if implemented at all [SSG+22]. With upcoming connectivity to external devices based on SOAs, an IAM is necessary to limit the privileges of a compromised service (*principle of least privileges* [JM75]). This also is an outcome of the TARA performed by Kreissl [Kre17] for Scalable service-oriented Middleware over IP (SOME/IP) (Chapter A.9).

Table A.11: Extract of medical attribute information [PVR+22]

Category	Attribute Name	Attribute Value
Subject	angiography system	OP-1
Action	height	move up
Environment	OR_Table_state	locked
Resource	OR_Table	control

As there are several access control models for IAM, Rumez et al. [RGKS20] provide a detailed overview of these, among other security measures. In [PVR+22], an IAM using an Attribute-based Access Control (ABAC) approach is presented for an OR table interoperating with an angiography system in a network based on SDC (Table A.11 and listing A.1).

Listing A.1: Exemplary access control policy for medical devices

```
policy AccessMovementORtable {
    target clause
    attrib.role == "angiography system"
    apply firstApplicable AccessMovementORtable}

rule control_OR_table {
    permit target clause
    attrib.action      == "control"
    condition
    attrib.OR_table_state  != "locked" }
```

A.7.5 Heavens 2.0

Table A.12: Attack feasibility rating [LAO21]

Parameter Sum (A_{sum})	Attack Feasibility
$0.00 \leq x < 0.30$	Very Low
$0.30 \leq x < 0.60$	Low
$0.60 \leq x < 0.80$	Medium
$0.80 \leq x \leq 1.00$	High

Table A.13: Impact rating [LAO21]

Parameter Sum (I_{sum})	Impact Rating
$0.00 \leq x < 0.01$	Negligible
$0.01 \leq x < 0.05$	Moderate
$0.05 \leq x < 0.45$	Major
$0.45 \leq x \leq 1.00$	Severe

Table A.14: Risk matrix for Heavens 2.0 [LAO21]

		Impact Rating			
		Negligible	Moderate	Major	Severe
Attack Feasibility Rating	Very Low	1	1	2	3
	Low	1	2	3	4
	Medium	2	3	4	5
	High	2	4	5	5

Table A.15: Attack feasibility parameter values [LAO21]

Expertise	**Value**	**Knowledge of Item**	**Value**	**Window of opportunity**	**Value**	**Equipment**	**Value**
Multiple Experts	0	Critical	0	Small	0	Multiple Bespoke	0
Expert	1	Sensitive	1	Medium	1	Bespoke	1
Proficient	2	Restricted	2	Large	2	Specialized	2
Layman	3	Public	3	Unlimited	3	Standard	3

Table A.16: Sub-Parameter window of opportunity [SBT18] based on Tables A.17 & A.18

	Physical 1 comp. dissambly	**Physical 2** comp. access	**Physical 3** no dissambly	**Remote 1** proximity	**Remote 2** anywhere
Rare	Small	Small	Small	Small	Medium
Sporadic	Small	Small	Small	Medium	Large
Frequet	Small	Small	Medium	Large	Unlimited
Unlimited	Small	Medium	Large	Large	Unlimited

Table A.17: Levels for access means [SBT18]

Level	Explanation	Examples
Physical 1 - component disassembly	Some disassembly of a vehicle component with electronic tools is needed	Any type of low level physical access to read or control a components state, such as attaching a hardware debugger to an electronic control unit (ECU), using a flash reader, etc.
Physical 2 - component access	Some disassembly of the vehicle body with physical tools is needed	Installation or replacement of components, or attaching to a network bus that is otherwise unreachable
Physical 3 - no disassembly	Physical access to the vehicle interior or exterior is needed	Connecting to the OBD-II port, NFC, USB, etc.
Remote 1 - vehicle proximity	Access to a local vehicle network is needed	Bluetooth, Wi-Fi, wireless sensors, V2X, etc.
Remote 2 - anywhere	Remote Internet or telecommunication access is needed	Remote access through the telecommunication network or an external access point

Table A.18: Levels for sub-parameter asset exposure time [SBT18]

Level	Explanation	Examples
Rare	A single rare moment of exposure that cannot be triggered by the attacker	Factory programming of a specific component, installation of a new component in a workshop, pairing of immobilizer and key fob, etc.
Sporadic	A sporadic moment of exposure that cannot be triggered by the attacker	Certain start-up events, sporadic incoming remote connections, diagnostic tests, infrequent state transitions, etc.
Frequent	A frequent moment of exposure that cannot be triggered by the attacker	Vehicle functions that are typically active, such as specific infotainment applications, normal operational states for ECUs, etc.
Unlimited	An unlimited moment of exposure, or one that can be triggered by the attacker	Vehicle functions that are always active or can be activated by an attacker, such as sensors, Bluetooth receivers, wireless gateways, diagnostics servers, etc.

A.8 Service-oriented Device Connectivity (SDC)

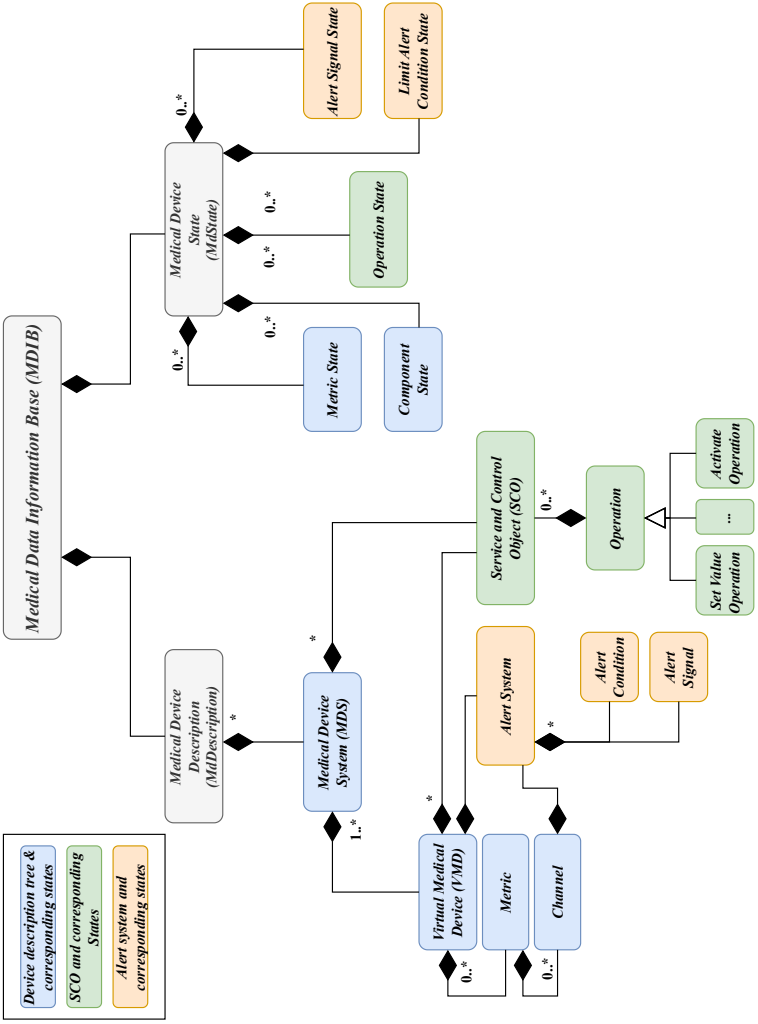


Figure A.31: Medical Device Information Base (MDIB) structure based on [KSA+18] and [IEE18]

A.8.1 SDC Service Composition

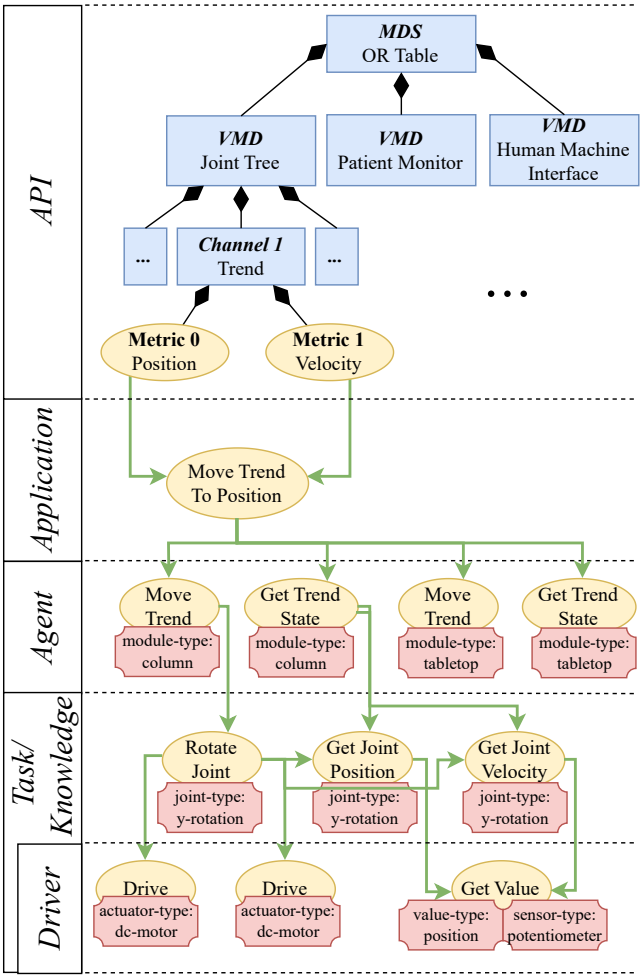


Figure A.32: SDC example for service composition based on Fig. 3.20

A.8.2 Joint Tree SDC

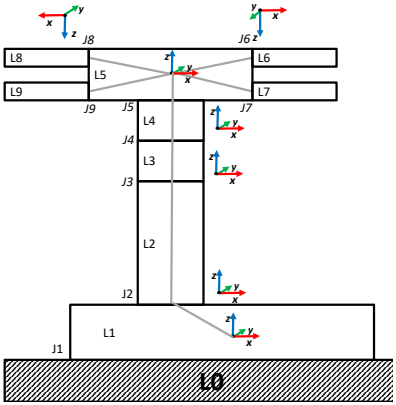


Figure A.33: Connectivity graph (Chapter A.6.1) for an OR table with local coordinate systems of the joints

Joint ID	Name
J1	Traction Drive
J2	Height
J3	Trendelenburg
J4	Tilt
J5	Longitudinal Shift
J6	Back Right
J7	Back Left
J8	Leg Right
J9	Leg Left

Figure A.34: Proposal for a generic OR table joint naming convention (Chapter 2.6, Fig. 2.24)

Since the examined OR tables have no kinematic loops (Chapter 2.4), only parent joints must be communicated to determine the whole structure by a connected device. In addition, each joint is connected to a link called the parent joint of the link, which undergoes the same movement as its parent joint. Thus, for connected devices, which joint moves that link by following the parent joints backward is determinable.

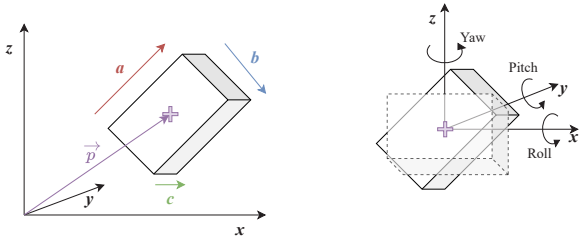


Figure A.35: Description of the collision box of a link

The corresponding link is described by its collision shape and current origin, whereas each link is represented as a box. Other shapes, such as cylinders or shapes describing the exact geometry of the link, are also considerable but need further investigation as they introduce additional possible combinations. By providing the position of its origin $\overrightarrow{p}$ and the rotation (roll, pitch, yaw) as well as its dimension (Fig. A.35), the collision box can be fully described (Requirement 16).

A.9 SOA Applications and Technologies

Table A.19: Comparison of criteria for the life cycle of different architecture paradigms [SSG+22]

Phase	Paradigm	Incrementality	Variant Management	Compatibility	Security (IAM)	Portability	Reconfigurability
Design	Signal-based Service-oriented	⇓	⇓⇓	⇓⇓	⇓	⇓⇓	⇓⇓
Implementation	Signal-based Service-oriented	⇒	⇓	⇒	⇑	⇓⇓	⇓⇓
Testing	Signal-based Service-oriented	⇑	⇑	⇓	⇑⇑	⇓⇓	⇑⇑
Operation	Signal-based Service-oriented	⇓⇓	⇓⇓	⇓⇓	⇓	⇓⇓	⇓⇓

⇑⇑ Significant advantage for signal-based paradigm
⇑ Advantage for signal-based paradigm
⇒ equal
⇓ Advantage for service-oriented paradigm
⇓⇓ Significant advantage for service-oriented paradigm

The trend of manufacturer-independent interoperability and open standards to further decouple software and hardware is noticeable in several industries coping with Cyber-Physical Systems (CPSs). Furthermore, the industries presented can be the transition from self-contained systems to connected and interoperable devices with plug-and-play features. While the automotive industry has already relied on open standards for several decades, the automation industry has overcome the manufacturer-dependent protocols later. This trend holds with Open Platform Communications Unified Architecture (OPC UA) being on the rise since its publication in 2006 [Lan21].

A.9.1 SOA in Automotive Industry

AUTOSAR is standardized by a consortium of car manufacturers, suppliers, software vendors, and hardware vendors. Meanwhile, it includes aspects of software architecture, such as a software stack, a communication middleware, or a metamodel [VS21, Sta17].

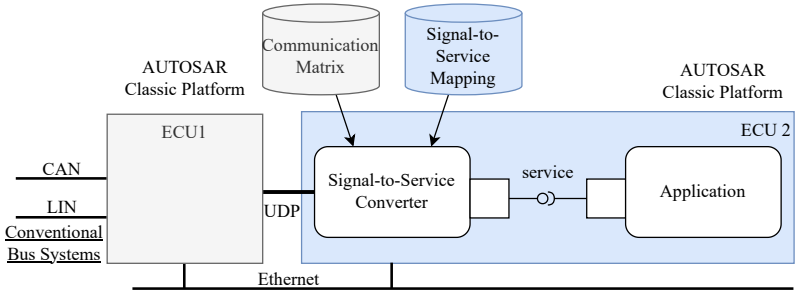


Figure A.36: Connection to an AUTOSAR Classic signal gateway [Tis18]

With the extension of AUTOSAR classic, which is a representative of signal-based architectures, to AUTOSAR Adaptive, new functions such as automated driving and Over The Air (OTA)-updates are addressed. In addition, the application of Ethernet in modern cars results in increased bandwidth, enabling new communication paradigms such as SOA. [Sto21] Modern cars use a mixed architecture based on Ethernet in the so-called backbone for service-oriented communication and signal-based communication in the periphery based on protocols such as CAN, Local Interconnect Network (LIN) but also Ethernet [SGG+20].

Since SOA has been considered a suitable paradigm in the automotive sector [SSG+22], AUTOSAR Adaptive has been developed to support service-oriented communication based on SOME/IP and DDS (Chapter A.9.4). While classic AUTOSAR is based on OSEK Operating System (OS)[1], which relies on C, AUTOSAR Adaptive is Portable Operating System Interface (POSIX) based and is also capable of C++ [AUT19].

[1] by the *Offene Systeme und deren Schnittstellen für die Elektronik im Kraftfahrzeug (OSEK) (eng. Open Systems and their Interfaces for the Electronics in Motor Vehicles)* standards body.

A.9.2 SOA in Automation Engineering

The fourth industrial revolution, *Industry 4.0* [Bab21], led to a technological change forced by requirements such as decentralized tasks and function distribution [KS15]. Therefore, service orientation seems to be an appropriate approach here as well, as can be seen in the proposed middleware in Glock et al. [GBK+19] and the success of the service-oriented OPC UA protocol, which is a part of the IEC62541 standard series [Lan21].

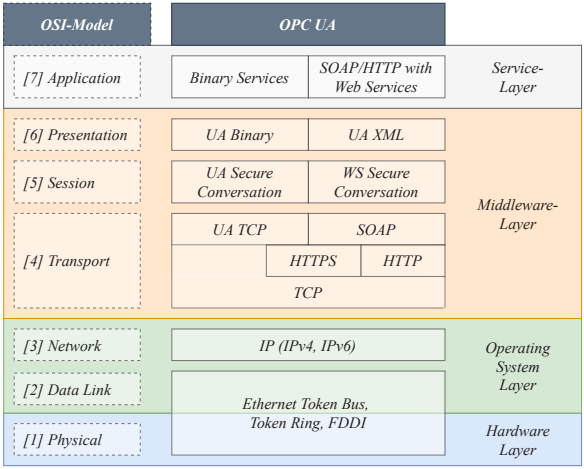


Figure A.37: OPC UA ISO/OSI reference [Bab21]

OPC UA is one of the most advanced protocols in automation engineering and is published by the Open Platform Foundation (OPF) [OPC15] as an open standard for a platform-independent SOA. It enables machine-to-machine and Programmable Logic Controller (PLC)/Supervisory Control and Data Acquisition (SCADA)/Human-Machine-Interface (HMI)/Manufacturing Execution System (MES)-to-machine communication by making machine data such as control parameters transportable and machine-readable in a semantic way. Furthermore, OPC UA is also based on Ethernet and the OSI model (Fig. A.37) and provides semantic interoperability. [Bab21] The OPC UA architecture is based on a client-server structure [Lan21] and extended with a publish/subscribe specification enabling data exchange without knowing the origin of that data [PTD+19].

A.9.3 SOA in Robotics

SOA is a trend in robotics [CDGA10], and as [Rob11] pointed out in 2011 already, the robot frameworks Microsoft Robotics Developer Studio (MRDS) and Robot Operating System (ROS) both rely on an SOA. In the meantime, Microsoft stopped MRDS, predominantly in 2015 [Bue15], but ROS even introduced a new version with ROS2, which still follows the SOA trail. Furthermore, the ArmarXRT software framework developed by the H2T of the Karlsruhe Institute of Technology (KIT) also relies on the abstraction of the communication by a middleware layer [PK22]. Today, ROS/ROS2 is becoming the de facto standard for advanced robotic systems [Kou21].

Although the name suggests it, ROS is not an OS in the conventional sense, as it does not offer features such as process management and scheduling. Moreover, it provides a communication layer between different applications in a heterogeneous system [QCG+09]. The first version of ROS focused on the research for mobile robots supported by a performant workstation and network infrastructure without the necessity for real-time [Ger22]. ROS2 also abandons this research-centric approach to be applicable in industrial applications. Disadvantages such as the lack of real-time capability have been addressed, and it can operate without workstations and with an unreliable network infrastructure [Sto21].

ROS2 uses DDS (Chapter A.9.4) as middleware and thus does not rely on its ROS-specific middleware anymore as the previous version did (Fig. A.38). Furthermore, the ROS master from the first version has been abandoned in ROS2. These changes allow the ROS 2 nodes to find their services decentral, making them more robust as there is no single point of failure. Therefore, the second version is more promising for safety-critical systems. Furthermore, using cross-platform middleware such as DDS, ROS 2 runs on Linux and on other platforms like Windows, Mac, or other POSIX compatible OS. In addition, with Micro-ROS [ROS22], there also exists a micro-controller version for POSIX-based Real-Time Operating System (RTOS) such as FreeRTOS or Zephyr compatible with ROS2 [ROS22].

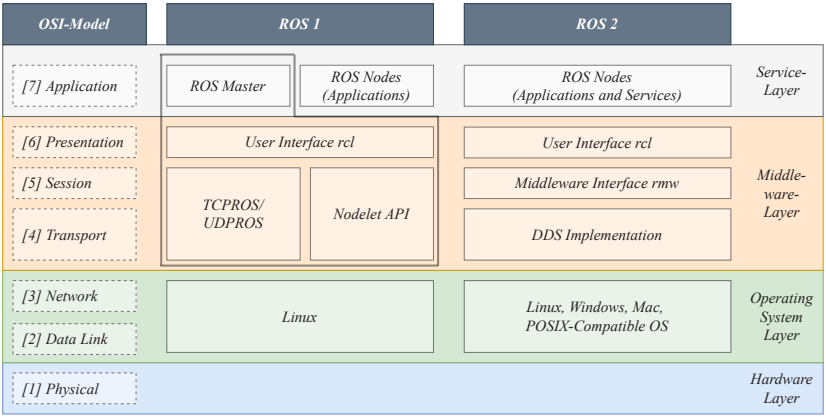


Figure A.38: ROS and ROS 2 layer architecture according to [Sto21] based on [MKA16]

Oliveira also points out that SOAs are attracting increasing attention because of their flexibility, reusability, and integrability in robotic applications. He also systematically reviews SOA systems in robotics [OON13] [Bue15]. For automatic cataloging and discovery of services, Oliveira also proposes a taxonomy for services based on a layered service architecture for robotic applications (Fig. A.39).

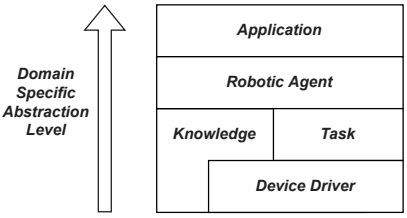


Figure A.39: Robotics services dependency stack [Bue15]

The ROS2-based Plug-and-Play approach by Stoll et al. [SGS+21b] is designed to connect different services dynamically and is built on a service description contained in a YAML Ain't Markup Language (YAML) file. When a service is started and successfully configured, it enters an inactive state while it broadcasts its capabilities and seeks services that can fulfill its dependencies. When another service in the ROS2 networks provides a matching dependency or capability, both services switch to an active state and begin exchanging data. When the dependencies of a service are no longer satisfied, it enters the inactive state again to reduce unnecessary network traffic [SGS+21b] [SGLS22].

A.9.4 SOA Middlewares

Middlewares generally abstract the communication of different components of a distributed system [PTD+19] (Chapter 2.5.1). About the ISO/OSI model, middlewares for SOA can be allocated to layers 4 (transport), 5 (session), and 6 (presentation) (Fig. 2.14). A benefit of this abstraction level is that developers can focus on data and the requirements according to Quality of Service (QoS) (e.g., ROS2 [Ope22a] or DDS from RTI Connext [Rea15]) because the middleware handles these [APG18]. Otherwise, the transfer of the data and ensuring QoS must be considered, too, which increases the challenges and efforts during development. QoS is a central aspect of modern SOA middleware and describes a set of configurable parameters that a service must hold. In real-time applications, this can be, for example, that specific deadlines must be held, e.g., that the response of a service must be received within 100 ms. In particular, these QoS contribute to the fact that, from a research perspective, middleware is considered a substantial challenge for autonomous systems [YL21].

Ungurean et al. [UGG16] state that DDS, OPC UA, and Message Queue Telemetry Transport (MQTT) belong to the most essential standards in terms of the Internet of Things (IoT). Profanter et al. [PTD+19] focus on the most widely used protocols for Industry 4.0, namely OPC UA, DDS, MQTT, and ROS. Furthermore, as AUTOSAR adaptive relies on SOME/IP, it is also considered a vital middleware, although it is focused on automotive applications. The middlewares discussed in the following do not represent a complete list of available SOA middlewares since there are more examples, such as the already mentioned MQTT [OAS19b], RESTful Web Services [OAS19a] or the *embedded Service-Oriented Communication (eSOC)* protocol outlined by Wagner et al. [WSP16] that is based on CAN (Chapter A.3.1). A comprehensive comparison for automotive relevant protocols and middlewares can be found in [RGKS20]. For elaborate middleware comparisons in industrial applications, see [PTD+19] [UGG16] [Lan21].

Data Distribution Service

DDS is a standard defined by the Object Management Group (OMG) for communication systems. Its goal is to transmit real-time data efficiently, reliably, and robustly in an IP-based network. [Lan21] It allows applications to provide or retrieve data in a publish/subscribe communication pattern by matching publishers to subscribers via so-called *topics*. The middleware manages the distribution of the system state to the different heterogeneous components. This is especially helpful in systems where the individual components need access to a shared data model. [APG18] This shared data model is also referred to as *Global Data Space* [Lan21] (Fig. A.40) and represents a common system state between the participants, making DDS a data-centric middleware [PTD+19]. The targeted application areas of DDS are real-time systems with functional safety and low latencies [Lan21].

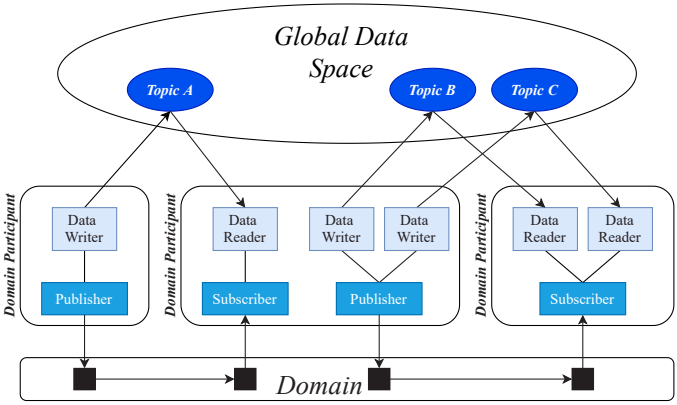


Figure A.40: Conceptional structure of DDS according to [Lan21]

The DDS data writer communicates the data to other subscribers who have subscribed to the corresponding topic [APG18]. The discovery of services is fully decentralized [Sto21], and supported communication patterns are publish-subscribe, fire & forget, and R/R. DDS is based on TCP/IP and User Datagram Protocol (UDP) and provides security features such as authentication, encryption, access control, and security tagging [RGKS20].

The German Aerospace Center DLR provides a medical example using DDS with *MiroSurge* [DLR21] [HKT+10] (Chapter A.1.3). Here, the researchers were able to synchronize the movements of the individual systems with the patient's heartbeat to enable surgeries while the heart still beats [DN19]. Thus, for the surgeon, it seems like the heart is standing still, improving the treatment possibilities. Furthermore, it shows that with DDS for medical applications, it is possible to enable deterministic communication between 1 and 3kHz in a safety-critical environment using a service-oriented middleware.

SOME/IP

SOME/IP, first proposed by the BMW Group in 2011, is a central part of AUTOSAR Adaptive and focused on the automotive domain [GD20]. Like SDC, OPC UA, and DDS, SOME/IP is located in layers 5 to 7 (Fig. A.41) in the ISO/OSI reference model (Chapter 2.5.1). SOME/IP is function-oriented, which means it focuses on services being functions, which are provided rather than the data exchange. Thus, it is comparable rather to OPC UA than to DDS.

Supported communication patterns are publish-subscribe, fire & forget, and request-response (R/R) [RGKS20]. Unlike DDS, SOME/IP currently does not provide security measures such as encryption or access control, which is addressed by Iorio et al. [IBR+20]. The service discovery is not directly part of SOME/IP but part of the SOME/IP Service Discovery (SOME/IP-SD) protocol. It mediates between service providers and consumers by determining the IP addresses and port numbers of the individual services [GD20]. Unlike DDS, SOME/IP discovers services by a broker (Chapter 2.5.1), which is a disadvantage in terms of safety and security [RGKS20] as a single point of failure.

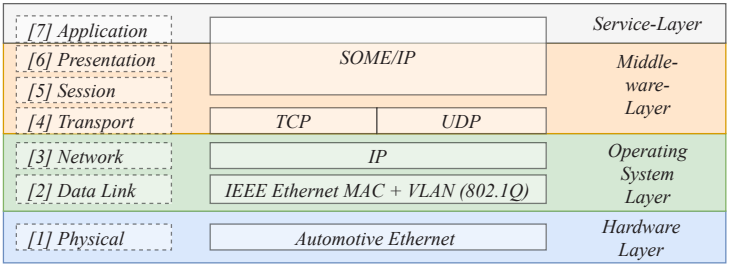


Figure A.41: ISO/OSI layer model (Fig. 2.14) for SOME/IP based on [GD20]

Table A.20: Comparison of SOA middlewares [Sto21]

Property	DDS & RTPS	SOME/IP
Origin	Whole industry	Automotive industry
Communication	Discovery mechanism allows individual participants to identify available services at runtime. Consumers can access data and services directly without intermediaries (peer to peer). No customization to specific service version necessary.	Object-oriented communication, consumers must access predefined service classes and must therefore be adapted to these at development time.
API	APIs standardized for various programming languages or part of manufacturer-specific DDS implementations	Standard API is not specified, but is defined in AUTOSAR Classic instead.
Network	Uses RTPS, which abstracts from TCP and UDP, but also others such as shared memory, in which a shared memory area is accessed. In addition, further transport protocols can be added and features such as QoS can still be used.	Network protocols TCP (connection-oriented, more reliable) and UDP (connectionless, less overhead) are supported. No extension by own transport protocols.
Security	Can use the transport encryption of the underlying transport protocol, but offers security mechanisms independent of these.	Transport encryption (e.g. Transport Layer Security for TCP) is used. There are no separate security mechanisms.
QoS	Several different QoS policies are supported. [Sto21]	Only one reliability policy is supported, based on TCP.
Used in	AUTOSAR Adaptive, ROS2	AUTOSAR Classic (via Transformer), AUTOSAR Adaptive

A.10 Functional Requirements

Use cases are considered here for further examination to structure and derive functional requirements for OR tables in a connected OR environment (Fig. A.42) for different users. The clinical staff, in general, is interested in positioning the OR table and preparing it for surgery, while a service technician also needs to perform maintenance and software updates. With increased connectivity, other connected devices are interested in the information provided by the OR table, such as its current position (Chapter A.1.2) or the positioning of the patient. Furthermore, backend systems enable remote data retrieval and maintenance.

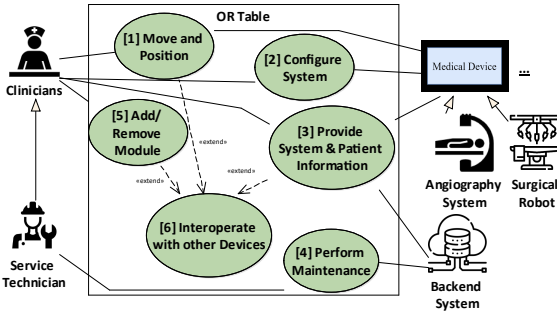


Figure A.42: Use cases for a cyber-physical, interoperable and modular OR table (using surgery icons designed by Linector from Flaticon)

By considering these users, the following use cases for an OR table can be derived:

UC 1 Move and Position: Functions related to the positioning (Chapter A.10.1, Fig. A.44)

UC 2 Configure System: Settings such as motion speed or locking to prevent unintentional motions (Chapter A.10.2, Fig. A.45)

UC 3 Provide System & Patient Information: Information related to the OR table, such as the current position, as well as gathered patient information, such as its orientation or weight (Chapter A.10.3, Fig. A.46)

UC 4 Perform Maintenance: Maintenance such as calibration of the joint position sensors or software update (Chapter A.10.4, Fig. A.47)

UC 5 Add/Remove Modules: Exchange of modules such as accessories or tabletops (Chapter A.10.5, Fig. A.48)

UC 6 Interoperate with other Devices: Interactions with other connected devices (Chapter A.10.6, Fig. A.49)

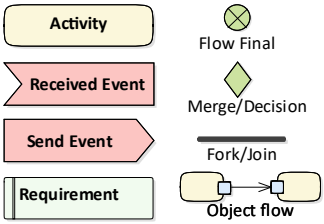


Figure A.43: Activity diagram legend [Spa24]

The needs of the surgeon, the anesthesiologist, and the patient indirectly define the fundamental requirements of an OR table (Chapter 2.6). They can be summarized according to [KAKA06] with positioning and transporting the patient without harm (Requirements 1, 2, 3, 4, 5, 8, 54). To fulfill the core functionality (functions 1 & 2) and its intended use (Chapter 2.6), it must allow the individual motion of its joints (Requirements 1, 2, 3, and 4) with a desired velocity (Requirement 12).

A.10.1 Use Case 1 - Move and Position

Requirement 1 - Lift/Lowering Patient: The system must be able to adjust the patient's height to allow ergonomic work.

Requirement 2 - Slant Patient (Trendelenburg): The system must always allow the patient to be slanted in the event of health-threatening events.

Requirement 3 - Tilt Patient: The system must provide a clear view of the body cavity and allow organs to be positioned by tilting the patient during minimally invasive procedures.

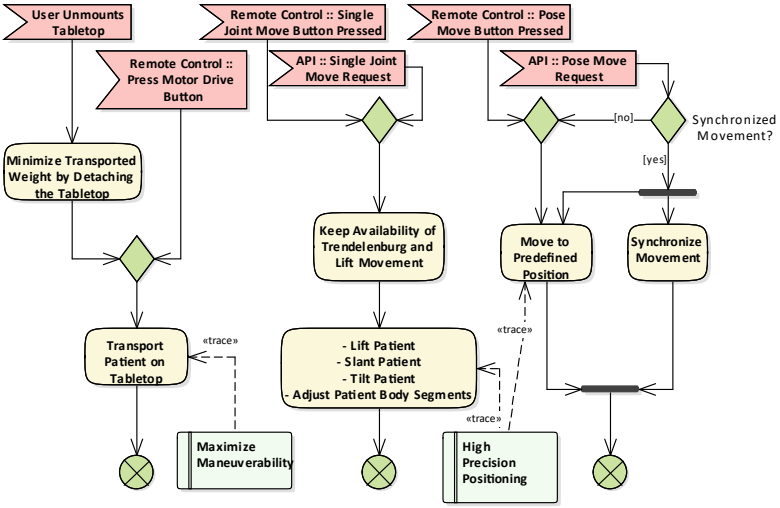


Figure A.44: Activity diagram as requirement specification for use case *Move and Position*

Requirement 4 - Adjust Patient Body Segments: The system must allow the patient's body to be bent into anatomically correct positions and the extremities to be positioned as required for the operation (e.g. bending, spreading).

Requirement 5 - Transport Patient on Tabletop: The system must allow the patient to be transported without removing them from the patient board.

Requirement 6 - Minimize Transported Weight by Detaching the Tabletop: The system must minimize the transported weight to improve transportability of the patient by detaching the tabletop.

Requirement 7 - Maximize Maneuverability: The system should be maneuvered easily by the hospital staff especially in case a patient is on the tabletop.

Requirement 8 - Keep Availability of Trendelenburg and Height Movement: The system must always provide Trendelenburg (Requirement 2) and height movements (Requirement 1).

Requirement 9 - Move to Predefined Position: The system shall move autonomously to a predefined position.

Requirement 10 - Synchronize Movements: The system shall coordinate synchronized movements autonomously.

Requirement 11 - High Precision Positioning: The system must allow to be positioned with high precision.

A.10.2 Use Case 2 - Configure System

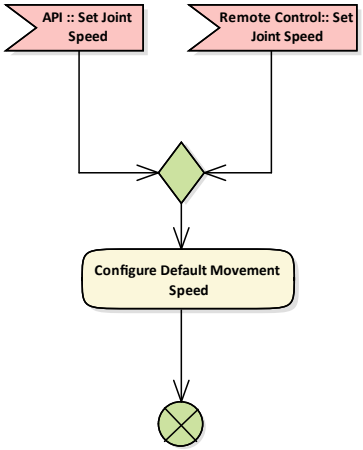


Figure A.45: Activity diagram as requirement specification for use case *Configure System*

Requirement 12 - Configure Default Movement Speed: The system should be able to modify the velocity via an external interface.

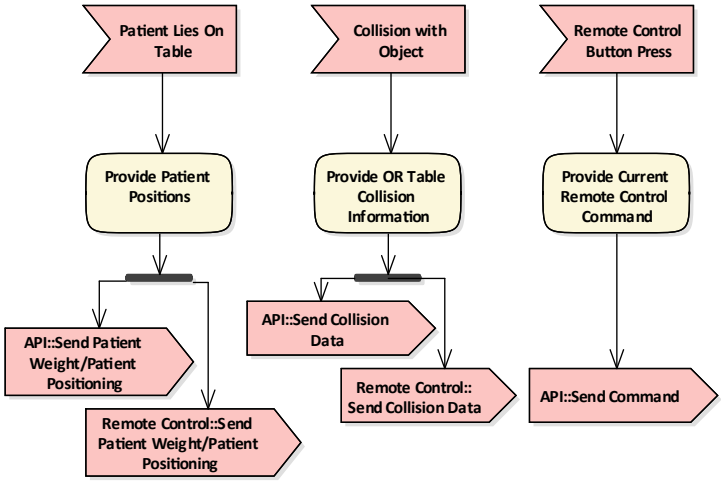


Figure A.46: Activity diagram as requirement specification for use case *Provide System & Patient Information*

A.10.3 Use Case 3 - Provide System & Patient Information

Requirement 13 - Provide Joint Positions: The system must provide the joint positions to other devices.

Requirement 14 - Provide Patient Weight: The system should provide the determined patient weight to other devices.

Requirement 15 - Provide Patient Positioning: The system should provide the determined patient positioning to other devices.

Requirement 16 - Provide Collision Information: The system must provide its current geometry based on its configuration and position to other devices.

Requirement 17 - Provide Remote Control Command: The system should provide the current remote control command to other devices.

A.10.4 Use Case 4 - Perform Maintenance

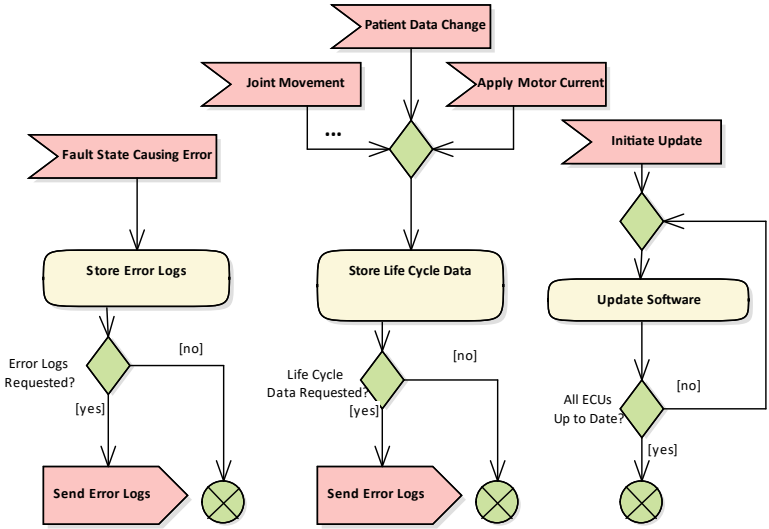


Figure A.47: Activity diagram as requirement specification for use case *Perform Maintenance*

Requirement 18 - Store Error Logs: The system should store potential errors persistently and provide them on demand.

Requirement 19 - Store Life Cycle Data: The system should store life cycle data of the system and provide it on demand.

Requirement 20 - Update Software: The system must be able to update the software on each ECU.

A.10.5 Use Case 5 - Add/Remove Modules

Requirement 21 - Exchange Module: "The system must be adaptable to different (surgical) disciplines by adding or removing OR table modules" [PVR$^+$22].

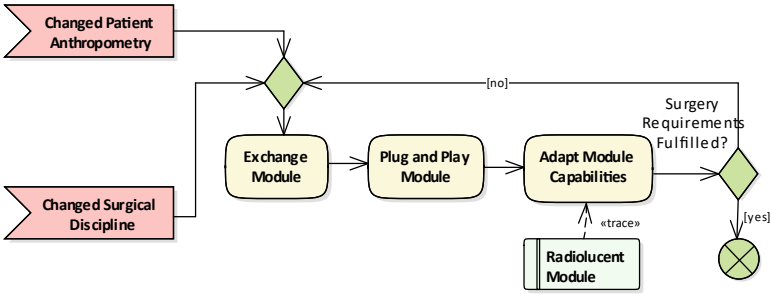


Figure A.48: Activity diagram as requirement specification for use case *Add/Remove Modules*

Requirement 22 - Adapt Module Capabilities: The system must consider the capabilities geometry, movable joints and mounting points of mounted OR table modules and adapt the provided functionality accordingly [PVR+22].

Requirement 23 - Radiolucent Module: The system must provide the ability to trans-illuminate the patient while working with the x-ray image intensifier to be able to adapt to image guided surgery (imaging procedures during surgery) by providing (fully) radiolucent modules.

Requirement 24 - Plug and Play Module: The system must allow connection and interoperability of different modules at run time.

A.10.6 Use Case 6 - Interoperate with other Devices

Requirement 25 - Plug and Play Device: "The system must allow connection and interoperability with other medical devices at run time [DH12]" [PVR+22].

Requirement 26 - Control via Connected Device: The system should be controllable via the human-machine interface of another device [CVK+18].

Requirement 27 - Show Connected Device Information: The system should show information from other systems on its HMI.

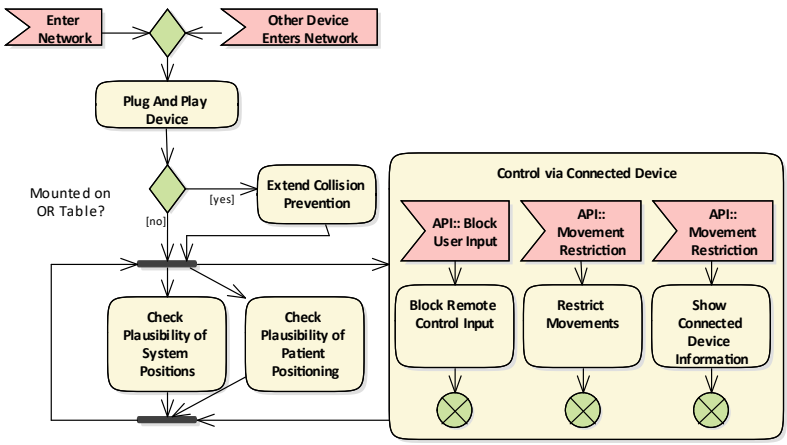


Figure A.49: Activity diagram as requirement specification for use case *Interoperate with other Devices*

Requirement 28 - Check Plausibility of Patient Positioning: The system should use the information on patient positioning from other devices to check the plausibility of the positioning of the patient.

Requirement 29 - Check Plausibility of System Positions: The system should use the position information of other devices to check the plausibility of its positions.

Requirement 30 - Extend Collision Prevention: The system should use the positions and bounding boxes of mounted external devices for collision prevention during movements.

Requirement 31 - Block Remote Control Input: The system must lock or unlock the remote control input when requested by other devices.

A.10.7 OR Table Functions

Function 1 - Move Patient Body Parts: This function includes features that allow adapting or adjusting individual patient body parts. This encompasses positioning, rotation, or manipulation to achieve proper alignment or accessibility for medical procedures (Requirements 4, 11, 13, 30, 33, 47).

Function 2 - Move Patient: The *Move Patient* function in a medical device or software includes features related to moving the patient's body as a whole with the same aims as a function 1 (Requirements 1, 2, 3, 5, 7, 8, 11, 13, 33).

Function 3 - Move To Predefined Pose: The aim of this function is the precise movement of the patient to a previously defined position. It is based on the individual motions (functions 1 & 2) and includes positions stored by the user (Requirement 9).

Function 4 - System Interoperability: The *System Interoperability* function covers features related to interoperability with other systems. It includes the ability to communicate with medical devices (Requirements 10, 16, 17, 25, 26, 27, 31, 39, 51, 53, 56, 57, 58).

Function 5 - Exchange Modules: The function aims to encompass all the features related to module exchange, allowing for the customization of the OR table according to specific procedures and patient needs (Requirements 6, 7, 21, 22, 23, 24, 40, 45, 48, 50, 53, 54, 55, 56).

Function 6 - Patient Data Determination: All features related to patient data determination are covered by the function *Patient Data Determination* (Requirements 14, 15, 32).

Function 7 - Collect System Diagnose Data: This function is related to all system diagnosis data features that may include the ability to monitor and diagnose issues related to the performance and operation of the device or software (Requirements 18, 19, 35, 44).

Function 8 - Update: The *Update* function relates to features allowing for the timely and secure delivery of software updates and patches, essential for ensuring the ongoing performance, functionality, and security of the device or software (Requirements 20, 34, 48, 52).

A.11 Non-Functional Requirements

The non-functional requirements are structured according to [SOP21] based on [Rup21] (Chapter A.2.2). The quality metric *functionality* is not considered here because it is already handled within the functional requirements (Chapter A.10).

A.11.1 Safety

Requirement 32 - Ensure Patient Context: The system must ensure that the interoperating device is connected to the same patient.

Requirement 33 - Single Fault Safety: The system must provide single fault safety for functions, which can harm the user or patients.

Requirement 34 - Adaption From Safety Risks: The system must allow fast adaptions to safety risks discovered after deployment.

Requirement 35 - Report any Safety Incidents: The system must collect necessary data to report and detect a safety incident.

Requirement 36 - Detect Dangerous System Situations: The system must detect dangerous system states dependent on the current system configuration and load.

Requirement 37 - Monitor System State Plausibility: The system must detect if the system state is plausible.

Requirement 38 - Monitor Collaborative Movements: The system should detect collaborative movements with a partner system is plausible and have the same reference.

Requirement 39 - Software Safety Classification Restrictions: The system must only use functions with writing behavior of other devices, which have a similar or lower Software Safety Classification (SSC).

A.11.2 Usability

Requirement 40 - Generic Remote Control: The system should be controllable via a generic remote control for different OR tables.

A.11.3 Security

Requirement 41 - Analyze System During Life Cycle: The system should analyze the product data during life cycle for anomalies.

Requirement 42 - Payload Inspection: The system should be able to detect attacks by inspecting the payload [PRGS22].

Requirement 43 - Unknown Attacks: The system must be able to detect unknown attacks. [PRGS22]

Requirement 44 - Report any Security Incidents: The system must collect all necessary data to report and detect a security incident.

Requirement 45 - Effective Segregation of Software Items: The software of the system should be segregated in loosely coupled components.

Requirement 46 - Check Plausibility of Communicated Positions: The system must detect if the communicated link and joint positions are implausible.

Requirement 47 - Maintain Intended Use On Security Incidents: The system must not influence the intended use when reacting to security incidents.

A.11.4 Changeability/Maintainability

Requirement 48 - Future Proof Extension: “The system must be forward compatible to new OR table components to ensure future-proofing over the whole life cycle” [PVR$^+$22].

Requirement 49 - Backwards Compatibility: “The system must be backward compatible to also use legacy components over the whole life cycle” [PVR$^+$22].

Requirement 50 - Keep Intended Use: The system must retain the intended use of its used components.

Requirement 51 - Keep Intended Use During Interoperation: The system must not interfere with its own and the intended use of the connected device, when interacting with other devices.

Requirement 52 - Design for Updatability: “The system must allow fast adaptions to security (and safety) risks discovered after deployment” [PVR$^+$22].

A.11.5 Reliability

Requirement 53 - Data Throughput: “The system must have a data throughput for control data of at least 100 kbit/s at maximum of 10 ms latency and for non-control data of 1 Mbit/s to 10 Mbit/s at 50 ms to 1 for external and internal interfaces ”(Fig. A.21) [PVR$^+$22]

Requirement 54 - Adaption to Patient Physiology: The system should avoid that the patient takes harm from pressure sores by incorrect positioning

Requirement 55 - Reduced Column Foot Interfering: The system should allow a maximized foot space for the hospital staff.

Table A.21: Results for the interface analysis of existing devices [PDDL15]

Data Class	Data Throughput	Max. Tolerable Latencies	Used Interfaces
Control data	< 100kbit/s	10-50ms	CAN, SPI, USB, RS232, Ethernet
Video Streams	< 4Gbit/s	10-300ms	HDMI, VGA, DVI, S-Video, Ethernet
Other (images, parameters, patient data)	< 10Mbit/s	50ms - 1s	USB, RS232, Ethernet

A.11.6 Portability

Requirement 56 - Protocol Dependency: "The system should use protocol independent (SOA-)features for anomaly detection" [PRGS22]

A.11.7 Compatibility

Requirement 57 - Life Cycle overarching Interoperability: The system should be easily connected to other medical systems (also third-party devices) during the whole life cycle to provide interoperability [PVR+22].

Requirement 58 - Unified System Interface: The system must implement a unified standard interface used by other devices.

A.12 Novel Anomaly Detection

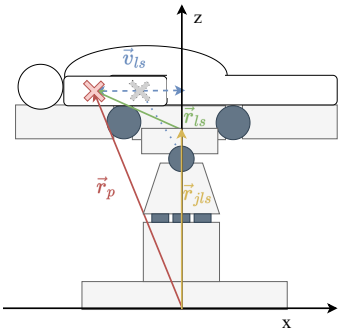


Figure A.50: Longitudinal shift used for the dynamic model

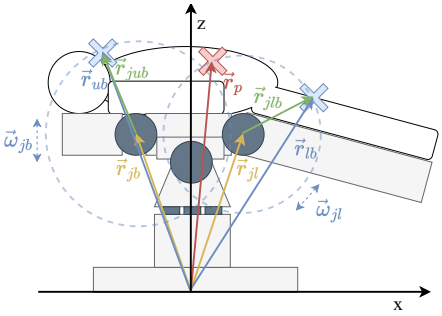


Figure A.51: Patient in supine position with a patient CoG position $\vec{r}_p$ in dependency to upper body distance $\vec{r}_{ub}$, lower body distance $\vec{r}_{ub}$ (Table 4.5)

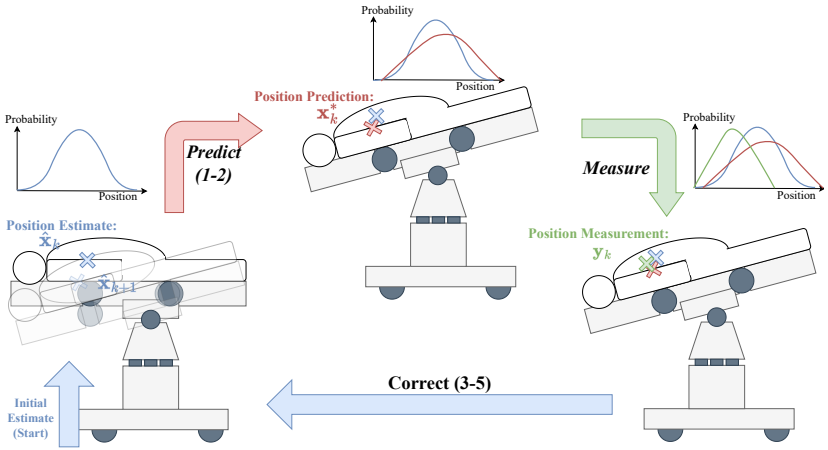


Figure A.52: Position estimation for the patient position $\vec{r}_p$ based on a KF (Fig. 2.7)

A.12.1 Kalman Filter Whole Body

Whole body movement with Trend and tilt joints (Table 4.4):

$$\vec{\omega}_{tb} = \begin{pmatrix} 0 \\ \omega_{tb} \\ 0 \end{pmatrix} \tag{A.10}$$

$$\vec{v}_{tb} = \begin{pmatrix} 0 \\ \omega_{tb} \\ 0 \end{pmatrix} \times \begin{pmatrix} r_{x,tb} \\ r_{y,tb} \\ r_{z,tb} \end{pmatrix} = \omega_{tb} \begin{pmatrix} r_{z,tb} \\ 0 \\ -r_{x,tb} \end{pmatrix} \tag{A.11}$$

with

$$|\vec{\omega}_{tb}| = \omega_{tb} \tag{A.12}$$

$$\vec{\omega}_{tl} = \begin{pmatrix} \cos(\varphi_{tb}) & 0 & \sin(\varphi_{tb}) \\ 0 & 1 & 0 \\ -\sin(\varphi_{tb}) & 0 & \cos(\varphi_{tb}) \end{pmatrix} \cdot \begin{pmatrix} \omega_{tl} \\ 0 \\ 0 \end{pmatrix} \tag{A.13}$$

$$\vec{v}_{tl} = \begin{pmatrix} \omega_{tl}\cos(\varphi_{tb}) \\ 0 \\ -\omega_{tl}\sin(\varphi_{tb}) \end{pmatrix} \times \begin{pmatrix} r_{x,tl} \\ r_{y,tl} \\ r_{z,tl} \end{pmatrix} = \omega_{tl} \begin{pmatrix} \sin(\varphi_{tb})\cdot r_{y,tl} \\ -\sin(\varphi_{tb})\cdot r_{x,tl} - \cos(\varphi_{tb})\cdot r_{z,tl} \\ \cos(\varphi_{tb})\cdot r_{y,tl} \end{pmatrix} \tag{A.14}$$

with

$$|\vec{\omega}_{tl}| = \omega_{tl} \tag{A.15}$$

$$\vec{v}_{ls} = \begin{pmatrix} \cos(\varphi_{tb}) & 0 & \sin(\varphi_{tb}) \\ 0 & 1 & 0 \\ -\sin(\varphi_{tb}) & 0 & \cos(\varphi_{tb}) \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi_{tl}) & -\sin(\varphi_{tl}) \\ 0 & \sin(\varphi_{tl}) & \cos(\varphi_{tl}) \end{pmatrix} \cdot \begin{pmatrix} v_{ls} \\ 0 \\ 0 \end{pmatrix} \tag{A.16}$$

$$\vec{v}_{ls} = v_{ls} \cdot \begin{pmatrix} \cos(\varphi_{tb}) \\ 0 \\ -\sin(\varphi_{tb}) \end{pmatrix} \tag{A.17}$$

A.12.2 Kalman Filter Partial Body

Partial body movement of lower body with leg joints (Table 4.5):

$$\vec{v}_{lb} = \begin{bmatrix} 0 \\ \omega_{jl}\cos(\varphi_{tl}) \\ \omega_{jl}\sin(\varphi_{tl}) \end{bmatrix} \times \begin{bmatrix} r_{jlb,x} \\ r_{jlb,y} \\ r_{jlb,z} \end{bmatrix} = \omega_{jl} \begin{bmatrix} \cos(\varphi_{tl})\cdot r_{jlb,z} - \sin(\varphi_{tl})\cdot r_{jlb,y} \\ \sin(\varphi_{tl})\cdot r_{jlb,x} \\ -\cos(\varphi_{tl})\cdot r_{jlb,x} \end{bmatrix} \tag{A.18}$$

with

$$|\vec{\omega}_{jl}| = \omega_{jl} \tag{A.19}$$

$$\vec{\omega}_{jl} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi_{tl}) & -\sin(\varphi_{tl}) \\ 0 & \sin(\varphi_{tl}) & \cos(\varphi_{tl}) \end{bmatrix} \cdot \begin{bmatrix} 0 \\ \omega_{jl} \\ 0 \end{bmatrix} \tag{A.20}$$

Partial body movement of upper body with back joints (Table 4.5):

$$\vec{\tilde{v}}_{ub} = \begin{bmatrix} 0 \\ \omega_{jb}\cos(\varphi_{tl}) \\ \omega_{jb}\sin(\varphi_{tl}) \end{bmatrix} \times \begin{bmatrix} r_{jub,x} \\ r_{jub,y} \\ r_{jub,z} \end{bmatrix} = \omega_{jb} \begin{bmatrix} \cos(\varphi_{tl})\cdot r_{jub,z} - \sin(\varphi_{tl})\cdot r_{jub,y} \\ \sin(\varphi_{tl})\cdot r_{jub,x} \\ -\cos(\varphi_{tl})\cdot r_{jub,x} \end{bmatrix} \quad \text{(A.21)}$$

with

$$|\vec{\tilde{\omega}}_{jb}| = \omega_{jb} \quad \text{(A.22)}$$

$$\vec{\tilde{\omega}}_{jb} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\varphi_{tl}) & -\sin(\varphi_{tl}) \\ 0 & \sin(\varphi_{tl}) & \cos(\varphi_{tl}) \end{bmatrix} \cdot \begin{bmatrix} 0 \\ \omega_{jb} \\ 0 \end{bmatrix} \quad \text{(A.23)}$$

A.12.3 Distributed SDC Anomaly Detection

Table A.22: Exemplary role-based IAM in OR networks (X: control allowed, 0: control restricted)

Controlling Device → **Controlled Device ↓**	**Angiography system**	**OR table**	**OR Lights**	**Ventilator**	**OR table Remote Control**
Angiography system	X	X	X	X	0
OR table	0	X	X	0	X
OR Lights	0	0	0	0	0
Ventilator	0	0	0	0	0
OR table Remote Control	0	X	X	X	0

A.12.4 Embedded Anomaly Detection in a State-of-The-Art System and Limitations

Using a proprietary OR table prototype equipped with a load recognition system (Chapter 2.6.4), the dynamic checks are implemented to examine potentials and limitations in the current state-of-the-art OR table, especially regarding

data collection. The system uses a signal-based architecture (Chapter 3.4.4) using XMC4700 [Inf23] Microcontrollers. Furthermore, the approach presented in Chapter 4.3 must be adapted due to the non-linearities to be runable on the desired test platform. As input, the joints' velocities are derived directly from the position sensors. The velocity of the patient's CoG can be determined with the sum (Chapter 4.3.2):

$$\vec{v}_p = \vec{\omega}_{tl/tb} \times \vec{r} + \vec{v}_{ls} \tag{A.24}$$

The transition matrix $\mathbf{A}$ then can be determined as a linear model with constant velocity for a three DoF point:

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & T & 0 & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & 0 & 0 & T \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{A.25}$$

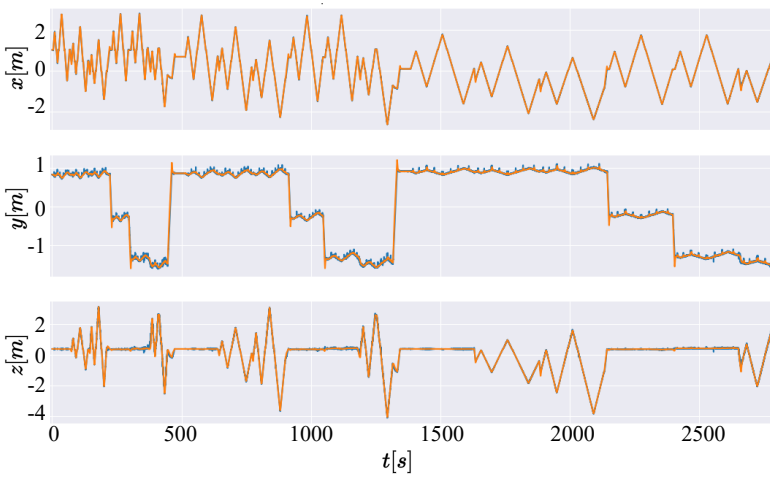


Figure A.53: Comparison between reduced Kalman Filter estimation (orange) running embedded and measurement (blue) with a weight of sandbags with approximately 100kg

The curve progression of measurement and estimation in each of the axes are similar (Fig. A.53), and the Kalman Filter smoothes the noise of the measurement. Yet, an educated statement on real-world data is intricate as the ground truth, which refers to the authentic, real-world data used as a reference or benchmark

for evaluating the accuracy and performance of algorithms, is missing. Although a measurement system is in place, the data underlies measurement errors resulting, for example, from noise and calculation errors due to deformations not encompassed by the system. Therefore, this approach does not provide a basis for a quantifiable comparison of different algorithms. In addition, an anesthetized patient's body behaves differently from the body of an awake person, and obese patient bodies are particularly relevant here, which raises ethical concerns, so the ethics committee must be consulted. Thus, simulation is necessary in the first step.

A.12.5 Comparison Simulation Tools

Table A.23: Simulation tool comparison according to [Lab22] (Table A.24)

Tool	Realtime	Costs	Implementation Effort	Widely Used	Versatility	Third Party	Σ
Weight	3	2	2	3	1	3	
Simscape	2	1	3	2	2	2	28
Simulink	3	1	3	3	3	2	35
LabVIEW	3	2	2	2	3	2	32
Modelica	3	3	2	2	2	2	33
Godot	0	3	1	2	3	2	23
Gazebo	2	3	3	3	2	3	**38**

Table A.24: Simulation tool comparison ratings [Lab22]

Rating	Description
0	Requirement not fulfilled
1	Requirement partially fulfilled
2	Requirement fulfilled
3	Requirement exceeded

Table A.25: Simulation tool comparison according to [Kin22] (Table A.24)

Tool	Compatibility	Implementation Effort	Costs	Realtime	Usability	Third Party	Σ
Weight	3	2	2	2	2	1	
LabVIEW	2	2	2	2	2	2	24
Blender	2	2	3	2	3	2	28
Simulink	2	1	1	2	3	3	23
SimScape	2	1	1	2	2	2	20
Gazebo	3	2	3	3	3	1	**32**

A.12.6 Evaluation Results

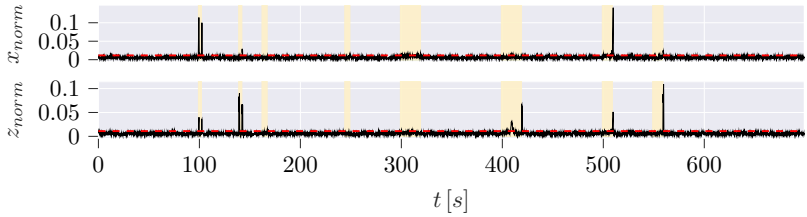


Figure A.54: AE MAE of measurements (window size 10, confusion matrix: Table 6.8)

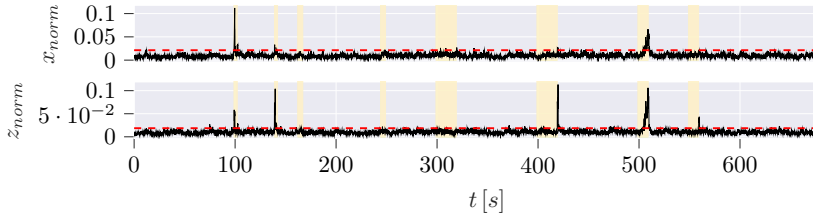


Figure A.55: Hybrid AE 1 MAE of measurements (window size 10, confusion matrix: Table 6.8)

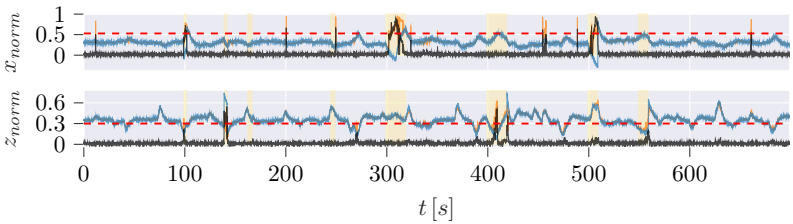


Figure A.56: Absolute error of the UKF predicted by an LSTM with window size 10 using difference of estimation and measurement for CoG position x and z (absolute difference: black, estimated error: orange, ground truth error: blue; confusion matrix Table A.26)

Table A.26: Confusion matrices of hybrid LSTMs for window sizes 20 & 1

Model	**Hybrid LSTM III window size 20**		**Hybrid LSTM window size 1**	
	Predicted Label			
True Label	**False**	**True**	**False**	**True**
False	6213	1	6220	13
True	571	195	440	326

Table A.27: Comparison 99th percentile of x and z positions for learning and hybrid checks

99th Percentile	**LSTM**	**Hybrid LSTM**	**AE**	**Hybrid AE**
x_{norm}	0.403	0.497	0.011	0.069
z_{norm}	0.220	0.305	0.013	0.077

Table A.28: Comparison 99th percentile of x and z positions for dynamic checks

99th Percentile	**UKF**	**EKF**
x	0.044 m	0.036 m
z	0.059 m	0.067 m

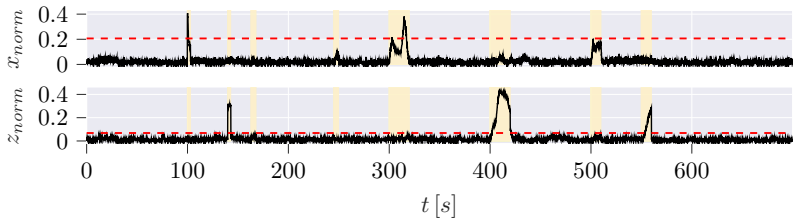


Figure A.57: Prediction of the position difference of in x and z direction based on a feature set of 10 including position measurement x & z, velocity measurements and the full state of the UKF (confusion matrix Table A.26)

A.12.7 Real Data Evaluation Results

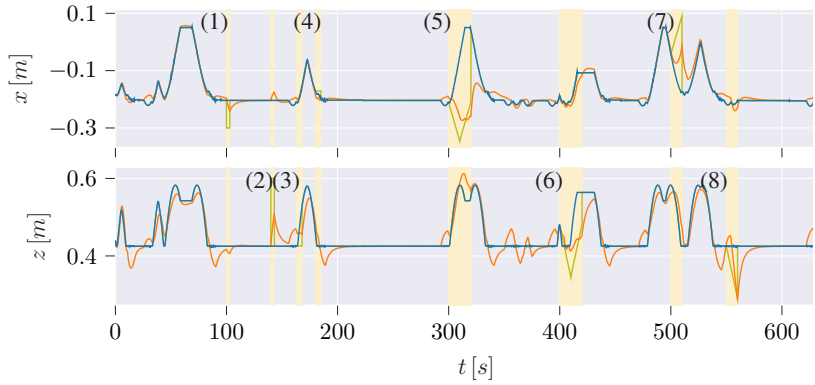


Figure A.58: CoG estimation (*orange*) of the partial body movement UKF compared to measurements using a real system's CAN traces (*green*) and ground truth (*blue*) with anomalies (background marked *yellow*, Table 6.1) using real data

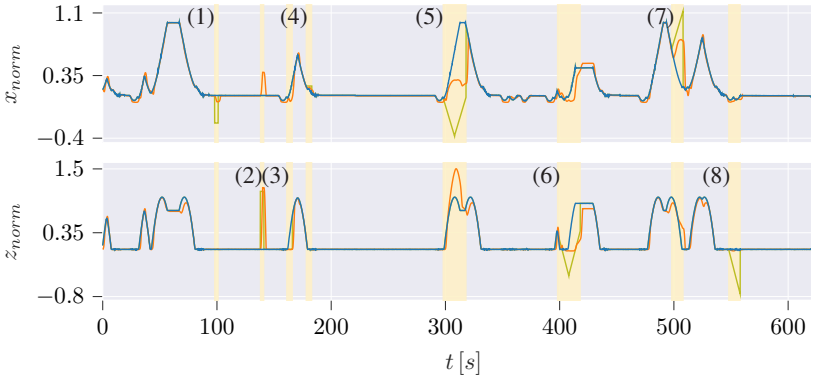


Figure A.59: Normalized prediction of LSTM network (measured: green, estimated: orange, ground truth: blue) with anomalies (background marked *yellow*, Table 6.1) using real data

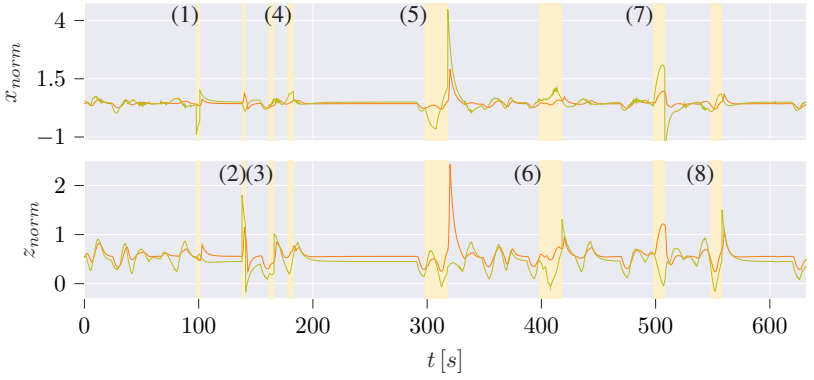


Figure A.60: UKF error ($\hat{\mathbf{x}}_{\mathbf{UKF}} - \mathbf{y}$, Table A.3) predicted by the LSTM (prediction to measurement: orange, prediction to ground truth: blue) with anomalies (background marked *yellow*, Table 6.1) using real data

Own Publications

Journal Articles

[PHS23] Puder, Andreas ; Henle, Jacqueline ; Sax, Eric: Threat Assessment and Risk Analysis (TARA) for Interoperable Medical Devices in the Operating Room Inspired by the Automotive Industry. In: *Healthcare* 11 (2023), Nr. 6. http://dx.doi.org/10.3390/healthcare11060872. – DOI 10.3390/healthcare11060872. – ISSN 2227–9032

[PRGS22] Puder, Andreas ; Rumez, Marcel ; Grimm, Daniel ; Sax, Eric: Generic Patterns for Intrusion Detection Systems in Service-Oriented Automotive and Medical Architectures. In: *Journal of Cybersecurity and Privacy* 2 (2022), Nr. 3, 731–749. http://dx.doi.org/10.3390/jcp2030037. – DOI 10.3390/jcp2030037. – ISSN 2624–800X

[PVR+22] Puder, Andreas ; Vetter, Andreas ; Rumez, Marcel ; Henle, Jacqueline ; Sax, Eric: A Mixed E/E-Architecture for Interconnected Operating Tables Inspired by the Automotive Industry. In: *Journal of Medical Robotics Research* (2022). http://dx.doi.org/10.1142/S2424905X22410082. – DOI 10.1142/S2424905X22410082. – ISSN 2424–905X

[PZSS24] Puder, Andreas ; Zink, Moritz ; Seidel, Luca ; Sax, Eric: Hybrid Anomaly Detection in Time Series by Combining Kalman Filters and Machine Learning Models. In: *Sensors* 24 (2024), Nr. 9. http://dx.doi.org/10.3390/s24092895. – DOI 10.3390/s24092895. – ISSN 1424–8220

Conference Papers

[PSS23] Puder, Andreas ; Schindewolf, Marc ; Sax, Eric: Ontology-Based Service Composition for Interoperable and Modular Medical Devices. In: *2023 IEEE 36th International Symposium on Computer-Based Medical Systems (CBMS)*, 2023, p. 791–797

[PVR+22] Puder, Andreas ; Vetter, Andreas ; Rumez, Marcel ; Henle, Jacqueline ; Sax, Eric: A Mixed E/E-Architecture for Interconnected Operating Tables Inspired by the Automotive Industry. In: *2022 International Symposium on Medical Robotics (ISMR)*, 2022, p. 1–8

[SSG+22] Schindewolf, Marc ; Stoll, Hannes ; Guissouma, Houssem ; Puder, Andreas ; Sax, Eric ; Vetter, Andreas ; Rumez, Marcel ; Henle, Jacqueline: A Comparison of Architecture Paradigms for Dynamic Reconfigurable Automotive Networks. In: *2022 International Conference on Connected Vehicle and Expo (ICCVE)*, IEEE, 3/7/2022 - 3/9/2022. – ISBN 978–1–6654–1687–0, p. 1–7

Patents

[PD22] Puder, Andreas ; Del Alcazar von Buchwald, Rodrigo: *Sicherheitssystem zur Detektion von Fehlern in Medizinischen Tischen (Safety system for detecting errors in medical tables)*. 03.05.2022. – DE102022110888A1/WO2023213868A1

[SPRD22] Schäfer, Achim ; Puder, Andreas ; Rempp, Thibaut ; Del Alcazar von Buchwald, Rodrigo: *Sicherheitssystem zur Detektion einer Kollision eines medizinischen Tischs (Safety system for detecting a collision with a medical table)*. 20.06.2022. – DE102022115287A1/WO2023247447A1

Supervised Student Work

[Gon22] Gondolf, Joscha: *Evaluierung zur Einsetzbarkeit des quelloffenen, serviceorientierten Robotikframeworks ROS2 in der Medizintechnik*. Karlsruhe, Karlsruhe Institute for Technology, Seminar Work, 17.08.2022

[Hal23] Halilovic, Alma: *Physikalisches Simulationsmodell eines modularen Operationstisches im Medizingeräteverbund*. Karlsruhe, Karlsruhe Institute for Technology, Master, 21.08.2023

[Hei24] Heim, Robin: *Lokalisation medizinischer Ausrüstung im Krankenhaus*. Karlsruhe, Karlsruhe Institute for Technology, Master, 02.05.2024

[Kin22] Kinfak Idole, Arthur: *Anomalieerkennung für die vorausschauende Wartung und Sicherheitsüberwachung von Operationstischen*. Rüsselsheim, Hochschule RheinMain, Master, 30.09.2022

[Kli23] Klinkner, Sven: *Architektur zur dynamischen Servicekomposition eines modularen Operationstisches*. Karlsruhe, Karlsruhe Institute for Technology, Bachelor, 26.11.2023

[Koc22] Koch, Laurin J.: *Anomalie-Erkennung mittels maschineller Lernverfahren zur Verbesserung von Safety & Security in Operationstischen*. Karlsruhe, Karlsruhe Institute for Technology, Bachelor, 01.02.2022

[Koc23] Koch, Laurin J.: *Optimierung der Zustandsschätzung in der Robotik: Methodenvergleich und Bewertung*. Karlsruhe, Karlsruhe Institute for Technology, Seminar Work, 01.10.2023

[Lab22] Labidi, Maissa: *Physikalisches Simulationsmodell eines modularen Operationstisches*. Karlsruhe, Karlsruhe Institute for Technology, Bachelor, 23.03.2022

[Mli22] Mlitzko, Sebastian: *Konzept zur Vernetzung von SDC und ROS2 in Medizingeräten*. Karlsruhe, Karlsruhe Institute for Technology, Seminar Work, 03.12.2022

[Pel22] Pellegrini, Sophia: *Digital Twins im Gesundheitswesen*. Karlsruhe, Karlsruhe Institute for Technology, Seminar Work, 01.08.2022

[Pel23] Pellegrini, Sophia: *Anomalie Erkennung im Operationssaal mittels Digitalem Zwilling eines Patienten*. Karlsruhe, Karlsruhe Institute for Technology, Bachelor, 07.09.2023

[Sch22] Schumm, Martin: *Rekonfigurierbare Medizinger¨ate*. Karlsruhe, Karlsruhe Institute for Technology, Seminar Work, 17.10.2022

[Shk23] Shkurtaj, Ema: *Integration von Digital Twins in Safety und Security Maßnahmen für Operationstische*. Karlsruhe, Karlsruhe Institute for Technology, Bachelor, 12.07.2023

[Stu22] Sturm, Jan N.: *Anomalieerkennung in Medizingeräten*. Karlsruhe, Karlsruhe Institute for Technology, Seminar Work, 02.08.2022

[Was22] Wasser, Philipp: *Einsatz von ROS 2 in Medizingeräten*. Karlsruhe, Karlsruhe Institute for Technology, Seminar Work, 18.08.2022

Bibliography

[AAC+20] Anisetti, Marco ; Ardagna, Claudio ; Cremonini, Marco ; Damiani, Ernesto ; Sessa, Jadran ; Costa, Luciana: *Security Threat Landscape*. 2020. – White Paper Security Threats

[ADSW03] Alberts, Christopher J. ; Dorofee, Audrey J. ; Stevens, James F. ; Woody, Carol: *Introduction to the OCTAVE Approach*. https://resources.sei.cmu.edu/library/Asset-view.cfm?assetid=51546. Version: 2003, Accessed 17.12.2022

[AFB+10] Ackerman, Michael J. ; Filart, Rosemarie ; Burgess, Lawrence P. ; Lee, Insup ; Poropatich, Ronald K.: Developing next-generation telehealth tools and technologies: patients, systems, and data perspectives. In: *Telemedicine journal and e-health : the official journal of the American Telemedicine Association* 16 (2010), Nr. 1, p. 93–95. http://dx.doi.org/10.1089/tmj.2009.0153. – DOI 10.1089/tmj.2009.0153

[AGWL09] Arney, David ; Goldman, Julian M. ; Whitehead, Susan ; Lee, Insup: Synchronizing an X-Ray and Anesthesia Machine Ventilator: A Medical Device Interoperability Case Study. In: *Proceedings of the International Conference on Biomedical Electronics and Devices*, SciTePress - Science and and Technology Publications, 2009. – ISBN 978–989–8111–64–7, p. 52–60

[AHJ+18] Andrade, Ricardo de ; Hodel, Kleber N. ; Justo, Joao F. ; Lagana, Armando M. ; Santos, Max M. ; Gu, Zonghua: Analytical and Experimental Performance Evaluations of CAN-FD Bus. In: *IEEE Access* 6 (2018), p. 21287–21295. http://dx.doi.org/10.1109/ACCESS.2018.2826522. – DOI 10.1109/ACCESS.2018.2826522

[AJMD+19] Al-Jarrah, Omar Y. ; Maple, Carsten ; Dianati, Mehrdad ; Oxtoby, David ; Mouzakitis, Alex: Intrusion Detection Systems for Intra-Vehicle Networks: A Review. In: *IEEE Access* 7 (2019), p. 21266–21289. http://dx.doi.org/10.1109/ACCESS.2019.2894183. – DOI 10.1109/ACCESS.2019.2894183

[AKU+18] Andersen, Björn ; Kasparick, Martin ; Ulrich, Hannes ; Franke, Stefan ; Schlamelcher, Jan ; Rockstroh, Max ; Ingenerf, Josef: Connecting the clinical IT infrastructure to a service-oriented architecture of medical devices. In: *Biomedizinische Technik. Biomedical engineering* 63 (2018), January, Nr. 1, p. 57–68. http://dx.doi.org/10.1515/bmt-2017-0021. – DOI 10.1515/bmt–2017–0021

[Ald22] Alder, Steve: Study Confirms Increase in Mortality Rate and Poorer Patient Outcomes After Cyberattacks. In: *The HIPAA Journal* (09.08.2022). https://www.hipaajournal.com/study-confirms-increase-in-mortality-rate-and-poorer-patient-outcomes-after-cyberattacks/, Accessed 18.01.2024

[Alp20] Alpaydin, Ethem: *Introduction to machine learning*. Fourth edition. Cambridge, Massachusetts : The MIT Press, 2020 (Adaptive computation and machine learning series). – ISBN 978–0262043793

[And13] Andreas Göb: *SOA und Softwarequalität*, Technical University Munich (TUM), Dissertation, 2013

[And20] Anderson, Ross: *Security Engineering: A Guide to Building Dependable Distributed Systems*. Indianapolis : John Wiley and Sons, 2020. – ISBN 9781119642787

[ANS19] ANSI/AAMI ; ANSI/AAMI (Ed.): *ANSI/AAMI 2700-1:2019; Medical Devices and Medical Systems—Essential safety and performance requirements for equipment comprising the patient-centric integrated clinical environment (ICE)—Part 1: General requirements and conceptual model*

[APG18] Arney, David ; Plourde, Jeffrey ; Goldman, Julian M.: OpenICE medical device interoperability platform overview and requirement analysis. In: *Biomedizinische Technik. Biomedical engineering* 63 (2018), Nr. 1, p. 39–47. http://dx.doi.org/10.1515/bmt-2017-0040. – DOI 10.1515/bmt–2017–0040

[APVA23] Antonini, Mattia ; Pincheira, Miguel ; Vecchio, Massimo ; Antonelli, Fabio: An Adaptable and Unsupervised TinyML Anomaly Detection System for Extreme Industrial Environments. In: *Sensors (Basel, Switzerland)* 23 (2023), Nr. 4. http://dx.doi.org/10.3390/s23042344. – DOI 10.3390/s23042344

[ASN22] Ahmed, Mohammed A. ; Sindi, Hatem F. ; Nour, Majid: Cybersecurity in Hospitals: An Evaluation Model. In: *Journal of Cybersecurity and Privacy* 2 (2022), Nr. 4, p. 853–861. http://dx.doi.org/10.3390/jcp2040043. – DOI 10.3390/jcp2040043

[AST13] ASTM: *Medical Devices and Medical Systems: Essential safety requirements for equipment comprising the patient-centric integrated clinical environment (ICE) - Part 1: General requirements and conceptual model*. http://www.astm.org/Standards/F2761.htm. Version: 2013

[ASW+21] Ahmad, Zeeshan ; Shahid Khan, Adnan ; Wai Shiang, Cheah ; Abdullah, Johari ; Ahmad, Farhan: Network intrusion detection system: A systematic study of machine learning and deep learning approaches. In: *Transactions on Emerging Telecommunications Technologies* 32 (2021), Nr. 1. http://dx.doi.org/10.1002/ett.4150. – DOI 10.1002/ett.4150. – ISSN 2161–3915

[ATGSK21] Awasthi, Shashank (Ed.) ; Travieso-González, Carlos M. (Ed.) ; Sanyal, Goutam (Ed.) ; Kumar Singh, Dinesh (Ed.): *Artificial Intelligence for a Sustainable*

Industry 4.0. 1st ed. 2021. Cham : Springer International Publishing and Imprint Springer, 2021 (Springer eBook Collection). http://dx.doi.org/10.1007/978-3-030-77070-9. http://dx.doi.org/10.1007/978-3-030-77070-9. – ISBN 9783030770709

[AUT19] AUTOSAR: *Explanation ofAdaptive Platform Design,*. https://www.autosar.org/fileadmin/user_upload/standards/adaptive/19-11/AUTOSAR_EXP_PlatformDesign.pdf. Version: 2019, Accessed 22.02.2022

[AVSL11] Arney, David ; Venkatasubramanian, Krishna K. ; Sokolsky, Oleg ; Lee, Insup: Biomedical devices and systems security. In: *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference* 2011 (2011), p. 2376–2379. http://dx.doi.org/10.1109/IEMBS.2011.6090663. – DOI 10.1109/IEMBS.2011.6090663

[AWMC17] Azimuddin, Anam F. ; Weitzel, Erik K. ; McMains, Kevin C. ; Chen, Philip G.: An ergonomic assessment of operating table and surgical stool heights for seated otolaryngology procedures. In: *Allergy & rhinology (Providence, R.I.)* 8 (2017), Nr. 3, p. 182–188. http://dx.doi.org/10.2500/ar.2017.8.0215. – DOI 10.2500/ar.2017.8.0215. – ISSN 2152–6567

[AWW18] Adamiec-Wójcik, Iwona ; Wojciech, Stanisław: Application of the finite segment method to stabilisation of the force in a riser connection with a wellhead. In: *Nonlinear Dynamics* 93 (2018), Nr. 4, p. 1853–1874. http://dx.doi.org/10.1007/s11071-018-4294-y. – DOI 10.1007/s11071–018–4294–y. – ISSN 1573–269X

[Bab21] Babel, Wolfgang: *Industrie 4.0, China 2025, IoT*. Wiesbaden : Springer Fachmedien Wiesbaden, 2021. http://dx.doi.org/10.1007/978-3-658-34718-5. http://dx.doi.org/10.1007/978-3-658-34718-5. – ISBN 978–3–658–34717–8

[Bal11] Balzert, Helmut: *Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb*. Heidelberg : Spektrum Akademischer Verlag, 2011. http://dx.doi.org/10.1007/978-3-8274-2246-0. http://dx.doi.org/10.1007/978-3-8274-2246-0. – ISBN 978–3–8274–1706–0

[Bar21] Baraniuk, Chris: Why is there a chip shortage? In: *BBC News* (27.8.2021). https://www.bbc.com/news/business-58230388, Accessed 01.09.2022

[BE17] Berdigh, Asmaa ; El Yassini, Khalid: Connected car overview. In: Hamdan, Hani (Ed.) ; Boubiche, Djallel E. (Ed.) ; Klett, Fanny (Ed.): *Proceedings of the 1st International Conference on Internet of Things and Machine Learning*. New York, NY, USA : ACM, 10172017. – ISBN 9781450352437, p. 1–7

[Bey20] Beyaz, Salih: A brief history of artificial intelligence and robotic surgery in orthopedics & traumatology and future expectations. In: *Joint diseases and*

related surgery 31 (2020), Nr. 3, p. 653–655. http://dx.doi.org/10.5606/ehc.2020.75300. – DOI 10.5606/ehc.2020.75300

[Bij18] Bijan, Elahi: *Safety Risk Management for Medical Devices*. Academic Press, 2018. – ISBN 9780128130988

[BJB+19] Beger, Frank ; Janß, Armin ; Bürger, Sebastian ; Kasparick, Martin ; Clusmann, Hans: PriMed – Interdisziplinäre Arbeitsstationen für den OP 4.0. In: *mt|medizintechnik* 4 (2019), p. 25–28

[BKSU23] Borsdorf, Szilvia ; Klodmann, Julian ; Schwier, Andrea ; Unterhinninghofen, Roland: Technologietransfer und Innovationen in der Medizinrobotik. Version: 2023. http://dx.doi.org/10.1007/978-3-658-37157-9{_}29. In: Pfannstiel, Mario A. (Ed.) ; Dautovic, Alma (Ed.): *Transferinnovationen und Innovationstransfer zwischen Wissenschaft und Wirtschaft: Grundlagen, Erkenntnisse und Praxisbeispiele*. Wiesbaden : Springer Fachmedien Wiesbaden, 2023. – DOI 10.1007/978–3–658–37157–9_29. – ISBN 978–3–658–37157–9, p. 569–595

[Bly19] Blyler, John: *What is middle-out systems engineering?* https://www.designnews.com/electronics-test/what-middle-out-systems-engineering. Version: 2019, Accessed 01.11.2022

[BMR+96] Buschmann, Frank ; Meunier, Regine ; Rohnert, Hans ; Sommerlad, Peter ; Stal, Michael: *Pattern-oriented software architecture: A System of Patterns*. Wiley, 1996. – ISBN 9780470059029

[BP20] Bucaioni, Alessio ; Pelliccione, Patrizio: Technical Architectures for Automotive Systems. In: *2020 IEEE International Conference on Software Architecture (ICSA)*, IEEE, 2020. – ISBN 978–1–7281–4659–1, p. 46–57

[BRKW17] Brunner, Stefan ; Roder, Jurgen ; Kucera, Markus ; Waas, Thomas: Automotive E/E-architecture enhancements by usage of ethernet TSN. In: *2017 13th Workshop on Intelligent Solutions in Embedded Systems (WISES)*, IEEE, 2017. – ISBN 978–1–5386–1157–9, p. 9–13

[BRS+19] Berger, Johann ; Rockstroh, Max ; Schreiber, Erik ; Yoshida, Yukishige ; Okamoto, Jun ; Masamune, Ken ; Muragaki, Yoshihiro ; Neumuth, Thomas: GATOR: connecting integrated operating room solutions based on the IEEE 11073 SDC and ORiN standards. In: *International journal of computer assisted radiology and surgery* 14 (2019), Nr. 12, p. 2233–2243. http://dx.doi.org/10.1007/s11548-019-02056-3. – DOI 10.1007/s11548–019–02056–3

[BS13] Bender, Duane ; Sartipi, Kamran: HL7 FHIR: An Agile and RESTful approach to healthcare information exchange. In: *Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems*, IEEE, 2013. – ISBN 978–1–4799–1053–3, p. 326–331

[BSF94] Bengio, Y. ; Simard, P. ; Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. In: *IEEE Transactions on Neural Networks* 5

(1994), Nr. 2, p. 157–166. http://dx.doi.org/10.1109/72.279181. – DOI 10.1109/72.279181

[Bue15] Bueno Ruas de Oliveira, Lucas: *Architectural design of service-oriented robotic systems*, Université de Bretagne Sud, Dissertation, 2015

[BUK+22] Berger, Johann ; Unger, Michael ; Keller, Johannes ; Reich, C. M. ; Neumuth, Thomas ; Melzer, Andreas: Design and validation of a medical robotic device system to control two collaborative robots for ultrasound-guided needle insertions. In: *Frontiers in Robotics and AI* 9 (2022). http://dx.doi.org/10.3389/frobt.2022.875845. – DOI 10.3389/frobt.2022.875845

[Bun17a] Bundesinstitut für Arzneimittel und Medizinprodukte: *Cybersicherheit - Dringende Sicherheitsinformation zu Robot Imager, Gantry Imager von Siemens Healthcare GmbH*. https://www.bfarm.de/SharedDocs/Kundeninfos/DE/18/2017/08284-17_kundeninfo_de.html?nn=597752. Version: 2017, Accessed 22.12.2022

[Bun17b] Bundesinstitut für Arzneimittel und Medizinprodukte: *Cybersicherheit - Dringende Sicherheitsmitteilung für Reinigungs- und Desinfektionsgeräte PG 8527 / 8528 / 8535 / 8536, Miele & Cie. KG*. https://www.bfarm.de/SharedDocs/Kundeninfos/DE/02/2017/03160-17_kundeninfo_de.html?nn=597752. Version: 2017, Accessed 22.12.2022

[Bun18] Bundesamt für Sicherheit in der Informationstechnik: *Redundanz - Modularität - Skalierbarkeit*. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/RZ-Sicherheit/redundanz-modularitaet-skalierbarkeit.html. Version: 2018

[Bun19] Bundesinstitut für Arzneimittel und Medizinprodukte: *Cybersicherheit - Dringende Sicherheitsinformation zu GSS67H von Getinge Sterilization AB*. https://www.bfarm.de/SharedDocs/Kundeninfos/DE/02/2019/13581-19_kundeninfo_de.html?nn=597752. Version: 2019, Accessed 21.12.2022

[Bun21] Bundesgesundheitsministerium: *Krankenhauszukunftsgesetz (KHZG)*. https://www.bundesgesundheitsministerium.de/krankenhauszukunftsgesetz.html. Version: 24.05.2021, Accessed 24.05.2021

[Bun23] Bundesamt für Sicherheit in der Informationstechnik: *Positionspapier Zero Trust 2023*. https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Informationen-und-Empfehlungen/Zero-Trust/zero-trust_node.html. Version: 2023, Accessed 15.08.2023

[BVWB+05] Busch-Vishniac, Ilene J. ; West, James E. ; Barnhill, Colin ; Hunter, Tyrone ; Orellana, Douglas ; Chivukula, Ram: Noise levels in Johns Hopkins Hospital. In: *The Journal of the Acoustical Society of America* 118 (2005), Nr. 6, p. 3629–3645. http://dx.doi.org/10.1121/1.2118327. – DOI 10.1121/1.2118327. – ISSN 0001–4966

[CAL96] CALEM, ROBERT E.: New York's Panix Service Is Crippled by Hacker Attack. In: *New York Times* (14.09.1996). https://archive.nytimes.com/www.nytimes.com/library/cyber/week/0914panix.html, Accessed 25.05.2021

[CAN22] CAN in Automation: *CAN in Automation (CiA): Controller Area Network Extra Long (CAN XL)*. https://www.can-cia.org/can-knowledge/can/can-xl/. Version: 03.03.2022, Accessed 03.03.2022

[Can24] Canonical: *Ubuntu Homepage*. https://ubuntu.com/. Version: May 2024, Accessed 19.05.2024

[CBK09] Chandola, Varun ; Banerjee, Arindam ; Kumar, Vipin: Anomaly detection. In: *ACM Computing Surveys* 41 (2009), Nr. 3, p. 1–58. – ISSN 0360–0300

[CDGA10] Chen, Yinong ; Du, Zhihui ; García-Acosta, Marcos: Robot as a Service in Cloud Computing. In: *2010 Fifth IEEE International Symposium on Service Oriented System Engineering*, IEEE, 062010. – ISBN 978–1–4244–7327–4, p. 151–158

[CDRV09] Christensen, Kaare ; Doblhammer, Gabriele ; Rau, Roland ; Vaupel, James W.: Ageing populations: the challenges ahead. In: *The Lancet* 374 (2009), Nr. 9696, p. 1196–1208. http://dx.doi.org/10.1016/S0140-6736(09)61460-4. – DOI 10.1016/S0140–6736(09)61460–4. – ISSN 01406736

[CEP19] CEP: Buyers' Guide. Version: 2019. http://dx.doi.org/10.1515/9783748602224-011. In: Wilke, Guido (Ed.) ; Ortmeier, Jürgen (Ed.): *Coatings for Plastics*. Vincentz Network, 2019. – DOI 10.1515/9783748602224–011. – ISBN 9783748602224, p. 142–143

[CGMB+24] Comas-González, Zhoe ; Mardini, Johan ; Butt, Shariq A. ; Sanchez-Comas, Andres ; Synnes, Kåre ; Joliet, Aurelian ; Delahoz-Franco, Emiro ; Molina-Estren, Diego ; Piñeres-Espitia, Gabriel ; Naz, Sumera ; Ospino-Balcázar, Daniela: Sensors and Machine Learning Algorithms for Location and POSTURE Activity Recognition in Smart Environments. In: *Automatic Control and Computer Sciences* 58 (2024), February, Nr. 1, 33–42. http://dx.doi.org/10.3103/S0146411624010048, Accessed 08.06.2024. – DOI 10.3103/S0146411624010048. – ISSN 0146–4116, 1558–108X

[CGV+23] Cournapeau, David ; Grisel, Olivier ; Varoquaux, Gaël ; Gramfort, Alexandre ; Mueller, Andreas: *scikit-learn: machine learning in Python*. https://scikit-learn.org/stable/index.html. Version: 08.12.2023, Accessed 08.12.2023

[Cha01] Chambrin, M. C.: Alarms in the intensive care unit: how can the number of false alarms be reduced? In: *Critical care (London, England)* 5 (2001), Nr. 4, p. 184–188. http://dx.doi.org/10.1186/cc1021. – DOI 10.1186/cc1021. – ISSN 1364–8535

[Cli07] Clinical Alarms Task Force: Impact of Clinical Alarms on Patient Safety: A Report From the American College of Clinical Engineering Healthcare Technology Foundation. In: *Journal of Clinical Engineering* 32 (2007), Nr. 1. https://journals.lww.com/jcejournal/Fulltext/2007/01000/Im

pact_of_Clinical_Alarms_on_Patient_Safety__A.24.aspx. – ISSN 0363–8855

[CLX+20] Chen, Tingting ; Liu, Xueping ; Xia, Bizhong ; Wang, Wei ; Lai, Yongzhi: Unsupervised Anomaly Detection of Industrial Robots Using Sliding-Window Convolutional Variational Autoencoder. In: *IEEE Access* 8 (2020), p. 47072–47081. http://dx.doi.org/10.1109/ACCESS.2020.2977892. – DOI 10.1109/ACCESS.2020.2977892

[Con68] Conway, Melvin E.: How Do Committees Invent? (1968), 28–31. http://www.melconway.com/Home/Conways_Law.html, Accessed 31.08.2021

[Cox24] Cox, David: The Global IT Outage Sends Hospitals Reeling. In: *Wired* (2024). https://www.wired.com/story/hospitals-crowdstrike-microsoft-it-outage-meltdown/, Accessed 23.08.2024. – ISSN 1059–1028. – Section: tags

[CSKP16] Corbett, Christopher ; Schoch, Elmar ; Kargl, Frank ; Preussner, Felix: Automotive Ethernet: security opportunity or challenge? In: Meier, Michael (Ed.) ; Reinhardt, Delphine (Ed.) ; Wendzel, Steffen (Ed.): *Sicherheit 2016 - Sicherheit, Schutz und Zuverlässigkeit*. Bonn : Gesellschaft für Informatik e.V, 2016, p. 45–54

[CSL12] Culjat, Martin ; Singh, Rahul ; Lee, Hua: *Medical Devices : Surgical and Image-Guided Technologies : Surgical and Image-Guided Technologies*. Somerset, UNITED STATES : John Wiley & Sons, Incorporated, 2012 http://ebookcentral.proquest.com/lib/karlsruhetech/detail.action?docID=947726. – ISBN 9781118452790

[CTJ21] Chicco, Davide ; Tötsch, Niklas ; Jurman, Giuseppe: The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. In: *BioData mining* 14 (2021), Nr. 1, p. 13. http://dx.doi.org/10.1186/s13040-021-00244-z. – DOI 10.1186/s13040–021–00244–z. – ISSN 1756–0381

[Cva12] Cvach, Maria: Monitor alarm fatigue: an integrative review. In: *Biomedical instrumentation & technology* 46 (2012), Nr. 4, p. 268–277. http://dx.doi.org/10.2345/0899-8205-46.4.268. – DOI 10.2345/0899–8205–46.4.268. – ISSN 0899–8205

[CVK+18] Czaplik, Michael ; Voigt, Verena ; Kenngott, Hannes ; Clusmann, Hans ; Hoffmann, Rüdiger ; Will, Armin: Why OR.NET? Requirements and perspectives from a medical user's, clinical operator's and device manufacturer's points of view. In: *Biomedizinische Technik. Biomedical engineering* 63 (2018), Nr. 1, p. 5–10. http://dx.doi.org/10.1515/bmt-2017-0043. – DOI 10.1515/bmt–2017–0043

[CWA17] Chhetri, Sujit R. ; Wan, Jiang ; Al Faruque, Mohammad A.: Cross-domain security of cyber-physical systems. In: *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*, IEEE, 2017. – ISBN 978–1–5090–1558–0, p. 200–205

[CWH+19] Chng, Chin-Boon ; Wong, Pooi-Mun ; Ho, Nicholas ; Tan, Xiaoyu ; Chui, Chee-Kong: Towards a Cyber-Physical Systems Based Operating Room of the Future. Version: 2019. http://dx.doi.org/10.1007/978-3-030-32695-1{_}6. In: Zhou, Luping (Ed.) ; Sarikaya, Duygu (Ed.) ; Kia, Seyed M. (Ed.) ; Speidel, Stefanie (Ed.) ; Malpani, Anand (Ed.) ; Hashimoto, Daniel (Ed.) ; Habes, Mohamad (Ed.) ; Löfstedt, Tommy (Ed.) ; Ritter, Kerstin (Ed.) ; Wang, Hongzhi (Ed.): *OR 2.0 Context-Aware Operating Theaters and Machine Learning in Clinical Neuroimaging* vol. 11796. Cham : Springer International Publishing, 2019. – DOI 10.1007/978–3–030–32695–1_6. – ISBN 978–3–030–32694–4, p. 47–55

[DCK+06] Deugd, S. de ; Carroll, R. ; Kelly, K. ; Millett, B. ; Ricker, J.: SODA: Service Oriented Device Architecture. In: *IEEE Pervasive Computing* 5 (2006), Nr. 3, p. 94–96, c3. http://dx.doi.org/10.1109/MPRV.2006.59. – DOI 10.1109/MPRV.2006.59. – ISSN 1536–1268

[DDP+15] Dingler, Max ; Dietz, Christian ; Pfeiffer, Jonas ; Lueddemann, Tobias ; Lüth, Tim: A framework for automatic testing of medical device compatibility. In: *2015 13th International Conference on Telecommunications (ConTEL)*, 2015, p. 1–8

[Dec05] Decotignie, J.-D.: Ethernet-Based Real-Time and Industrial Communications. In: *Proceedings of the IEEE* 93 (2005), Nr. 6, p. 1102–1117. http://dx.doi.org/10.1109/JPROC.2005.849721. – DOI 10.1109/JPROC.2005.849721. – ISSN 0018–9219

[Del20] Dell: *Dell Precision 7740 Product Specification*. https://i.dell.com/sites/csdocuments/Product_Docs/en/precision-7740-spec-sheet.pdf. Version: 2020, Accessed 06.07.2024

[Del22] Dell EMC: *Medical Device Security: ADDRESSING THE EVOLVING THREAT LANDSCAPE OF MEDICAL DEVICE CYBERATTACKS*. https://i.dell.com/sites/csdocuments/Business_solutions_whitepapers_Documents/en/Dell_EMC_HC_MedDeviceSecurity_WP_final.pdf. Version: 2022

[Den11] Densen, Peter: Challenges and opportunities facing medical education. In: *Transactions of the American Clinical and Climatological Association* 122 (2011), p. 48–58. – ISSN 0065–7778

[Dep21] Department of Defense: *Zero Trust Reference Architecture*. https://dodcio.defense.gov/Portals/0/Documents/Library/(U)ZT_RA_v1.1(U)_Mar21.pdf. Version: 2021

[DG20] Decker, Peter ; Garnatz, Oliver: CAN XL für zukünftige Fahrzeugarchitekturen. In: *Hanser Automotive* (04.05.2020). https://assets.vector.com/cms/content/know-how/_technical-articles/CAN_XL_Introduction_Hanser_automotive_202007_Pressarticle_DE.pdf

[DH12] Dilcher, Bettina ; Hammerschlag, Lutz: *Klinikalltag und Arbeitszufriedenheit*. Wiesbaden : Springer Fachmedien Wiesbaden, 2012. http://dx.doi.org

/10.1007/978-3-8349-7155-5. http://dx.doi.org/10.1007/978-3-8349-7155-5. – ISBN 978–3–8349–3195–5

[DIC04] DICOM Standards Committee: *DICOM Structured Reporting: Part 1. Overview and Characteristics1*. https://pubs.rsna.org/doi/abs/10.1148/rg.243035710?journalCode=radiographics#d6757896e1. Version: 2004

[Dij78] Dijkstra, Edsger W.: The Humble Programmer. Version: 1978. http://dx.doi.org/10.1007/978-1-4612-6315-9{_}2. In: Gries, David (Ed.): *Programming Methodology: A Collection of Articles by Members of IFIP WG2.3*. New York, NY : Springer New York, 1978. – DOI 10.1007/978–1–4612–6315–9_2. – ISBN 978–1–4612–6315–9, p. 9–22

[DLR21] DLR: *DLR - Institute of Robotics and Mechatronics - MiroSurge*. https://www.dlr.de/rm/en/desktopdefault.aspx/tabid-11674/#gallery/28728. Version: 28.11.2021, Accessed 28.11.2021

[DN19] Duwe, Reiner ; Niewolny, David: *Medizinische Robotik: Auf diese Weise hilft DDS bei der Entwicklung*. https://www.all-electronics.de/elektronik-entwicklung/medizinische-robotik-auf-diese-weise-hilft-dds-bei-der-entwicklung.html. Version: 26. April 2019, Accessed 28.11.2021

[Dow20] Dowalil, Herbert: *Modulare Softwarearchitektur: Nachhaltiger Entwurf durch Microservices, Modulithen und SOA 2.0*. 2., überarbeitete Auflage. München : Hanser, 2020. – ISBN 9783446463776

[Dre17] Dreyfuss, Emily: As Cyberattacks Destabilize the World, the State Department Turns a Blind Eye. In: *WIRED* (21.7.2017). https://www.wired.com/story/state-department-cybersecurity/, Accessed 18.01.2024

[DRF+23] Derrick Tin ; Ryan Hata ; Fredrik Granholm ; Robert G. Ciottone ; Richard Staynings ; Gregory R. Ciottone: Cyberthreats: A primer for healthcare professionals. In: *The American Journal of Emergency Medicine* 68 (2023), 179–185. http://dx.doi.org/10.1016/j.ajem.2023.04.001. – DOI 10.1016/j.ajem.2023.04.001. – ISSN 0735–6757

[DSG+22] Del Alcazar von Buchwald, Rodrigo ; Schäfer, Achim ; Golde, Tim ; Gaiser, Immanuel ; Olszewski, Jan D. ; Obert, Mike: *Operationstisch mit Lastsensoranordnung*. 2022. – DE 10 2021 107 833 A1

[DSL+21] Das, Subrat ; Siroky, Gregory P. ; Lee, Shawn ; Mehta, Davendra ; Suri, Ranjit: Cybersecurity: The need for data and patient safety with cardiac implantable electronic devices. In: *Heart rhythm* 18 (2021), Nr. 3, p. 473–481. http://dx.doi.org/10.1016/j.hrthm.2020.10.009. – DOI 10.1016/j.hrthm.2020.10.009

[DW15] Dröschel, Wolfgang ; Wiemers, Manuela: *Das V-Modell 97: Der Standard fuer die Entwicklung von IT-Systemen mit Anleitung fuer den Praxiseinsatz*. Berlin, Boston : Oldenbourg Wissenschaftsverlag, 2015. http://dx.doi.org/10.1515/9783486800265. http://dx.doi.org/10.1515/9783486800265. – ISBN 978–3–486–80026–5

[EAS23] EAST-ADL Association: *EAST-ADL*. https://east-adl.info/Specification.html. Version: 2023, Accessed 28.01.2024

[Ebe23] Eberhard, Peter: *Flexible Multibody Systems*. https://www.itm.uni-stuttgart.de/en/research/flexible_multibody_systems/. Version: 2023, Accessed 11.04.2023

[EG18] Estdale, John ; Georgiadou, Elli: Applying the ISO/IEC 25010 Quality Models to Software Product. Version: 2018. http://dx.doi.org/10.1007/978-3-319-97925-0{_}42. In: Larrucea, Xabier (Ed.) ; Santamaria, Izaskun (Ed.) ; O'Connor, Rory V. (Ed.) ; Messnarz, Richard (Ed.): *Systems, Software and Services Process Improvement* vol. 896. Cham : Springer International Publishing, 2018. – DOI 10.1007/978–3–319–97925–0_42. – ISBN 978–3–319–97924–3, p. 492–503

[EM16] Ebert, Christof ; Metzker, Eduard: Risiko-orientierte Methodik: Praxis-Erfahrungen zur Anwendung von Cyber Security. In: *Elektronik automotive* Soderausgabe Software (2016). https://www.elektroniknet.de/automotive/software-tools/praxis-erfahrungen-zur-anwendung-von-cyber-security.133930/seite-2.html, Accessed 31.10.2021

[Eng19] Engelke, Julia: Tuttlingen im TV: SWR berichtet über Probleme mit der MDR. In: *devicemed.de* (23.7.2019). https://www.devicemed.de/tuttlingen-im-tv-swr-berichtet-ueber-probleme-mit-der-mdr-a-849083/, Accessed 24.05.2021

[ENI22] ENISA: *ENISA Cybersecurity Threat Landscape Methodology*. 07.2022

[ENI23] ENISA: *ENISA Threat Landscape: Health Sector*. 07.2023

[Eur17] European Parliament and the Council: *Medical Device Regulation: MDR*. https://www.medical-device-regulation.eu/download-mdr/. Version: 05.04.2017

[FDA22] FDA: *[DRAFT] Cybersecurity in Medical Devices: Quality System Considerations and Content of Premarket Submissions Draft Guidance for Industry and Food and Drug Administration Staff*. 2022

[FH19] Fernando Puente León ; Holger Jäkel: *Signale und Systeme*. Berlin, Boston : De Gruyter Oldenbourg, 2019. http://dx.doi.org/10.1515/9783110626322. http://dx.doi.org/10.1515/9783110626322. – ISBN 9783110626322

[Fli15] Fließbach, Torsten: *Mechanik: Lehrbuch zur Theoretischen Physik I*. 7. Aufl. 2015. Berlin, Heidelberg : Springer Berlin Heidelberg, 2015 http://nbn-resolving.org/urn:nbn:de:bsz:31-epflicht-1560155. – ISBN 9783642554322

[Fou24] Foundation, Blender: *Blender Website*. https://www.blender.org/. Version: 2024, Accessed 18.05.2024

[Fra18] Frauenhofer SIT: *EVITA: E-safety vehicle intrusion protected applications*. https://www.evita-project.org/. Version: 13.07.2018, Accessed 28.06.2022

[FSK10] Ferguson, Niels ; Schneier, Bruce ; Kohno, Tadayoshi: *Cryptography engineering: Design principles and practical applications / Niels Ferguson, Bruce Schneier, Tadayoshi Kohno*. Indianapolis, Ind. : Wiley, 2010. – ISBN 9780470474242

[GA08] Grewal, Mohinder S. ; Andrews, Angus P.: *Kalman filtering : theory and practice using MATLAB*. 3rd ed. Hoboken, N.J, 2008 (IEEE Xplore Digital Library)

[GBC16] Goodfellow, Ian ; Bengio, Yoshua ; Courville, Aaron: *Deep Learning*. MIT Press, 2016 https://www.deeplearningbook.org/

[GBK+19] Glock, Thomas ; Betancourt, Victor P. ; Kern, Matthias ; Liu, Bo ; Reiß, Thomas ; Sax, Eric ; Becker, Jürgen: *Proceedings of 2019 8th International Conference on Industrial Technology and Management: ICITM 2019 : March 2-4, 2019, Cambridge, UK*. Piscataway, NJ : IEEE, 2019 https://ieeexplore.ieee.org/servlet/opac?punumber=8700166. – ISBN 9781728132686

[GBM12] Gregorczyk, David ; Bußhaus, Timm ; Mildner, Raimund: *SOA zur Vernetzung medizinischer Geräte - Ein norddeutscher Ansatz*. http:www.doop-projekt.de/soa-vernetzungskonzept.html. Version: 2012

[GBMS18] Gatouillat, Arthur ; Badr, Youakim ; Massot, Bertrand ; Sejdic, Ervin: Internet of Medical Things: A Review of Recent Contributions Dealing With Cyber-Physical Systems in Medicine. In: *IEEE Internet of Things Journal* 5 (2018), Nr. 5, p. 3810–3822. http://dx.doi.org/10.1109/JIOT.2018.2849014. – DOI 10.1109/JIOT.2018.2849014

[GC21] Garbis, Jason ; Chapman, Jerry W.: *Zero Trust Security*. Berkeley, CA : Apress, 2021. http://dx.doi.org/10.1007/978-1-4842-6702-8. http://dx.doi.org/10.1007/978-1-4842-6702-8. – ISBN 978–1–4842–6701–1

[GD20] Gehrmann, Tobias ; Duplys, Paul: Intrusion Detection for SOME/IP: Challenges and Opportunities. In: *2020 23rd Euromicro Conference on Digital System Design (DSD)*, IEEE, 2020. – ISBN 978–1–7281–9535–3, p. 583–587

[GDS18] Guissouma, Houssem ; Diewald, Axel ; Sax, Eric: A Generic System for Automotive Software Over the Air (SOTA) Updates Allowing Efficient Variant and Release Management. Version: 2018. http://dx.doi.org/10.1007/978-3-319-99981-4{_}8. In: Borzemski, Leszek (Ed.) ; Świątek, Jerzy (Ed.) ; Wilimowska, Zofia (Ed.): *Information Systems Architecture and Technology: Proceedings of 39th International Conference on Information Systems Architecture and Technology – ISAT 2018* vol. 852. Cham : Springer International Publishing, 2018. – DOI 10.1007/978–3–319–99981–4_8. – ISBN 978–3–319–99980–7, p. 78–89

[GEKW22] Golde, Tim ; Ertel, Philipp ; Kieffer, Cédric ; Welsch, Michael: *A medical system with an operating table and a remote control, an operating table and a remote control, and a method of operating a medical system with an operating table*. 2022. – EP4123663A1 / WO2023001976A1

[gem23] gematik GmbH: *Feinkonzept Zero Trust Architektur für die Telematikinfrastruktur*. https://fachportal.gematik.de/fileadmin/Fachportal

/Downloadcenter/gemKPT_Zero_Trust_V1.0.0.pdf. Version: 2023, Accessed 15.08.2023

[Gér19] Géron, Aurélien: *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. Second edition. Beijing and Boston and Farnham and Sebastopol and Tokyo : O'Reilly, September 2019. – ISBN 9781492032618

[Ger22] Gerkey, Brian: *Why ROS 2?* https://design.ros2.org/articles/why_ros2.html. Version: 11.02.2022, Accessed 22.02.2022

[Get21] Getinge: *Tegris - Verbesserte OP-Workflows auf Knopfdruck.* https://www.getinge.com/siteassets/products-a-z/tegris/low.res.sal13019_01_de---tegris-brochure.pdf. Version: 2021, Accessed 11.03.2021

[Get22a] Getinge: *Maquet Otesus Operating - A reliable system that improves workflows and patient safety Table System.* https://www.getinge.com/int/product-catalog/maquet-otesus-operating-table-system/. Version: 06.02.2022, Accessed 06.02.2022

[Get22b] Getinge: *Maquet Yuno II operating table.* https://www.getinge.com/int/product-catalog/maquet-yuno-ii-mobile-operating-table/. Version: 2022, Accessed 28.03.2022

[Get23a] Getinge: *Maquet Meera.* https://www.getinge.com/de/produkte/maquet-meera/?tab=1. Version: 01.10.2023, Accessed 01.10.2023

[Get23b] Getinge: *Patient Positioning in the Operating Room.* https://www.getinge.com/int/insights/articles/operating-room/patient-positioning-in-the-or/. Version: 2023, Accessed 01.04.2023

[Get23c] Getinge: *Maquet Corin.* https://www.getinge.com/int/products/maquet-corin/. Version: 30.11.2023, Accessed 30.11.2023

[Get24] Getinge: *Instructions for Use - 7700.01XX Mobile OR Table CORIN.* June 2024

[Gha20] Gharbi, Mahbouba: *Basiswissen für Softwarearchitekten, 4th Edition.* [S.l.] : dpunkt, 2020. – ISBN 3864907810

[GHK+21] Geissler, Frederik ; Heiß, Rafael ; Kopp, Markus ; Wiesmüller, Marco ; Saake, Marc ; Wuest, Wolfgang ; Wimmer, Andreas ; Prell, Veronika ; Uder, Michael ; May, Matthias S.: Personalized computed tomography - Automated estimation of height and weight of a simulated digital twin using a 3D camera and artificial intelligence. In: *RoFo : Fortschritte auf dem Gebiete der Rontgenstrahlen und der Nuklearmedizin* 193 (2021), Nr. 4, p. 437–445. http://dx.doi.org/10.1055/a-1253-8558. – DOI 10.1055/a–1253–8558. – ISSN 1438–9010

[Git22] GitHub: *linux-can: Linux-CAN / SocketCAN user space applications*. https://github.com/linux-can/. Version: 27.09.2022, Accessed 27.09.2022

[GKMS21] Guissouma, Houssem ; Kroger, Janis ; Maelen, Sebastian V. ; Sax, Eric: Extension of Contracts for Variability Modeling and Incremental Update Checks of

Cyber Physical Systems. In: *2021 IEEE International Symposium on Systems Engineering (ISSE)*, IEEE, 2021. – ISBN 978–1–6654–3168–2, p. 1–8

[GPS20] Grimm, Daniel ; Pistorius, Felix ; Sax, Eric: Network Security Monitoring in Automotive Domain. Version: 2020. http://dx.doi.org/10.1007/978-3-030-39445-5{_}57. In: Arai, Kohei (Ed.) ; Kapoor, Supriya (Ed.) ; Bhatia, Rahul (Ed.): *Advances in Information and Communication* vol. 1129. Cham : Springer International Publishing, 2020. – DOI 10.1007/978–3–030–39445–5_57. – ISBN 978–3–030–39444–8, p. 782–799

[Gra22] Grand View Research: *Medical Robotic Systems Market Size, Share & Trends Analysis Report By Type (Surgical Robots, Exo-robots, Pharma Robots, Cleanroom Robots, Robotic Prosthetics, Medical Service Robots), By Region, And Segment Forecasts, 2023 - 2030*. https://www.grandviewresearch.com/industry-analysis/medical-robotic-systems-market#. Version: 2022, Accessed 22.01.2024

[Gre14] Gregorczyk, David: *Technologien für eine interoperable und automatisierte Vernetzung medizinischer IT-Systeme*. Lübeck, University of Lübeck, Dissertation, 07/14/2014

[Gri23] Grimm, Rainer: *Patterns in der Softwarearchitektur: Das Broker-Muster*. https://www.heise.de/blog/Patterns-in-der-Softwarearchitektur-Das-Broker-Muster-8717881.html. Version: 2023, Accessed 21.01.2024

[GS22] Grimm, Daniel ; Sax, Eric: Context-aware vehicle and fleet security combining a Knowledge Graph and an object-oriented model. In: *2022 International Conference on Connected Vehicle and Expo (ICCVE)*, IEEE, 2022. – ISBN 978–1–6654–1687–0, p. 1–8

[GSJW05] Goldman, Julian M. ; Schrenker, Richard A. ; Jackson, Jennifer L. ; Whitehead, Susan F.: Plug-and-Play in the OperatingRoom of the Future. 39 (2005), p. 194–199

[GSS21a] Grimm, Daniel ; Stang, Marco ; Sax, Eric: Context-Aware Security for Vehicles and Fleets: A Survey. In: *IEEE Access* 9 (2021), p. 101809–101846. http://dx.doi.org/10.1109/ACCESS.2021.3097146. – DOI 10.1109/ACCESS.2021.3097146

[GSS21b] Guissouma, Houssem ; Schindewolf, Marc ; Sax, Eric: ICARUS - Incremental Design and Verification of Software Updates in Safety-Critical Product Lines. In: *2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, IEEE, 2021. – ISBN 978–1–6654–2705–0, p. 371–378

[Gui24] Guissouma, Houssem: Positionspapier: Chancen durch Standardisierung und Open-Source-Software im Mobilitätssektor. (2024)

[GVHJ15] Gamma, Erich ; Vlissides, John ; Helm, Richard ; Johnson, Ralph: *Design patterns: Entwurfsmuster als Elemente wiederverwendbarer objektorientierter Software*. Frechen : MITP, 2015 (mitp Professional). – ISBN 3826699033

[GWS18] Grimm, Daniel ; Weber, Marc ; Sax, Eric: An Extended Hybrid Anomaly Detection System for Automotive Electronic Control Units Communicating via Ethernet - Efficient and Effective Analysis using a Specification- and Machine Learning-based Approach. In: *Proceedings of the 4th International Conference on Vehicle Technology and Intelligent Transport Systems*, SCITEPRESS - Science and Technology Publications, 2018. – ISBN 978–989–758–293–6, p. 462–473

[HAMPGCRM23] Hernández-Aceituno, Javier ; Méndez-Pérez, Juan A. ; González-Cava, José M. ; Reboso-Morales, José A.: Towards intelligent supervision of operating rooms using stencil-based character recognition. In: *Computers in biology and medicine* 162 (2023), p. 107071. http://dx.doi.org/10.1016/j.compbiomed.2023.107071. – DOI 10.1016/j.compbiomed.2023.107071

[Har22] Harloff, Thomas: Erster weiblicher Crashtest-Dummy. (16.11.2022). https://www.auto-motor-und-sport.de/verkehr/erster-weiblicher-crashtest-dummy-eva-astrid-linder-vti-schweden/#:~:text=Die%20Standard%2DDummies%20sind%20m%C3%A4nnlich&text=Mit%20der%20Folge%2C%20dass%20Sicherheitssysteme,gro%C3%9F%20und%20wiegt%2078%20Kilogramm., Accessed 10.04.2023

[Haw80] Hawkins, D. M.: *Identification of Outliers*. Dordrecht : Springer Netherlands, 1980. – ISBN 978–94–015–3996–8

[HAZ00] Haider, S. ; Abbas, A. ; Zaidi, A. K.: A multi-technique approach for user identification through keystroke dynamics. In: *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics. 'cybernetics evolving to systems, humans, organizations, and their complex interactions' (cat. no.0* vol. 2, 2000, p. 1336–1341 vol.2

[Hea16] Healthcare IT News: *Hackers hit two California hospitals with ransomware*. https://www.healthcareitnews.com/news/hackers-hit-two-california-hospitals-ransomware. Version: 2016, Accessed 21.12.2022

[Hea22] Health Information Sharing and Analysis Center | H-ISAC: *Home - Health Information Sharing and Analysis Center | H-ISAC*. https://h-isac.org/. Version: 2022, Accessed 02.04.2022

[Hen12] Henniger, Olaf: *E-safety vehicle intrusion protected applications: D0 - Project Summary*. April 2012

[HH20] Hao, Jingjing ; Han, Guangsheng: On the Modeling of Automotive Security: A Survey of Methods and Perspectives. In: *Future Internet* 12 (2020), Nr. 11, 198. http://dx.doi.org/10.3390/fi12110198. – DOI 10.3390/fi12110198

[HHC+20] He, Shilin ; He, Pinjia ; Chen, Zhuangbin ; Yang, Tianyi ; Su, Yuxin ; Lyu, Michael R.: *A Survey on Automated Log Analysis for Reliability Engineering*. http://arxiv.org/pdf/2009.07237v2. Version: 15.09.2020

[HHF+08] Halperin Daniel ; Heydt-Benjamin Thomas S. ; Fu Kevin ; Kohno Tadayoshi ; Maisel William H.: Security and Privacy for Implantable Medical Devices.

In: *IEEE Pervasive Computing* 7 (2008), Nr. 1, p. 30–39. http://dx.doi.org/10.1109/MPRV.2008.16. – DOI 10.1109/MPRV.2008.16. – ISSN 1536–1268

[HHWB02] Hawkins, Simon ; He, Hongxing ; Williams, Graham ; Baxter, Rohan: Outlier Detection Using Replicator Neural Networks. Version: 2002. http://dx.doi.org/10.1007/3-540-46145-0{_}17. In: Goos, Gerhard (Ed.) ; Hartmanis, Juris (Ed.) ; van Leeuwen, Jan (Ed.) ; Kambayashi, Yahiko (Ed.) ; Winiwarter, Werner (Ed.) ; Arikawa, Masatoshi (Ed.): *Data Warehousing and Knowledge Discovery* vol. 2454. Berlin, Heidelberg : Springer Berlin Heidelberg, 2002. – DOI 10.1007/3–540–46145–0_17. – ISBN 978–3–540–44123–6, p. 170–180

[Hil22a] Hillrom Holding: *TS7000 Operating Table*. https://www.hillrom.com/en/products/ts7000-or-table/. Version: 04.02.2022, Accessed 06.02.2022

[Hil22b] Hillrom Holding: *Robotic Operating Rooms*. https://www.hillrom.com/en/surgical-strategic-alliances/robotic-operating-rooms/. Version: 28.01.2022, Accessed 06.02.2022

[HIM15] HIMSS Europe: *Auf den Spuren der Zeitdiebe im Krankenhaus: Die wahre Belastung durch Dokumentation an deutschen Akutkrankenhäusern wird unterschätzt*. 2015

[HKP+18] Hanif, Muhammad A. ; Khalid, Faiq ; Putra, Rachmad Vidya W. ; Rehman, Semeen ; Shafique, Muhammad: Robust Machine Learning Systems: Reliability and Security for Deep Neural Networks. In: *2018 IEEE 24th International Symposium on On-Line Testing And Robust System Design (IOLTS)*, IEEE, 2018. – ISBN 978–1–5386–5992–2, p. 257–260

[HKT+10] Hagn, Ulrich ; Konietschke, R. ; Tobergte, A. ; Nickl, M. ; Jörg, S. ; Kübler, B. ; Passig, G. ; Gröger, M. ; Fröhlich, F. ; Seibold, U. ; Le-Tien, L. ; Albu-Schäffer, A. ; Nothhelfer, A. ; Hacker, F. ; Grebenstein, M. ; Hirzinger, G.: DLR MiroSurge: a versatile system for research in endoscopic telesurgery. In: *International journal of computer assisted radiology and surgery* 5 (2010), Nr. 2, p. 183–193. http://dx.doi.org/10.1007/s11548-009-0372-4. – DOI 10.1007/s11548–009–0372–4

[HL724] HL7 International: *HL7 Standards Product Brief - HL7 Version 3 Standard: Virtual Medical Record for Clinical Decision Support (vMR-CDS) XML Specification, Release 1 | HL7 International*. https://www.hl7.org/implement/standards/product_brief.cfm?product_id=342. Version: 17.01.2024, Accessed 17.01.2024

[HMA09] Hill, David J. ; Minsker, Barbara S. ; Amir, Eyal: Real-time Bayesian anomaly detection in streaming environmental data. In: *Water Resources Research* 45 (2009), Nr. 4. http://dx.doi.org/10.1029/2008WR006956. – DOI 10.1029/2008WR006956

[HNNK14] Hashemi Farzaneh, Morteza ; Nair, Suraj ; Nasseri, Mohammad A. ; Knoll, Alois: Reducing communication-related complexity in heterogeneous networked medical systems considering non-functional requirements. In: *16th International*

Conference on Advanced Communication Technology, IEEE, 2014. – ISBN 978–89–968650–3–2, p. 547–552

[Hof19] Hofmockel, Julia: *Anomalieerkennung in Kommunikationsdaten zur Datenselektion im Fahrzeug*. Karlsruhe, Karlsruhe Institute for Technology, Dissertation, 2019. http://dx.doi.org/10.5445/IR/1000095922. – DOI 10.5445/IR/1000095922

[HS97] Hochreiter, S. ; Schmidhuber, J.: Long short-term memory. In: *Neural computation* 9 (1997), Nr. 8, p. 1735–1780. http://dx.doi.org/10.1162/neco.1997.9.8.1735. – DOI 10.1162/neco.1997.9.8.1735. – ISSN 0899–7667

[HS18] Holle, Jan ; Shukla, Siddharth: Gatekeeper for In-vehicle Network Communication. In: *ATZelektronik worldwide* 13 (2018), Nr. 6, p. 40–43. http://dx.doi.org/10.1007/s38314-018-0075-0. – DOI 10.1007/s38314–018–0075–0

[HSW89] Hornik, Kurt ; Stinchcombe, Maxwell ; White, Halbert: Multilayer feedforward networks are universal approximators. In: *Neural Networks* 2 (1989), Nr. 5, 359-366. http://dx.doi.org/https://doi.org/10.1016/0893-6080(89)90020-8. – DOI https://doi.org/10.1016/0893–6080(89)90020–8. – ISSN 0893–6080

[IBM22] IBM Corporation: *Cost of a Data Breach*. https://www.ibm.com/reports/data-breach. Version: 2022, Accessed 18.01.2024

[IBR+20] Iorio, Marco ; Buttiglieri, Alberto ; Reineri, Massimo ; Risso, Fulvio ; Sisto, Riccardo ; Valenza, Fulvio: Protecting In-Vehicle Services: Security-Enabled SOME/IP Middleware. In: *IEEE Vehicular Technology Magazine* 15 (2020), Nr. 3, p. 77–85. http://dx.doi.org/10.1109/MVT.2020.2980444. – DOI 10.1109/MVT.2020.2980444. – ISSN 1556–6072

[IBZ19] Islam, Shama N. ; Baig, Zubair ; Zeadally, Sherali: Physical Layer Security for the Smart Grid: Vulnerabilities, Threats, and Countermeasures. In: *IEEE Transactions on Industrial Informatics* 15 (2019), Nr. 12, p. 6522–6530. http://dx.doi.org/10.1109/TII.2019.2931436. – DOI 10.1109/TII.2019.2931436. – ISSN 1551–3203

[IEC10] IEC: *IEC 61508-1:2010: Functional safety of electrical/electronic/programmable electronic safety-related systems: Part 1: General requirements*. 2. 2010

[IEC16a] IEC: *IEC60601-2-46:2016 Medical electrical equipment – Part 2-46: Particular requirements for the basic safety and essential performance of operating tables*. 2016

[IEC16b] IEC: *IEC62304:2016 Medical device software: Software life cycle processes*. Edition 1.1. 2016

[IEC18] IEC: *IEC62443-4-1:2018 Security for industrial automation and control systems: Part 4-1: Secure product development lifecycle requirements*. 2018

[IEC20] IEC: *IEC60601-1:2020 Medical electrical equipment: Part 1: General requirements for basic safety and essential performance*. 3.2. 2020

[IEC21] IEC: *IEC81001-5-1: Health software and health IT systems safety, effectiveness and security: Part 5-1: Security – Activities in the product life cycle*. 2021

[IEE] IEEE: *IEEE802.1Q-2018 - IEEE Standard for Local and Metropolitan Area Network: Bridges and Bridged Networks*. https://ieeexplore.ieee.org/servlet/opac?punumber=8403925

[IEE16] IEEE: IEEE802.3bp-2016 Standard for Ethernet Amendment 4: Physical Layer Specifications and Management Parameters for 1 Gb/s Operation over a Single Twisted-Pair Copper Cable. In: *IEEE Std 802.3bp-2016 (Amendment to IEEE Std 802.3-2015 as amended by IEEE Std 802.3bw-2015, IEEE Std 802.3by-2016, and IEEE Std 802.3bq-2016)* (2016), p. 1–211. http://dx.doi.org/10.1109/IEEESTD.2016.7564011. – DOI 10.1109/IEEESTD.2016.7564011

[IEE18] IEEE: *11073-10201-2018 - IEEE Standard for Health Informatics - Point-of-care medical device communication - Part 10201: Domain Information Model*. https://ieeexplore.ieee.org/servlet/opac?punumber=8734167. Version: 2018

[IEE19] IEEE: *IEEE Std 802.3cg ™–2019: IEEE Standard for Ethernet Amendment 5: Physical Layers Specifications and Management Parameters for 10 Mb/s Operation and Associated Power Delivery over a Single Balanced Pair of Conductors*

[ILW+23] Illiano, Ludovica ; Liu, Xiaokang ; Wu, Xinglong ; Grassi, Flavia ; Pignari, Sergio A.: Circuit Modeling of Fast Ethernet Signal for EMC and SI Analysis. In: *2023 International Symposium on Electromagnetic Compatibility – EMC Europe*, IEEE, 2023. – ISBN 979–8–3503–2400–6, p. 1–6

[Inf23] Infineon Technologies AG: *XMC4700 - Infineon Technologies*. https://www.infineon.com/cms/en/product/microcontroller/32-bit-industrial-microcontroller-based-on-arm-cortex-m/32-bit-xmc4000-industrial-microcontroller-arm-cortex-m4/xmc4700/. Version: 05.12.2023, Accessed 05.12.2023

[Int08] International Organization for Standardization: *Systems and software engineering: Requirements for designers and developers ofuser documentation,*. https://www.iso.org/obp/ui/#iso:std:43073:en. Version: 2008, Accessed 07.11.2021

[Int15] International Organization for Standardization: *Software and systems engineering: Reference model for product line engineering and management*. 2015

[Int20] International Federation of Robotics: *World Robotics 2020 – Service Robots Report*. Frankfurt am Main, Germany, 2020

[ISB+16] Islam, Mafijul ; Sandberg, Christian ; Bokesand, Andreas ; Olovsson, Tomas ; Kleberger, Pierre ; Lautenbach, Aljoscha ; Söderberg-Söderberg-Rivkin, Andrew ; Kadhirvelan, Sathya P. ; Hansson, Anders ; Broberg, Henrik: *Security models: HEAling Vulnerabilities to ENhance Software Security and Safety*. 2.0. 2016

[ISO11a] ISO: *ISO/IEC 25010:2011 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRhersE): System and software quality models*. 2011

[ISO11b] ISO/IEC/IEEE: *ISO/IEC/IEEE 42010:2011 Systems and software engineering – Architecture description*

[ISO14] ISO: *ISO14708-1:2014 Implants for surgery - Active implantable medical devices: Part 1: General requirements for safety, marking and for information to be provided by the manufacturer*. https://www.iso.org/standard/52804.html. Version: 2, 08.2014, Accessed 19.01.2024

[ISO15] ISO/IEC: *ISO/IEC 27039:2015 Information technology — Security techniques: Selection, deployment and operations of intrusion detection and prevention systems (IDPS)*. 1. 2015

[ISO16] ISO/IEEE: *ISO/IEEE11073-10424 Health informatics - Personal health device communication: Part 10424: Device specialization - Sleep apnoea breathing therapy equipment (SABTE)*. https://www.iso.org/obp/ui#iso:std:iso-ieee:11073:-10424:ed-1:v1:en:term:3.1.17. Version: 2016

[ISO17] ISO/IEC/IEEE: *ISO/IEC/IEEE 24765: Systems and software engineering: Vocabulary*. 2. 2017

[ISO18a] ISO: *ISO26262-9:2018 Road vehicles - Functional safety: Part 9: Automotive safety integrity level (ASIL)-oriented and safety-oriented analyses*. https://www.iso.org/standard/68391.html. Version: 12.2018, Accessed 19.01.2024

[ISO18b] ISO/IEC: *ISO/IEC 27000:2018 Information technology — Security techniques — Information security management systems: Overview and*. 5. 2018

[ISO19a] ISO/IEC: *Guide to the development and inclusion of aspects of safety in International Standards for medical devices*

[ISO19b] ISO/IEEE: *ISO/IEEE11073-10207:2019 Health informatics - Personal health device communication: Part 10207: Domain information and service model for service-oriented point-ofcare medical device communication*. first edition. 2019

[ISO19c] ISO/IEEE: *ISO/IEEE11073-10419 Health informatics — Personal health device communication: Part 10419: Device specialization — Insulin pump*. https://www.iso.org/obp/ui#iso:std:iso-ieee:11073:-10419:ed-2:v1:en:term:3.1.30. Version: 2019

[ISO21a] ISO: *ISO81001-1:2021 Health software and health IT systems safety, effectiveness and security: Part 1: Principles and concepts*. 1. 2021

[ISO21b] ISO/SAE: *ISO/SAE 21434:2021 Road vehicles - Cybersecurity engineering*. 1. 08.2021

[ISO01] ISO/IEC: *ISO/IEC Guide 51:2014 Safety Aspects: Guidelines for their inclusion in standards*. 3. 2014-04-01

[IT94] ITU-T, Hrsg.: *Information technology - Open Systems Interconnection, X.200*. https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=2820&lang=en. Version: 1994, Accessed 29.08.2021

[ITK21] ITK Engineering GmbH: *Medizinrobotik: Der Weg von der Idee bis zum fertigen Produkt*. 2021

[Jam13] James, John T.: A new, evidence-based estimate of patient harms associated with hospital care. In: *Journal of patient safety* 9 (2013), Nr. 3, p. 122–128. http://dx.doi.org/10.1097/PTS.0b013e3182948a69. – DOI 10.1097/PTS.0b013e3182948a69

[JK17] Jones, Richard W. ; Katzis, Konstantinos: Cybersecurity and the Medical Device Product Development Lifecycle. In: *Informatics Empowers Healthcare Transformation* 238 (2017)

[JK18] Jalali, Mohammad S. ; Kaiser, Jessica P.: Cybersecurity in Hospitals: A Systematic, Organizational Perspective. In: *Journal of medical Internet research* 20 (2018), Nr. 5, e10059. http://dx.doi.org/10.2196/10059. – DOI 10.2196/10059

[JKL21] Jiang, Jehn-Ruey ; Kao, Jian-Bin ; Li, Yu-Lin: Semi-Supervised Time Series Anomaly Detection Based on Statistics and Deep Learning. In: *Applied Sciences* 11 (2021), Nr. 15, p. 6698. http://dx.doi.org/10.3390/app11156698. – DOI 10.3390/app11156698

[JM75] Jerome H. Saltzer ; Michael D. Schroeder: The protection of information in computer systems. In: *Proceedings of the IEEE* 63 (1975), p. 1278–1308. – ISSN 0018–9219

[JR94a] J.D. Connelly ; R.L. Huston: The dynamics of flexible multibody systems: A finite segment approach—I. Theoretical aspects. In: *Computers & Structures* 50 (1994), Nr. 2, 255–258. http://dx.doi.org/10.1016/0045-7949(94)90300-X. – DOI 10.1016/0045–7949(94)90300–X. – ISSN 0045–7949

[JR94b] J.D. Connelly ; R.L. Huston: The dynamics of flexible multibody systems: A finite segment approach—II. Example problems. In: *Computers & Structures* 50 (1994), Nr. 2, 259–262. http://dx.doi.org/10.1016/0045-7949(94)90301-8. – DOI 10.1016/0045–7949(94)90301–8. – ISSN 0045–7949

[JRS+21] Jin, Xue-Bo ; Robert Jeremiah, Ruben J. ; Su, Ting-Li ; Bai, Yu-Ting ; Kong, Jian-Lei: The New Trend of State Estimation: From Model-Driven to Hybrid-Driven Methods. In: *Sensors (Basel, Switzerland)* 21 (2021), Nr. 6. http://dx.doi.org/10.3390/s21062085. – DOI 10.3390/s21062085

[JWH17] Jesse, Bernd ; Weber, Marc ; Helmling, Markus: Die Zukunft mit SOA, POSIX, TSN Automotive Ethernet: Trends und Herausforderungen. In: *Automobil Elektronik* (2017), Nr. 11-12 2017. https://cdn.vector.com/cms/content/know-how/_technical-articles/Ethernet_Trends_AutomobilElektronik_201712_PressArticle_DE.pdf

[KAA+21] Kiranyaz, Serkan ; Avci, Onur ; Abdeljaber, Osama ; Ince, Turker ; Gabbouj, Moncef ; Inman, Daniel J.: 1D convolutional neural networks and applications: A survey. In: *Mechanical Systems and Signal Processing* 151 (2021), 107398. http://

dx.doi.org/https://doi.org/10.1016/j.ymssp.2020.107398. – DOI https://doi.org/10.1016/j.ymssp.2020.107398. – ISSN 0888–3270

[Käb13] Käbisch, Sebastian: *Resource Optimization of SOA-Technologies in Embedded Networks*. Passau, University of Passau, Dissertation, 2013

[Käf17] Käfer, Andreas: *Simulation dynamischer Positionsdaten eines Operationstisches mittels Hardware-in-the-Loop*. Karlsruhe, Karlsruhe Institute for Technology, Master, 2017

[KAK+19] Kampmann, Alexandru ; Alrifaee, Bassam ; Kohout, Markus ; Wüstenberg, Andreas ; Woopen, Timo ; Nolte, Marcus ; Eckstein, Lutz ; Kowalewski, Stefan: A dynamic service-oriented software architecture for highly automated vehicles. In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. Auckland, New Zealand : IEEE, 2019, p. 2101–2108

[KAKA06] Krettek, Christian (Ed.) ; Aschemann, Dirk (Ed.) ; Krettek-Aschemann (Ed.): *Positioning techniques in surgical applications: Thorax and heart surgery - vascular surgery - visceral and transplantation surgery - urology - surgery of the spinal cord and extremities - arthroscopy - paediatric surgery - navigation/ISO-C 3D*. Heidelberg : Springer, 2006. – ISBN 3540257160

[Kal60] Kalman, R. E.: A New Approach to Linear Filtering and Prediction Problems. In: *Journal of Basic Engineering* 82 (1960), Nr. 1, p. 35–45. http://dx.doi.org/10.1115/1.3662552. – DOI 10.1115/1.3662552. – ISSN 0021–9223

[Kal22] Kalinowsky, Anna: *So installieren Sie macOS in einer virtuellen Maschine*. https://www.heise.de/tipps-tricks/So-installieren-Sie-macOS-in-einer-virtuellen-Maschine-4102565.html. Version: April 2022, Accessed 19.05.2024

[Kar21] Karl Storz GmbH & Co. KG: *KARL STORZ OR1 FUSION – Der integrierte Operationssaal*. https://www.karlstorz.com/cps/rde/xbcr/karlstorz{_}assets/ASSETS/3312893.pdf. Version: 2021, Accessed 11.03.2021

[Kas20] Kasparick, Martin: *Zuverlässige und herstellerübergreifende Medizingeräteinteroperabilität& Beiträge zur IEEE 11073 SDC-Normenfamilie*

[KH04] Koenig, N. ; Howard, A.: Design and use paradigms for gazebo, an open-source multi-robot simulator. In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, IEEE, 2004. – ISBN 0–7803–8463–6, p. 2149–2154

[KH19] Klonoff, David ; Han, Julia: The First Recall of a Diabetes Device Because of Cybersecurity Risks. In: *Journal of diabetes science and technology* 13 (2019), Nr. 5, p. 817–820. http://dx.doi.org/10.1177/1932296819865655. – DOI 10.1177/1932296819865655

[Kil12] Kilgore, J. L.: Anthropometric variance in humans: Assessing Renaissance concepts in modern applications. In: *Anthropological Notebooks* 18 (2012), p. 13–23

[Kin10] Kindervag, John: *No More Chewy Centers: Introducing The Zero Trust Model Of Information Security*. 2010

[KM09] Killourhy, Kevin S. ; Maxion, Roy A.: Comparing anomaly-detection algorithms for keystroke dynamics. In: *2009 IEEE/IFIP International Conference on Dependable Systems & Networks*, 2009, p. 125–134

[Kou21] Koubaa, Anis: *Robot Operating System (ROS)*. vol. 962. Cham : Springer International Publishing, 2021. http://dx.doi.org/10.1007/978-3-030-75472-3. http://dx.doi.org/10.1007/978-3-030-75472-3. – ISBN 978–3–030–75471–6

[KRA+10] King, Andrew L. ; Roederer, Alex ; Arney, David ; Chen, Sanjian ; Fortino-Mullen, Margaret ; Giannareas, Ana ; Hanson, William ; Kern, Vanessa ; Stevens, Nicholas ; Tannen, Jonathan ; Trevino, Adrian V. ; Park, Soojin ; Sokolsky, Oleg ; Lee, Insup: GSA: A Framework for Rapid Prototyping of Smart Alarm Systems. In: *Proceedings of the 1st ACM International Health Informatics Symposium*. New York, NY, USA : Association for Computing Machinery, 2010 (IHI '10). – ISBN 9781450300308, p. 487–491

[Kre17] Kreissl, Jochen: *Absicherung der SOME/IP Kommunikation bei Adaptive AUTOSAR*, Univsersity of Stuttgart, Master's dissertation, 11/15/2017

[KRS11] Kroschel, Kristian ; Rigoll, Gerhard ; Schuller, Björn: *Statistische Informationstechnik: Signal- und Mustererkennung, Parameter- und Signalschätzung*. 5. Aufl. Heidelberg : Springer, 2011. – ISBN 3642159532

[KRS+16] Kasparick, Martin ; Rockstroh, Max ; Schlichting, Stefan ; Golatowski, Frank ; Timmermann, Dirk: Mechanism for safe remote activation of networked surgical and PoC devices using dynamic assignable controls. In: *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference* 2016 (2016), p. 2390–2394. http://dx.doi.org/10.1109/EMBC.2016.7591211. – DOI 10.1109/EMBC.2016.7591211

[KRS20] Khalid, Faiq ; Rehman, Semeen ; Shafique, Muhammad: Overview of Security for Smart Cyber-Physical Systems. Version: 2020. http://dx.doi.org/10.1007/978-3-030-45541-5{_}2. In: Karimipour, Hadis (Ed.) ; Srikantha, Pirathayini (Ed.) ; Farag, Hany (Ed.) ; Wei-Kocsis, Jin (Ed.): *Security of Cyber-Physical Systems*. Cham : Springer International Publishing, 2020. – DOI 10.1007/978–3–030–45541–5_2. – ISBN 978–3–030–45540–8, p. 5–24

[KS15] Köhler-Schute, Christiana (Ed.): *Industrie 4.0: Ein praxisorientierter Ansatz*. Berlin : KS-Energy-Verl., 2015. – ISBN 9783945622018

[KSA+18] Kasparick, Martin ; Schmitz, Malte ; Andersen, Björn ; Rockstroh, Max ; Franke, Stefan ; Schlichting, Stefan ; Golatowski, Frank ; Timmermann, Dirk: OR.NET: a service-oriented architecture for safe and dynamic medical device interoperability. In: *Biomedical Engineering / Biomedizinische Technik* 63 (2018), Nr. 1, 11–30. http://dx.doi.org/doi:10.1515/bmt-2017-0020. – DOI doi:10.1515/bmt–2017–0020

[KSFWK20] Karimipour, Hadis ; Srikantha, Pirathayini ; Farag, Hany ; Wei-Kocsis, Jin: *Security of Cyber-Physical Systems*. Cham : Springer International Publishing, 2020. http://dx.doi.org/10.1007/978-3-030-45541-5. http://dx.doi.org/10.1007/978-3-030-45541-5. – ISBN 978–3–030–45540–8

[KSKC22] Kao, Ming-Tsung ; Sung, Dian-Ye ; Kao, Shang-Juh ; Chang, Fu-Min: A Novel Two-Stage Deep Learning Structure for Network Flow Anomaly Detection. In: *Electronics* 11 (2022), Nr. 10, p. 1531. http://dx.doi.org/10.3390/electronics11101531. – DOI 10.3390/electronics11101531

[Kuc20] Kucera, Martin: Dräger: Anästhesieplattform mit SDC-Standard auf dem Markt. In: *Georg Thieme Verlag KG* (17.2.2020). https://www.kma-online.de/aktuelles/medizintechnik/detail/anaesthesieplattform-mit-sdc-standard-auf-dem-markt-a-42662, Accessed 19.02.2024

[Kul16] Kulik, Bernhard: Operationstischsysteme. Version: 2016. http://dx.doi.org/10.1007/978-3-662-45538-8{_}49-1. In: Kramme, Rüdiger (Ed.): *Medizintechnik: Verfahren - Systeme - Informationsverarbeitung*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2016. – DOI 10.1007/978–3–662–45538–8_49–1. – ISBN 978–3–662–45538–8, p. 1–12

[Kus13] Kushner, David: The Real Story of Stuxnet. In: *IEEE Spectrum* (26.2.2013). https://spectrum.ieee.org/the-real-story-of-stuxnet, Accessed 09.02.2024

[KWL+12] Koeny, Marcus ; Walter, Marian ; Leonhardt, Steffen ; Benzko, Julia ; Radermacher, Klaus ; Czaplik, Michael ; Rossaint, Rolf: Getting Anesthesia Online: The smartOR Network. In: *International Journal on Advances in Internet Technology* 5 (2012)

[Kö13] Köllner, Christian: *Transformation von Multiphysics-Modellen in einen FPGA-Entwurf für den echtzeitfähigen HiL-Test eingebetteter Systeme*, KIT Scientific Publishing, PhD Thesis, 2013. http://dx.doi.org/10.5445/KSP/1000036853. – DOI 10.5445/KSP/1000036853. – ISBN: 978-3-7315-0120-6

[Lab20] Labbe, Roger: Kalman and Bayesian Filters in Python. (2020). https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python, Accessed 27.01.2024

[Lab22] Labbe, Roger: *FilterPy*. https://github.com/rlabbe/filterpy. Version: 2022, Accessed 27.01.2024

[LAN16] LAN/MAN Committee of the IEEE Computer Society: *IEEE 802.3 Standard for Ethernet: Amendment 1: Physical layer specifications and management parameters for 100 Mb/s operation over a single balanced twisted pair cable (100BASE-T1)*. New York : IEEE, 2016 https://ieeexplore.ieee.org/servlet/opac?punumber=7433916. – ISBN 9781504401371

[Lan21] Langmann, Reinhard: *Vernetzte Systeme für die Automatisierung 4.0: Bussysteme – Industrial Ethernet – Mobile Kommunikation – Cyber-Physical Systems*. Munich : Hanser, 2021. – ISBN 978–3–446–46939–6

[LAO21] Lautenbach, Aljoscha ; Almgren, Magnus ; Olovsson, Tomas: Proposing HEAVENS 2.0 – an automotive risk assessment model. In: Brücher, Björn (Ed.) ; Krauß, Christoph (Ed.) ; Fritz, Mario (Ed.) ; Hof, Hans-Joachim (Ed.) ; Wasenmüller, Oliver (Ed.): *Computer Science in Cars Symposium*. New York, NY, USA : ACM, 2021. – ISBN 9781450391399, p. 1–12

[Law13] Lawrenz, Wolfhard: *CAN System Engineering*. London : Springer London, 2013. http://dx.doi.org/10.1007/978-1-4471-5613-0. http://dx.doi.org/10.1007/978-1-4471-5613-0. – ISBN 978–1–4471–5612–3

[Lee06] Lee, Edward A.: Cyber-Physical Systems - Are Computing Foundations Adequate? Position Paper for NSF Workshop On Cyber-Physical Systems: Research Motivation, Techniques and Roadmap, Austin. (2006)

[Len20] Lenhard, Thomas H.: *Datensicherheit*. Wiesbaden : Springer Fachmedien Wiesbaden, 2020. http://dx.doi.org/10.1007/978-3-658-29866-1. http://dx.doi.org/10.1007/978-3-658-29866-1. – ISBN 978–3–658–29865–4

[LLP22] Liu, Wing K. ; Li, Shaofan ; Park, Harold S.: Eighty Years of the Finite Element Method: Birth, Evolution, and Future. In: *Archives of Computational Methods in Engineering* 29 (2022), Nr. 6, p. 4431–4453. http://dx.doi.org/10.1007/s11831-022-09740-9. – DOI 10.1007/s11831–022–09740–9. – ISSN 1134–3060

[LNJ08] Larson Ulf E. ; Nilsson Dennis K. ; Jonsson Erland: An approach to specification-based attack detection for in-vehicle networks. In: *2008 IEEE Intelligent Vehicles Symposium*, 2008, p. 220–225

[LSC+12] Lee, Insup ; Sokolsky, O. ; Chen, Sanjian ; Hatcliff, J. ; Jee, Eunkyoung ; Kim, BaekGyu ; King, A. ; Mullen-Fortino, Margaret ; Park, Soojin ; Roederer, A. ; Venkatasubramanian, K. K.: Challenges and Research Directions in Medical Cyber–Physical Systems. In: *Proceedings of the IEEE* 100 (2012), Nr. 1, p. 75–90. http://dx.doi.org/10.1109/JPROC.2011.2165270. – DOI 10.1109/JPROC.2011.2165270. – ISSN 0018–9219

[LTZ08] Liu, Fei T. ; Ting, Kai M. ; Zhou, Zhi-Hua: Isolation Forest. In: *2008 Eighth IEEE International Conference on Data Mining*, IEEE, 2008. – ISBN 978–0–7695–3502–9, p. 413–422

[LWX20] Liu, Jingxian ; Wang, Zulin ; Xu, Mai: DeepMTT: A deep learning maneuvering target-tracking algorithm based on bidirectional LSTM network. In: *Information Fusion* 53 (2020), 289–304. http://dx.doi.org/10.1016/j.inffus.2019.06.012. – DOI 10.1016/j.inffus.2019.06.012. – ISSN 1566–2535

[Mad20] Madinejad, Mo: *Medical Device Cybersecurity in the Age of IoMT*. https://www.medtechintelligence.com/column/medical-device-

cybersecurity-in-the-age-of-iomt/. Version: 2020, Accessed 24.11.2022

[Mag17] Maggi, Federico: *A Vulnerability in Modern Automotive Standards and How We Exploited It*. https://documents.trendmicro.com/assets/A-Vulnerability-in-Modern-Automotive-Standards-and-How-We-Exploited-It.pdf. Version: 2017 (Technical Brief)

[MALA+22] Mattila, Joel ; Ala-Laurinaho, Riku ; Autiosalo, Juuso ; Salminen, Pauli ; Tammi, Kari: Using Digital Twin Documents to Control a Smart Factory: Simulation Approach with ROS, Gazebo, and Twinbase. In: *Machines* 10 (2022), Nr. 4. http://dx.doi.org/10.3390/machines10040225. – DOI 10.3390/machines10040225. – ISSN 2075–1702

[MAPR19] Martin Eigner ; Alexander Detzner ; Philipp Heiner Schmidt ; Rajeeth Tharma: Definition des Digital Twin im Produktlebenszyklus. In: *Zeitschrift für wirtschaftlichen Fabrikbetrieb* 114 (2019), Nr. 6. http://dx.doi.org/10.3139/104.112107. – DOI 10.3139/104.112107

[Mat24] MathWorks: *MATLAB*. https://de.mathworks.com/products/matlab.html. Version: 2024, Accessed 19.05.2024

[May22] Mayer, Boris: Security bei der US-Regierung: VPN, SMS-Codes und Passwörter sind out, Zero Trust ist in - Golem.de. In: *Golem.de* (02.08.2022). https://www.golem.de/news/security-bei-der-us-regierung-vpn-sms-codes-und-passwoerter-sind-out-zero-trust-ist-in-2202-162871.html, Accessed 09.02.2022

[MBB18] Maul, Mario ; Becker, Gerhard ; Bernhard, Ulrich: Serviceorientierte EE-Zonenarchitektur Schlüsselelement für neue Marktsegmente. In: *ATZelektronik* 13 (2018), Nr. 1, p. 36–41. http://dx.doi.org/10.1007/s35658-017-0105-3. – DOI 10.1007/s35658–017–0105–3. – ISSN 1862–1791

[MBF+19] Martin Kasparick ; Bjorn Butzin ; Frank Golatowski ; Jonas Pabst ; Hans-Joachim Cappius ; Peter Westerhoff ; Bjorn Andersen ; Dirk Timmermann: *The 2019 International Conference on Smart Applications, Communications and Networking (SmartNets 2019): December 17-19, 2019, Sharm El Sheikh, Egypt*. Piscataway, NJ : IEEE, 2019. http://dx.doi.org/10.1109/SmartNets48225.2019. http://dx.doi.org/10.1109/SmartNets48225.2019. – ISBN 9781728142753

[MBH+21] Münster, Marco ; Brost, Mascha ; Hahn, Robert ; Siefkes, Tjark ; Kopp, Gerhard ; Schmid, Stephan ; Knutzen, Tim: U-Shift Vehicle Concept: Modular on the Road. In: Bargende, Michael (Ed.) ; Reuss, Hans-Christian (Ed.) ; Wagner, Andreas (Ed.): *21. Internationales Stuttgarter Symposium*. Wiesbaden : Springer Fachmedien Wiesbaden, 2021. – ISBN 978–3–658–33466–6, p. 333–346

[MBT12] Moubarak, Paul ; Ben-Tzvi, Pinhas: Modular and reconfigurable mobile robotics. In: *Robotics and Autonomous Systems* 60 (2012), Nr. 12, p. 1648–1663. http://dx.doi.org/10.1016/j.robot.2012.09.002. – DOI 10.1016/j.robot.2012.09.002. – ISSN 09218890

[Med17] MedPAC: *Report to the Congress: Medicare and the Health Care Delivery System*. http://www.medpac.gov/docs/default-source/reports/jun17_reporttocongress_sec.pdf?sfvrsn=0. Version: 2017, Accessed 31.08.2021

[MGF10] Muter, Michael ; Groll, Andre ; Freiling, Felix C.: A structured approach to anomaly detection for in-vehicle networks. In: *2010 Sixth International Conference on Information Assurance and Security*, IEEE, 2010. – ISBN 978–1–4244–7407–3, p. 92–98

[MHVS+17] Maier-Hein, Lena ; Vedula, Swaroop S. ; Speidel, Stefanie ; Navab, Nassir ; Kikinis, Ron ; Park, Adrian ; Eisenmann, Matthias ; Feussner, Hubertus ; Forestier, Germain ; Giannarou, Stamatia ; Hashizume, Makoto ; Katic, Darko ; Kenngott, Hannes ; Kranzfelder, Michael ; Malpani, Anand ; März, Keno ; Neumuth, Thomas ; Padoy, Nicolas ; Pugh, Carla ; Schoch, Nicolai ; Stoyanov, Danail ; Taylor, Russell ; Wagner, Martin ; Hager, Gregory D. ; Jannin, Pierre: Surgical data science for next-generation interventions. In: *Nature biomedical engineering* 1 (2017), January, Nr. 9, p. 691–696. http://dx.doi.org/10.1038/s41551-017-0132-7. – DOI 10.1038/s41551–017–0132–7

[Mic16] Microsoft: *The STRIDE Threat Model*. https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20)?redirectedfrom=MSDN. Version: 2016, Accessed 12.07.2022

[Mil14] Mildner, Alexander: *Whitepaper: Risikoanalyse dynamisch vernetzter Medizingeräte*. 2014

[Min22] Mindray Global: *HyBase V8 - Mindray Global*. https://www.mindray.com/en/products-solutions/products/operating-tables/hybase-v8. Version: 06.02.2022, Accessed 06.02.2022

[Mit97] Mitchell, Tom M.: *Machine Learning*. New York and London : McGraw-Hill, 1997 (McGraw-Hill series in computer science). – ISBN 0070428077

[MJJ15] Mildner, Alexander ; Janss, Armin ; Jasmin dell'Anna: Geräte- und Serviceprofile dynamisch vernetzter Systeme: Update 2015 / OR.NET. 2015 (Milestone 3.4.1). – Research Report. – Whitepaper

[MKA16] Maruyama, Yuya ; Kato, Shinpei ; Azumi, Takuya: Exploring the performance of ROS2. In: *Proceedings of the 13th International Conference on Embedded Software*. New York, NY, USA : ACM, 102016. – ISBN 9781450344852, p. 1–10

[MMR+11] Matthias Borowski ; Matthias Görges ; Roland Fried ; Olaf Such ; Christian Wrede ; Michael Imhoff: Medical device alarms. 56 (2011), Nr. 2, p. 73–83. http://dx.doi.org/10.1515/bmt.2011.005. – DOI 10.1515/bmt.2011.005

[MN19] Mark, Richards ; Neal, Ford: *Fundamentals of Software Architecture : An Engineering Approach*. vol. First edition. O'Reilly Media, 2019. – ISBN 9781492043454

[Moo65] Moore, Gordon E.: Cramming more components onto integrated circuits. 38 (1965), Nr. 8

[MTK19] Mbakoyiannis, Dimitris ; Tomoutzoglou, Othon ; Kornaros, George: Secure over-the-air firmware updating for automotive electronic control units. In: Hung, Chih-Cheng (Ed.) ; Papadopoulos, George A. (Ed.): *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*. New York, NY, USA : ACM, 2019. – ISBN 9781450359337, p. 174–181

[MU15] Morana, Marco M. ; Uceda Vélez, Tony: *Risk centric threat modeling: Process for attack simulation and threat analysis / Tony Ucedavélez and Marco M. Morana*. Hoboken, New Jersey : John Wiley & Sons, 2015. – ISBN 978–0–470–50096–5

[MV14] Miller, Charlie ; Vasalek, Chris: *A survey of remote automotive attack surfaces*. 2014

[MVK+20] Miclăuş, Teodora ; Valla, Vasiliki ; Koukoura, Angeliki ; Nielsen, Anne A. ; Dahlerup, Benedicte ; Tsianos, Georgios-Ioannis ; Vassiliadis, Efstathios: Impact of Design on Medical Device Safety. In: *Therapeutic innovation & regulatory science* 54 (2020), Nr. 4, p. 839–849. http://dx.doi.org/10.1007/s43441-019-00022-4. – DOI 10.1007/s43441–019–00022–4

[NA10] Najjar, Meisam S. ; Abdollahi Azgomi, Mohammad: A distributed multi-approach intrusion detection system for web services. In: Makarevich, Oleg (Ed.): *Proceedings of the 3rd international conference on Security of information and networks*. New York, NY : ACM, 2010 (ACM Other conferences). – ISBN 9781450302340, p. 238

[Nat11] National Electrical Manufacturers Association: *DICOM PS 3.18 2011 - Web Access to DICOM Persistent Objects (WADO)*. 2011

[Nat18] National Institute of Standards and Technology: *Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1*

[NDA12] Nayebi, F. ; Desharnais, J-M ; Abran, A.: The state of the art of mobile application usability evaluation. In: *2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, IEEE, 2012. – ISBN 978–1–4673–1433–6, p. 1–4

[Neu17] Neumuth, Thomas: Surgical process modeling. In: *Innovative surgical sciences* 2 (2017), Nr. 3, p. 123–137. http://dx.doi.org/10.1515/iss-2017-0005. – DOI 10.1515/iss–2017–0005

[NH20] NITRD Program ; HITRD IWG: *The Interoperability of Medical Devices, Data, and Platforms to Enhance Patient Care: A summary of the February 2019 Request for Information and July 2019 Listening Session*. 2020

[NHF+12] Nollert, Georg ; Hartkens, Thomas ; Figel, Anne ; Bulitta, Clemens ; Altenbeck, Franziska ; Gerhar, Vanessa: The Hybrid Operating Room. Version: 2012.

http://dx.doi.org/10.5772/27599. In: Narin, Cuneyt (Ed.): *Special Topics in Cardiac Surgery*. Vienna, Austria : InTech, 2012. – DOI 10.5772/27599. – ISBN 978–953–51–0148–2, p. 73–107

[NKP+23] Narula, Sanjiv ; Kumar, Anil ; Prakash, Surya ; Dwivedy, Maheshwar ; Puppala, Harish ; Talwar, Vishal: Modelling and Analysis of Challenges for Industry 4.0 Implementation in Medical Device Industry to Post COVID -19 Scenario. In: *International Journal of Supply and Operations Management* 10 (2023), Nr. 2, 117–135. http://dx.doi.org/10.22034/ijsom.2021.2838. – DOI 10.22034/ijsom.2021.2838. – ISSN 23831359

[NM07] Nadkarni, Prakash M. ; Miller, Randolph A.: Service-oriented architecture in medical software: promises and perils. In: *Journal of the American Medical Informatics Association : JAMIA* 14 (2007), Nr. 2, p. 244–246. http://dx.doi.org/10.1197/jamia.M2349. – DOI 10.1197/jamia.M2349. – ISSN 1067–5027

[NSN08] Nilsson, Dennis K. ; Sun, Lei ; Nakajima, Tatsuo: A Framework for Self-Verification of Firmware Updates over the Air in Vehicle ECUs. In: *2008 IEEE Globecom Workshops*, IEEE, 2008. – ISBN 978–1–4244–3061–1, p. 1–5

[NWL+15] Navale, Varun M. ; Williams, Kyle ; Lagospiris, Athanassios ; Schaffert, Michael ; Schweiker, Markus-Alexander: (R)evolution of E/E Architectures. In: *SAE International Journal of Passenger Cars - Electronic and Electrical Systems* 8 (2015), Nr. 2, p. 282–288. http://dx.doi.org/10.4271/2015-01-0196. – DOI 10.4271/2015–01–0196. – ISSN 1946–4622

[OAS06] OASIS: *Reference Model for Service Oriented Architecture v1.0*. http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.html. Version: 2006, Accessed 27.01.2022

[OAS19a] OASIS: *XACML REST Profile Version 1.1*. https://docs.oasis-open.org/xacml/xacml-rest/v1.1/xacml-rest-v1.1.html. Version: v 1.1, 20 June 2019, Accessed 06.03.2022

[OAS19b] OASIS: *MQTT Version 5.0*. http://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html. Version: 2019, Accessed 27.01.2022

[Obe21] Obergfell, Philipp: *Entwurfsmethodik für hybride Software- und Systemarchitektur*. Karlsruhe, Karlsruhe Institute for Technology, Dissertation, 2021

[OFF24] OFFIS e.V.: *PoCSpec – Modular Specialisations for Point-of-Care Medical Devices*. https://pocspec.de/. Version: 17.01.2024, Accessed 17.01.2024

[OJB+20] Ondrej Burkacky ; Johannes Deichmann ; Benjamin Klein ; Klaus Pototzky ; Gundbert Scherf: *Cybersecurity in automotive: Mastering the challenge*. https://www.gsaglobal.org/wp-content/uploads/2020/03/Cybersecurity-in-automotive-Mastering-the-challenge.pdf. Version: 2020, Accessed 19.04.2022

[OMIM18] Okamoto, Jun ; Masamune, Ken ; Iseki, Hiroshi ; Muragaki, Yoshihiro: Development concepts of a Smart Cyber Operating Theater (SCOT) using ORiN

technology. In: *Biomedizinische Technik. Biomedical engineering* 63 (2018), Nr. 1, p. 31–37. http://dx.doi.org/10.1515/bmt-2017-0006. – DOI 10.1515/bmt–2017–0006

[OON13] Oliveira, Lucas Bueno R. ; Osório, Fernando S. ; Nakagawa, Elisa Y.: An investigation into the development of service-oriented robotic systems. In: Shin, Sung Y. (Ed.) ; Maldonado, José C. (Ed.): *Proceedings of the 28th Annual ACM Symposium on Applied Computing - SAC '13*. New York, New York, USA : ACM Press, 2013. – ISBN 9781450316569, p. 223

[OPC15] OPC Foundation: *Home Page - OPC Foundation*. https://opcfoundation.org/. Version: 28.10.2015, Accessed 27.11.2021

[Ope22a] Open Robotics: *About Quality of Service settings — ROS 2 Documentation: Galactic documentation*. https://docs.ros.org/en/galactic/Concepts/About-Quality-of-Service-Settings.html. Version: 05.03.2022, Accessed 06.03.2022

[ope22b] opeXpark: *OPeLiNK | Product | OPExPARK Inc*. https://www.opexpark.co.jp/en/product/opelink/. Version: 2022, Accessed 19.11.2022

[OR.19] OR.NET e.V.: *MoVE-Projekt*. https://ornet.org/services-2-3/services-2-5/services-2-3-3/. Version: 2019, Accessed 17.01.2024

[OSR24] OSRF, Open Source Robotics F.: *Features – Gazebo*. https://gazebosim.org/features. Version: 2024, Accessed 19.05.2024

[PAK+18] Peters, Brian S. ; Armijo, Priscila R. ; Krause, Crystal ; Choudhury, Songita A. ; Oleynikov, Dmitry: Review of emerging surgical robotic technology. In: *Surgical endoscopy* 32 (2018), Nr. 4, p. 1636–1655. http://dx.doi.org/10.1007/s00464-018-6079-2. – DOI 10.1007/s00464–018–6079–2

[PB17] Puente León, Fernando ; Bauer, Sebastian: *Praxis der Digitalen Signalverarbeitung*. 2., überarbeitete Auflage. Karlsruhe : KIT Scientific Publishing, 2017. http://dx.doi.org/10.5445/KSP/1000067012. http://dx.doi.org/10.5445/KSP/1000067012. – ISBN 9783731506515

[PBDH16] Pohl, Klaus ; Broy, Manfred ; Daembkes, Heinrich ; Hönninger, Harald: *Advanced Model-Based Engineering of Embedded Systems*. Cham : Springer International Publishing, 2016. http://dx.doi.org/10.1007/978-3-319-48003-9. http://dx.doi.org/10.1007/978-3-319-48003-9. – ISBN 978–3–319–48002–2

[PDDL15] Pfeiffer, Jonas H. ; Dingler, Max E. ; Dietz, Christian ; Lueth, Tim C.: Requirements and architecture design for open real-time communication in the operating room. In: *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, IEEE, 2015. – ISBN 978–1–4673–9675–2, p. 458–463

[Pev16] Pevný, Tomáš: Loda: Lightweight on-line detector of anomalies. In: *Machine Learning* 102 (2016), Nr. 2, p. 275–304. http://dx.doi.org/10.1007/s10994-015-5521-0. – DOI 10.1007/s10994–015–5521–0. – ISSN 0885–6125

[PGM+19] Paszke, Adam ; Gross, Sam ; Massa, Francisco ; Lerer, Adam ; Bradbury, James ; Chanan, Gregory ; Killeen, Trevor ; Lin, Zeming ; Gimelshein, Natalia ; Antiga, Luca ; Desmaison, Alban ; Kopf, Andreas ; Yang, Edward ; DeVito, Zachary ; Raison, Martin ; Tejani, Alykhan ; Chilamkurthy, Sasank ; Steiner, Benoit ; Fang, Lu ; Bai, Junjie ; Chintala, Soumith: PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Wallach, H. (Ed.) ; Larochelle, H. (Ed.) ; Beygelzimer, A. (Ed.) ; Alché-Buc, F. d'(Ed.) ; Fox, E. (Ed.) ; Garnett, R. (Ed.): *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, 8024–8035

[Phi18] Philips: *The rise of the digital twin: how healthcare can benefit*. https://www.philips.com/a-w/about/news/archive/blogs/innovation-matters/20180830-the-rise-of-the-digital-twin-how-healthcare-can-benefit.html. Version: 2018, Accessed 22.03.2022

[Pil16] Pilieci, Vito: *Ottawa Hospital hit with ransomware, information on four computers locked down | Ottawa Citizen*. https://ottawacitizen.com/news/local-news/ottawa-hospital-hit-with-ransomware-information-on-four-computers-locked-down/. Version: 2016, Accessed 21.12.2022

[PK13] Puente León, Fernando ; Kiencke, U.: *Ereignisdiskrete Systeme: Modellierung und Steuerung verteilter Systeme*. 3., vollständig überarbeitete und ergänzte Auflage. München : Oldenbourg Verlag München, 2013. – ISBN 9783486735741

[PK22] Peller-Konrad, Fabian: *H^2T Forschung - Roboter - Software- und Reglungsarchitektur*. https://h2t.anthropomatik.kit.edu/401.php. Version: 21.02.2022, Accessed 21.02.2022

[PKM+20] Papaioannou, Maria ; Karageorgou, Marina ; Mantas, Georgios ; Sucasas, Victor ; Essop, Ismael ; Rodriguez, Jonathan ; Lymberopoulos, Dimitrios: A Survey on Security Threats and Countermeasures in Internet of Medical Things (IoMT). In: *Transactions on Emerging Telecommunications Technologies* (2020). http://dx.doi.org/10.1002/ett.4049. – DOI 10.1002/ett.4049. – ISSN 2161–3915

[Pon22] Ponemon Institute: *Cyber Insecurity In Healthcare: The Cost And Impact On Patient Safety And Care*. https://www.proofpoint.com/us/cyber-insecurity-in-healthcare. Version: 2022

[PPE+23] Pottier, C ; Petzing, J ; Eghtedari, F ; Lohse, N ; Kinnell, P: Developing digital twins of multi-camera metrology systems in Blender. In: *Measurement Science and Technology* 34 (2023), July, Nr. 7, 075001. http://dx.doi.org/10.1088/1361-6501/acc59e, Accessed 24.05.2024. – DOI 10.1088/1361–6501/acc59e. – ISSN 0957–0233, 1361–6501

[PTD+19] Profanter, Stefan ; Tekat, Ayhun ; Dorofeev, Kirill ; Rickert, Markus ; Knoll, Alois: OPC UA versus ROS, DDS, and MQTT: Performance Evaluation of Industry 4.0 Protocols. In: *2019 IEEE International Conference on Industrial Technology (ICIT)*, IEEE, 2019. – ISBN 978–1–5386–6376–9, p. 955–962

[PWG+15] Prince, Martin J. ; Wu, Fan ; Guo, Yanfei ; Gutierrez Robledo, Luis M. ; O'Donnell, Martin ; Sullivan, Richard ; Yusuf, Salim: The burden of disease in older people and implications for health policy and practice. In: *The Lancet* 385 (2015), Nr. 9967, p. 549–562. http://dx.doi.org/10.1016/S0140-6736(14)61347-7. – DOI 10.1016/S0140–6736(14)61347–7. – ISSN 01406736

[QCG+09] Quigley, Morgan ; Conley, Ken ; Gerkey, Brian ; Faust, Josh ; Foote, Tully ; Leibs, Jeremy ; Wheeler, Rob ; Ng, Andrew: ROS: an open-source Robot Operating System. In: *ICRA workshop on open source software* vol. 3. Kobe, Japan : IEEE, January 2009, p. 5

[QGW+23] Qiao, Mengmeng ; Guo, Jiayu ; Wang, Rui ; Chen, Chong ; Li, Jing ; Lyu, Jun: Research progress on population aging and chronic diseases. In: *MEDS Public Health and Preventive Medicine* 3 (2023), Nr. 1, p. 28–35. http://dx.doi.org/10.23977/phpm.2023.030105. – DOI 10.23977/phpm.2023.030105

[RAMG20] Rajivan, Prashanth ; Aharonov-Majar, Efrat ; Gonzalez, Cleotilde: Update now or later? Effects of experience, cost, and risk preference on update decisions. In: *Journal of Cybersecurity* 6 (2020), Nr. 1. http://dx.doi.org/10.1093/cybsec/tyaa002. – DOI 10.1093/cybsec/tyaa002. – ISSN 2057–2085

[RBMC20] Rose, Scott ; Borchert, Oliver ; Mitchell, Stu ; Connelly, Sean: *Zero Trust Architecture*. – National Institute of Standards and Technology

[RCS+23] Ramesh, Goutham B. ; Chamas, Mohamad ; Schindewolf, Marc ; Kraus, David ; Sax, Eric: Plug-and-Play Feature for Automotive Camera Sensors Using ReCoIN Model With Smart Configuration. In: *2023 12th International Conference on Control, Automation and Information Sciences (ICCAIS)*, IEEE, 2023. – ISBN 979–8–3503–2878–3, p. 725–732

[Rea15] Real-Time Innovations Inc: *RTI® Connext™ DDS: Comprehensive Summary of QoS Policies*. https://community.rti.com/static/documentation/connext-dds/5.2.0/doc/manuals/connext_dds/RTI_ConnextDDS_CoreLibraries_QoS_Reference_Guide.pdf. Version: 2015, Accessed 06.03.2022

[Rea19] Reardon, Sara: Rise of Robot Radiologists. In: *Nature* 576 (2019), Nr. 7787, p. 54–58. http://dx.doi.org/10.1038/d41586-019-03847-z. – DOI 10.1038/d41586–019–03847–z

[RGKS20] Rumez, Marcel ; Grimm, Daniel ; Kriesten, Reiner ; Sax, Eric: An Overview of Automotive Service-Oriented Architectures and Implications for Security Countermeasures. In: *IEEE Access* 8 (2020), p. 221852–221870. http://dx.doi.org/10.1109/ACCESS.2020.3043070. – DOI 10.1109/ACCESS.2020.3043070

[RGRS+18] Rassweiler, J. J. ; Goezen, A. S. ; Rassweiler-Seyfried, M. C. ; Liatsikos, E. ; Bach, T. ; Stolzenburg, J-U ; Klein, J.: Der Roboter in der Urologie – eine Analyse aktueller und zukünftiger Gerätegenerationen. In: *Der Urologe. Ausg. A* 57 (2018),

Nr. 9, p. 1075–1090. http://dx.doi.org/10.1007/s00120-018-0733-0. – DOI 10.1007/s00120–018–0733–0

[RKG+19] Ratasich, Denise ; Khalid, Faiq ; Geissler, Florian ; Grosu, Radu ; Shafique, Muhammad ; Bartocci, Ezio: A Roadmap Toward the Resilient Internet of Things for Cyber-Physical Systems. In: *IEEE Access* 7 (2019), p. 13260–13283. http://dx.doi.org/10.1109/ACCESS.2019.2891969. – DOI 10.1109/ACCESS.2019.2891969

[RLF+20] Rumez, Marcel ; Lin, Jinghua ; Fuchß, Thomas ; Kriesten, Reiner ; Sax, Eric: Anomaly Detection for Automotive Diagnostic Applications Based on N-Grams. In: *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, IEEE, 2020. – ISBN 978–1–7281–7303–0, p. 1423–1429

[RLSS10] Rajkumar, Ragunathan ; Lee, Insup ; Sha, Lui ; Stankovic, John: Cyber-physical systems. In: Sapatnekar, Sachin (Ed.): *Proceedings of the 47th Design Automation Conference on - DAC '10*. New York, New York, USA : ACM Press, 2010. – ISBN 9781450300025, p. 731

[RMB+22] Rbah, Yahya ; Mahfoudi, Mohammed ; Balboul, Younes ; Fattah, Mohammed ; Mazer, Said ; Elbekkali, Moulhime ; Bernoussi, Benaissa: Machine Learning and Deep Learning Methods for Intrusion Detection Systems in IoMT: A survey. In: *2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, IEEE, 2022. – ISBN 978–1–6654–2209–3, p. 1–9

[RNC+22] Retana, Manuel ; Nalamwar, Kunal ; Conyers, Drew T. ; Atashzar, S. F. ; Alambeigi, Farshid: Autonomous Data-Driven Manipulation of an Unknown Deformable Tissue Within Constrained Environments: A Pilot Study. In: *2022 International Symposium on Medical Robotics (ISMR)*, IEEE, 2022. – ISBN 978–1–6654–6928–9, p. 1–7

[Rob11] Robot Mag: *Service Oriented Architectures – Two leading systems, MRDS and ROS, point to the future of robotics*. http://www.botmag.com/service-oriented-architectures-two-leading-systems-mrds-and-ros-point-to-the-future-of-robotics/. Version: 2011, Accessed 09.05.2021

[Rob17] Robertazzi, Thomas G.: *Introduction to Computer Networking*. Cham : Springer International Publishing, 2017. http://dx.doi.org/10.1007/978-3-319-53103-8. http://dx.doi.org/10.1007/978-3-319-53103-8. – ISBN 978–3–319–53102–1

[Rob18] Robotics, Open: *Joint Trajectory Controllerr - ROS Wiki*. http://wiki.ros.org/joint_trajectory_controller. Version: 2018, Accessed 09.08.2024

[Rob20] Robotics, Open: *Effort Controllers - ROS Wiki*. http://wiki.ros.org/effort_controllers. Version: 2020, Accessed 09.08.2024

[Rob21] Robert Bosch GmbH: *CAN XL – The next Step in CAN Evolution*. 25.02.2021

[Rob23] Robotics, Open: *ROS1 Bridge*. https://ros1-bridge.readthedocs.io/en/latest/index.html. Version: 2023, Accessed 09.08.2024

[Rod09] Rodrigues, J.J.P.C.: *Health Information Systems: Concepts, Methodologies, Tools, and Applications: Concepts, Methodologies, Tools, and Applications*. IGI Global, 2009 (Premier reference source). https://books.google.de/books?id=WnBJsRtfVbYC. – ISBN 9781605669892

[Ros57] Rosenblatt, Frank: The perceptron – A perceiving and recognizing automaton. (1957), Nr. Report No. 85-460-1

[ROS22] ROS micro: *micro-ROS Homepage*. https://micro.ros.org/. Version: 31.05.2022, Accessed 04.06.2022

[RS88] Roberson, Robert E. ; Schwertassek, Richard: *Dynamics of multibody systems*. Berlin : Springer, 1988. – ISBN 3540174478

[RSZ21] Rausch, Andreas ; Sedeh, Azarmidokht M. ; Zhang, Meng: Autoencoder-Based Semantic Novelty Detection: Towards Dependable AI-Based Systems. In: *Applied Sciences* 11 (2021), Nr. 21. http://dx.doi.org/10.3390/app11219881. – DOI 10.3390/app11219881

[Rum22] Rumez, Marcel: *Absicherung von Diagnosefunktionen in E/E-Fahrzeugarchitekturen durch verteilte Zugriffskontrolle und Anomalieerkennung*. Karlsruhe, Karlsruhe Institute for Technology, Dissertation, 08.11.2022

[Rup21] Rupp, Chris: *Requirements-Engineering und -Management: Das Handbuch für Anforderungen in jeder Situation*. 7., aktualisierte und erweiterte Auflage. München : Hanser, 2021. – ISBN 9783446455870

[RWL08] Ryherd, Erica E. ; Waye, Kerstin P. ; Ljungkvist, Linda: Characterizing noise and perceived work environment in a neurological intensive care unit. In: *The Journal of the Acoustical Society of America* 123 (2008), Nr. 2, p. 747–756. http://dx.doi.org/10.1121/1.2822661. – DOI 10.1121/1.2822661. – ISSN 0001–4966

[Sam59] Samuel, A. L.: Some Studies in Machine Learning Using the Game of Checkers. In: *IBM Journal of Research and Development* 3 (1959), Nr. 3, p. 210–229. http://dx.doi.org/10.1147/rd.33.0210. – DOI 10.1147/rd.33.0210. – ISSN 0018–8646

[SBHS06] Saad, Alexandre ; Bauer, Werner ; Haneberg, Michael ; Schiffers, Jutta: Intelligent Automotive System Services - An Emerging Design Pattern for an Advanced E/E-Architecture. Version: 2006. http://dx.doi.org/10.4271/2006-01-1286. SAE International400 Commonwealth Drive, Warrendale, PA, United States, 2006. (SAE Technical Paper Series). – Research Report. – Technical Paper

[SBT18] Sandberg, Christian ; Bokesand, Andreas ; Thorsson, Urban: *HoliSec Deliverable D4.1.1: Tailoring the HEAVENS risk assessment methodology for improved.* 2018

[Sch08] Schrempf, Zauner: *Informatik in der Medizintechnik: Grundlagen, Software, Computergestützte Systeme*. Springer, 2008

[Sch14] Schneider, Stan: *The Internet Of Things Can Save 50,000 Lives A Year*. 2014

[SCO+18] Shevchenko, Nataliya ; Chick, Timothy A. ; O'Riordan, Paige ; P. Scanlon, Thomas ; Woody, Carol: *Threat Modeling: A Summary of Available Methods*. 2018

[SCSH20] Sanchez-Comas, Andres ; Synnes, Kåre ; Hallberg, Josef: Hardware for Recognition of Human Activities: A Review of Smart Home and AAL Related Technologies. In: *Sensors* 20 (2020), July, Nr. 15, 4227. http://dx.doi.org/10.3390/s20154227, Accessed 08.06.2024. – DOI 10.3390/s20154227. – ISSN 1424–8220

[Sel13] Seligman, P.: Industry Surveys: Healthcare: Products & Supplies. (2013)

[SF13] Sendelbach, Sue ; Funk, Marjorie: Alarm Fatigue. In: *AACN Advanced Critical Care* 24 (2013), Nr. 4, p. 378–386. http://dx.doi.org/10.4037/NCI.0b013e3182a903f9. – DOI 10.4037/NCI.0b013e3182a903f9. – ISSN 1559–7768

[SGG+20] Scheer, Dietmar ; Glodd, Oliver ; Günther, Hartmut ; Duhr, Yves ; Schmid, Achim: STAR3 - Eine neue Generation der E/E-Architektur. In: *Sonderprojekte ATZ/MTZ* 25 (2020), Nr. S1, p. 72–79. http://dx.doi.org/10.1007/s41491-020-0056-5. – DOI 10.1007/s41491–020–0056–5. – ISSN 2509–4602

[SGGR19] Sango, Marc (Ed.) ; Godot, Jean (Ed.) ; Gonzalez, Antonio (Ed.) ; Ruiz Nolasco, Ricardo (Ed.): *Model-Based System, Safety and Security Co-Engineering Method and Toolchain for Medical Devices Design*. vol. *2019 Design of Medical Devices Conference*. 2019 (Frontiers in Biomedical Devices)

[SGLS22] Schindewolf, Marc ; Grimm, Daniel ; Lingor, Christian ; Sax, Eric: Toward a Resilient Automotive Service-Oriented Architecture by using Dynamic Orchestration. In: *2022 IEEE 1st International Conference on Cognitive Mobility (CogMob)*, 2022, p. 000147–000154

[SGS+21a] Stoll, Hannes ; Grimm, Daniel ; Schindewolf, Marc ; Brodatzki, Michel ; Sax, Eric: Dynamic Reconfiguration of Automotive Architectures Using a Novel Plug-and-Play Approach. In: *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*, IEEE, 2021. – ISBN 978–1–6654–7921–9, p. 70–75

[SGS+21b] Stoll, Hannes ; Grimm, Daniel ; Schindewolf, Marc ; Brodatzki, Michel ; Sax, Eric: Dynamic Reconfiguration of Automotive Architectures Using a Novel Plug-and-Play Approach. In: *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*. Nagoya, Japan : IEEE, 2021, p. 70–75

[SH09] Scarfone, Karen ; Hoffmann, Paul: *Guidelines on Firewalls and Firewall Policy: NIST Special Publication 800*. https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=901083. Version: 2009

[SHG+21] Strathmann, Thomas ; Hake, Georg ; Guissouma, Houssem ; Hohl, Carl P. ; Bebawy, Yosab ; Maelen, Sebastian V. ; Koerner, Andrew: Project Overview for Step-Up!CPS - Process, Methods and Technologies for Updating Safety-critical

Cyber-physical Systems. In: *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, IEEE, 2021. – ISBN 978–3–9819263–5–4, p. 1326–1329

[S.I21] S.I.M.E.O.N. Medical GmbH & Co KG: *Sim.MOVE 800*. https://www.simeonmedical.com/en/products/operating-table/simmove-800/. Version: 2021, Accessed 06.02.2022

[SJT15] Stefan Franke ; Jürgen Meixensberger ; Thomas Neumuth: Multi-perspective workflow modeling for online surgical situation models. In: *Journal of Biomedical Informatics* 54 (2015), 158–166. http://dx.doi.org/10.1016/j.jbi.2015.02.005. – DOI 10.1016/j.jbi.2015.02.005. – ISSN 1532–0464

[SK16] Siciliano, Bruno ; Khatib, Oussama: *Springer Handbook of Robotics*. Cham : Springer International Publishing, 2016. http://dx.doi.org/10.1007/978-3-319-32552-1. http://dx.doi.org/10.1007/978-3-319-32552-1. – ISBN 978–3–319–32550–7

[SKKS10] Stolz, Wolfgang ; Kornhaas, Robert ; Krause, Ralph ; Sommer, Tino: Domain Control Units - the Solution for Future E/E Architectures? In: *SAE Technical Paper Series*, SAE International400 Commonwealth Drive, Warrendale, PA, United States, 2010 (SAE Technical Paper Series), p. 7

[SKR18] Shafique, Muhammad ; Khalid, Faiq ; Rehman, Semeen: Intelligent Security Measures for Smart Cyber Physical Systems. In: *2018 21st Euromicro Conference on Digital System Design (DSD)*, IEEE, 2018. – ISBN 978–1–5386–7377–5, p. 280–287

[SOMM21] Sun, Xiao ; Okamoto, Jun ; Masamune, Ken ; Muragaki, Yoshihiro: Robotic Technology in Operating Rooms: a Review. In: *Current Robotics Reports* (2021). http://dx.doi.org/10.1007/s43154-021-00055-4. – DOI 10.1007/s43154–021–00055–4

[SOP21] SOPHIST-Gesellschaft für Innovatives Software-Engineering: *Requirements-Engineering und -Management (7th Edition) | SOPHIST GmbH - Gliederung Nicht-funktionaler Anforderungen*. https://www.sophist.de/fileadmin/user_upload/Bilder_zu_Seiten/Publikationen/RE7/Webinhalte_BT5/Gliederung_Nicht-funktionaler_Anforderungen.pdf. Version: 2021, Accessed 27.05.2022

[Spa24] Sparx Systems: *Activity Diagram | Enterprise Architect User Guide*. https://www.sparxsystems.com/enterprise_architect_user_guide/13.5/model_domains/activitydiagram.html. Version: 2024, Accessed 20.01.2024

[ST12] Streichert, Thilo ; Traub, Matthias: *Elektrik/Elektronik-Architekturen im Kraftfahrzeug*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2012. http://dx.doi.org/10.1007/978-3-642-25478-9. http://dx.doi.org/10.1007/978-3-642-25478-9. – ISBN 978–3–642–25477–2

[St.22] St. Antonius-Hospital: *Center for Robotic Medicine Germany (CRMG) - da Vinci® Zentrum Gronau - St. Antonius-Hospital Gronau*. https://www.st-antonius-gronau.de/medizinische-angebote/zentren/crmg/. Version: 10.03.2022, Accessed 10.03.2022

[Sta16] Stanganelli, Joe: Selecting a Threat Risk Model for Your Organization, Part Two. In: *eSecurityPlanet* (27.9.2016). https://www.esecurityplanet.com/networks/selecting-a-threat-risk-model-for-your-organization-part-two/, Accessed 17.12.2022

[Sta17] Staron, Miroslaw: *Automotive Software Architectures*. Cham : Springer International Publishing, 2017. http://dx.doi.org/10.1007/978-3-319-58610-6. http://dx.doi.org/10.1007/978-3-319-58610-6. – ISBN 978–3–319–58609–0

[Sta18] Starke, Gernot: *Effektive Softwarearchitekturen: Ein praktischer Leitfaden*. 8., überarbeitete Auflage. München : Hanser, 2018. – ISBN 9783446452077

[STB+18] Shafique, Muhammad ; Theocharides, Theocharis ; Bouganis, Christos-Savvas ; Hanif, Muhammad A. ; Khalid, Faiq ; Hafiz, Rehan ; Rehman, Semeen: An overview of next-generation architectures for machine learning: Roadmap, opportunities and challenges in the IoT era. In: *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, IEEE, 2018. – ISBN 978–3–9819263–0–9, p. 827–832

[Ste18] Steinbach, Till: *Ethernet-basierte Fahrzeugnetzwerkarchitekturen für zukünftige Echtzeitsysteme im Automobil*. Wiesbaden : Springer Fachmedien Wiesbaden, 2018. http://dx.doi.org/10.1007/978-3-658-23500-0. http://dx.doi.org/10.1007/978-3-658-23500-0. – ISBN 978–3–658–23499–7

[Ste22] Steris: *OT 1000 Series Orthopedic Surgical Table*. https://www.steris.com/healthcare/products/surgical-tables/steris-ot-1000-series-orthopedic-surgical-table. Version: 2022, Accessed 28.03.2022

[Sto21] Stoll, Hannes Frank: *Die (re-)konfigurierbare Fahrzeugarchitektur*. Karlsruhe, Karlsruhe Institute for Technology, Dissertation, 2021

[Str21] Stryker: *Stryker iSuitel*. https://www.stryker.com/us/en/communications/products/isuite-powered-by-connected-or-operating-system.html. Version: 2021, Accessed 11.03.2021

[SUD21] SUD, TUV: *FAQs: Functional safety in medical devices*. https://www.tuvsud.com/en/industries/healthcare-and-medical-devices/medical-devices-and-ivd/medical-device-testing/physical-testing-of-medical-devices/functional-safety/faqs. Version: 30.05.2021, Accessed 30.05.2021

[Sur22] Surgitaix AG: *SurgiTAIX AG Website*. https://surgitaix.com/wp/services-2/#/integrationservices. Version: 2022, Accessed 22.10.2022

[SVDP12] Sangiovanni-Vincentelli, Alberto ; Damm, Werner ; Passerone, Roberto: Taming Dr. Frankenstein: Contract-Based Design for Cyber-Physical Systems*. In: *European Journal of Control* 18 (2012), Nr. 3, p. 217–238. http://dx.doi.org/10.3166/ejc.18.217-238. – DOI 10.3166/ejc.18.217–238. – ISSN 09473580

[SWLP11] Stolz, Wolfgang ; Williams, Kyle ; Lorenz, Tobias ; Piastowski, Martin: Ethernet and IP - The Solution to Master Complexity, Safety and Security in Vehicle Communication Networks? In: *Ethernet and IP*, SAE International400 Commonwealth Drive, Warrendale, PA, United States, 2011 (SAE Technical Paper Series), p. 11

[SWP22] Schmidl, Sebastian ; Wenig, Phillip ; Papenbrock, Thorsten: Anomaly detection in time series. In: *Proceedings of the VLDB Endowment* 15 (2022), Nr. 9, p. 1779–1797. http://dx.doi.org/10.14778/3538598.3538602. – DOI 10.14778/3538598.3538602. – ISSN 2150–8097

[SYAY22] Sadhu, Pintu K. ; Yanambaka, Venkata P. ; Abdelgawad, Ahmed ; Yelamarthi, Kumar: Prospect of Internet of Medical Things: A Review on Security Requirements and Solutions. In: *Sensors (Basel, Switzerland)* 22 (2022), Nr. 15. http://dx.doi.org/10.3390/s22155517. – DOI 10.3390/s22155517

[SZ16] Schäuffele, Jörg ; Zurawka, Thomas: *Automotive Software Engineering*. Wiesbaden : Springer Fachmedien Wiesbaden, 2016. http://dx.doi.org/10.1007/978-3-658-11815-0. http://dx.doi.org/10.1007/978-3-658-11815-0. – ISBN 978–3–658–11814–3

[Tec22] Technischer Überwachungsverein (TÜV) SÜD: *Prüfung von Medizinprodukten nach IEC 60601-1*. https://www.tuvsud.com/de-de/branchen/gesundheit-und-medizintechnik/pruefung-von-medizintechnik/pruefung-nach-iec-60601. Version: 23.01.2022, Accessed 23.01.2022

[Tec23] Technik, Medizin ⋮ *OR.Net: Für Vernetzung im OP – SDC auch mit Echtzeit-Datenuebertragung*. https://medizin-und-technik.industrie.de/digitalisierung/or-net-fuer-vernetzung-im-op-sdc-auch-mit-echtzeit-datenuebertragung/. Version: 2023, Accessed 09.05.2024

[TEMH+20] Teber, D. ; Engels, C. ; Maier-Hein, L. ; Ayala, L. ; Onogur, S. ; Seitel, A. ; März, K.: Wie weit ist Chirugie 4.0? In: *Der Urologe. Ausg. A* 59 (2020), Nr. 9, p. 1035–1043. http://dx.doi.org/10.1007/s00120-020-01272-z. – DOI 10.1007/s00120–020–01272–z

[Ten23] TensorFlow: *TensorFlow*. https://www.tensorflow.org/. Version: 09.11.2023, Accessed 08.12.2023

[The17] Theissler, Andreas: Detecting known and unknown faults in automotive systems using ensemble-based anomaly detection. In: *Knowledge-Based Systems* 123 (2017), p. 163–173. http://dx.doi.org/10.1016/j.knosys.2017.02.023. – DOI 10.1016/j.knosys.2017.02.023. – ISSN 09507051

[The20] The Government of Japan - JapanGov -: *Cutting-edge Operating Theater Connected by IoT / The Government of Japan - JapanGov - .* https://www.japan.go.jp/tomodachi/2020/earlysummer2020/smart_treatment_room.html. Version: 19.11.2020, Accessed 26.06.2021

[Tis18] Tischer, Mirko: *AUTOSAR Adaptive - The Computing Center in the Vehicle.* https://assets.vector.com/cms/content/know-how/technical-articles/AUTOSAR/AUTOSARAdaptiveElektronikAutomotive201809PressArticleEN.pdf. Version: 2018, Accessed 06.05.2023

[TMB17] Traub, Matthias ; Maier, Alexander ; Barbehon, Kai L.: Future Automotive Architecture and the Impact of IT Trends. In: *IEEE Software* 34 (2017), Nr. 3, p. 27–32. http://dx.doi.org/10.1109/MS.2017.69. – DOI 10.1109/MS.2017.69. – ISSN 0740–7459

[TMR20] Toshniwal, Akanksha ; Mahesh, Kavi ; R., Jayashree: Overview of Anomaly Detection techniques in Machine Learning. In: *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. Palladam, India : IEEE, October 2020. – ISBN 978–1–72815–464–0, 808–815

[TS18] Tuma, Katja ; Scandariato, Riccardo: Two Architectural Threat Analysis Techniques Compared. In: Cuesta, Carlos E. (Ed.) ; Garlan, David (Ed.) ; Pérez, Jennifer (Ed.): *Software Architecture*. Cham : Springer International Publishing, 2018. – ISBN 978–3–030–00761–4, p. 347–363

[TSD+16] Thorn, J. ; Schmitz, Malte ; Dell'Anna, Jasmin ; Mildner, Alexander ; Janss, Armin: Erweiterung des Lebenszyklus von Medizingeräten um offene Vernetzungsfähigkeit. In: *Whitepaper* (2016)

[TW11] Tanenbaum, Andrew S. ; Wetherall, D.: *Computer networks*. 5th ed., International ed. / Andrew S. Tanenbaum, David J. Wetherall. Boston, Mass. and London : Pearson Education, 2011. – ISBN 9780132553179

[TZCB08] Tsai, Wei-Tek ; Zhou, Xinyu ; Chen, Yinong ; Bai, Xiaoying: On Testing and Evaluating Service-Oriented Software. In: *Computer* 41 (2008), Nr. 8, p. 40–46. http://dx.doi.org/10.1109/MC.2008.304. – DOI 10.1109/MC.2008.304. – ISSN 0018–9162

[UGG16] Ungurean, Ioan ; Gaitan, Nicoleta C. ; Gaitan, Vasile G.: A Middleware Based Architecture for the Industrial Internet of Things. In: *KSII Transactions on Internet and Information Systems* 10 (2016), Nr. 8. http://dx.doi.org/10.3837/tiis.2016.07.001. – DOI 10.3837/tiis.2016.07.001

[UP16] Ulriksen, Gro-Hilde ; Pedersen, Rune: Core Archetypes The Means to Build Confidence Around the Power of Structured EPR Systems? In: Hettinga, Marike (Ed.): *eTELEMED 2016*. Wilmington, DE, USA : IARIA, 2016. – ISBN 9781612084701, p. 174–179

[US 08] US Department of Health and Human Services: The HIPAA Privacy Rule. In: *US Department of Health and Human Services* (05.07.2008). https://ww

w.hhs.gov/hipaa/for-professionals/privacy/index.html, Accessed 20.11.2022

[U.S19a] U.S. Food & Drug Administration: *Cybersecurity Vulnerabilities Affecting Medtronic Implantable Cardiac Devices, Programmers, and Home Monitors: FDA Safety Communication*. https://public4.pagefreezer.com/content/FDA/16-06-2022T13:39/https://www.fda.gov/medical-devices/safety-communications/cybersecurity-vulnerabilities-affecting-medtronic-implantable-cardiac-devices-programmers-and-home. Version: 2019, Accessed 21.12.2022

[U.S19b] U.S. Food & Drug Administration: FDA warns patients and health care providers about potential cybersecurity concerns with certain Medtronic insulin pumps. In: *FDA* (27.06.2019). https://www.fda.gov/news-events/press-announcements/fda-warns-patients-and-health-care-providers-about-potential-cybersecurity-concerns-certain, Accessed 21.12.2022

[U.S20] U.S. Food & Drug Administration: *Cybersecurity Vulnerabilities in Certain GE Healthcare Clinical Information Central Stations and Telemetry Servers: Safety Communication*. https://www.fda.gov/medical-devices/safety-communications/cybersecurity-vulnerabilities-certain-ge-healthcare-clinical-information-central-stations-and. Version: 2020, Accessed 21.12.2022

[van03] van Rees, Reinout: Clarity in the usage of the terms ontology, taxonomy and classification. In: *CIB REPORT* 284 (2003), Nr. 432, p. 1–8

[van04] van der Merwe, Rudolph: *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*, OGI School of Science & Engineering at Oregon Health & Science University, Dissertation, 2004

[Vec19] Vector Group: SOA Software Architecture: Bridging the Gap Between AUTOSAR Classic and Adaptive Systems. (2019), Nr. 11/2019

[Vec21] Vector Informatik GmbH: *Vector Medical Engineering*. https://medical.vector.com/articles/vector-strengthens-sdc-standard. Version: 04.2021, Accessed 23.01.2024

[Vec24] Vector Informatik GmbH: *PREEvision | Die E/E-Engineering-Lösung*. https://www.vector.com/de/de/produkte/produkte-a-z/software/preevision/. Version: May 2024, Accessed 05.05.2024

[VOG+20] Vetter, Andreas ; Obergfell, Philipp ; Guissouma, Houssem ; Grimm, Daniel ; Rumez, Marcel ; Sax, Eric: Development Processes in Automotive Service-oriented Architectures. In: *2020 9th Mediterranean Conference on Embedded Computing (MECO)*, IEEE, 62020. – ISBN 978–1–7281–6949–1, p. 1–7

[VS21] Vetter, Andreas ; Sax, Eric: Hierarchical Versioning to Increase Compatibility in Signal-Oriented Vehicle Networks. In: Selvaraj, Henry (Ed.) ; Chmaj, Grzegorz (Ed.) ; Zydek, Dawid (Ed.): *Proceedings of the 27th International Conference on*

Systems Engineering, ICSEng 2020. Cham : Springer International Publishing, 2021. – ISBN 978–3–030–65796–3, p. 435–444

[Wal17] Wallenfels, Matthias: Medizintechnik rüstet sich für den Kampf gegen Cyberkriminelle. In: *gynäkologie + geburtshilfe* 22 (2017), Nr. 4, p. 44. http://dx.doi.org/10.1007/s15013-017-1208-x. – DOI 10.1007/s15013–017–1208–x. – ISSN 1439–3557

[WB06] Welch, Greg ; Bishop, Gary: An Introduction to the Kalman Filter. In: *Proc. Siggraph Course* 8 (2006)

[WCR+13] Weerakkody, Ruwan A. ; Cheshire, Nicholas J. ; Riga, Celia ; Lear, Rachael ; Hamady, Mohammed S. ; Moorthy, Krishna ; Darzi, Ara W. ; Vincent, Charles ; Bicknell, Colin D.: Surgical technology and operating-room safety failures: a systematic review of quantitative studies. In: *BMJ quality & safety* 22 (2013), Nr. 9, p. 710–718. http://dx.doi.org/10.1136/bmjqs-2012-001778. – DOI 10.1136/bmjqs–2012–001778

[WDG+23] Weingardt, Markus S. ; Del Alcazar von Buchwald, Rodrigo ; Golde, Tim ; Gaiser, Immanuel ; Roskorsch, Michael: *Safety system for use in medical tables*. 2023. – WO2023088666A1

[Web19] Weber, Marc: *Untersuchungen zur Anomalieerkennung in automotive Steuergeräten durch verteilte Observer mit Fokus auf die Plausibilisierung von Kommunikationssignalen*. Karlsruhe, Karlsruhe Institute for Technology, Dissertation, 2019

[Wen11] Wendel, Jan: *Integrierte Navigationssysteme: Sensordatenfusion, GPS und Inertiale Navigation*. 2nd ed. München : De Gruyter, 2011. – ISBN 9783486705720

[Win16] Winton, Richard: Hollywood hospital pays $17,000 in bitcoin to hackers; FBI investigating. In: *Los Angeles Times* (18.2.2016). https://www.latimes.com/business/technology/la-me-ln-hollywood-hospital-bitcoin-20160217-story.html, Accessed 21.12.2022

[WJL+17] Wurm, Jacob ; Jin, Yier ; Liu, Yang ; Hu, Shiyan ; Heffner, Kenneth ; Rahman, Fahim ; Tehranipoor, Mark: Introduction to Cyber-Physical System Security: A Cross-Layer Perspective. In: *IEEE Transactions on Multi-Scale Computing Systems* 3 (2017), Nr. 3, p. 215–227. http://dx.doi.org/10.1109/TMSCS.2016.2569446. – DOI 10.1109/TMSCS.2016.2569446

[WKN20] Wee, Ian Jun Y. ; Kuo, Li-Jen ; Ngu, James Chi-Yong: A systematic review of the true benefit of robotic surgery: Ergonomics. In: *The international journal of medical robotics + computer assisted surgery : MRCAS* 16 (2020), Nr. 4, p. e2113. http://dx.doi.org/10.1002/rcs.2113. – DOI 10.1002/rcs.2113

[WKSZ18] Weber, Marc ; Klug, Simon ; Sax, Eric ; Zimmer, Bastian: Embedded Hybrid Anomaly Detection for Automotive CAN Communication. In: *9th European Congress on Embedded Real Time Software and Systems (ERTS 2018)*. Toulouse, France : HAL, 2018, p. 10

[WM96] Wolpert, David ; Macready, William: No Free Lunch Theorems for Search. (1996), March

[WSP16] Wagner, Marco ; Schildt, Sebastian ; Poehnl, Michael: Service-Oriented Communication for Controller Area Networks. In: *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, IEEE, 2016. – ISBN 978–1–5090–1701–0, p. 1–5

[WWH20] Wolsing, Konrad ; Wagner, Eric ; Henze, Martin: Facilitating Protocol-independent Industrial Intrusion Detection Systems. In: Ligatti, Jay (Ed.) ; Ou, Xinming (Ed.) ; Katz, Jonathan (Ed.) ; Vigna, Giovanni (Ed.): *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. New York, NY, USA : ACM, 2020. – ISBN 9781450370899, p. 2105–2107

[WWRS20] Weber, Philipp ; Weiss, Philipp ; Reinhardt, Dominik ; Steinhorst, Sebastian: Energy-Optimized Elastic Application Distribution for Automotive Systems in Hybrid Cloud Architectures. In: *2020 23rd Euromicro Conference on Digital System Design (DSD)*, IEEE, 2020. – ISBN 978–1–7281–9535–3, p. 455–462

[YL21] Yuan Liao, Fraunhofer I.: *Autonome Systeme – Roboter-Betriebssysteme: ROS2 bügelt Schwächen aus – Magazin des Fraunhofer-Instituts für Kognitive Systeme IKS*. `https://safe-intelligence.fraunhofer.de/artikel/autonome-systeme-ros2`. Version: 01.10.2021, Accessed 19.10.2021

[Yos15] Yoshida, Junko: *EETimes - CAN Bus Can Be Encrypted, Says Trillium*. `https://www.eetimes.com/can-bus-can-be-encrypted-says-trillium/?page_number=1`. Version: 2015, Accessed 18.05.2021

[ZCD+22] Zhu, Jihong ; Cherubini, Andrea ; Dune, Claire ; Navarro-Alarcon, David ; Alambeigi, Farshid ; Berenson, Dmitry ; Ficuciello, Fanny ; Harada, Kensuke ; Kober, Jens ; LI, Xiang ; Pan, Jia ; Yuan, Wenzhen ; Gienger, Michael: Challenges and Outlook in Robotic Manipulation of Deformable Objects. In: *IEEE Robotics & Automation Magazine* (2022), p. 2–12. `http://dx.doi.org/10.1109/MRA.2022.3147415`. – DOI 10.1109/MRA.2022.3147415. – ISSN 1070–9932

[ZLK+19] Zhang Jeff Jun ; Liu Kang ; Khalid Faiq ; Hanif Muhammad Abdullah ; Rehman Semeen ; Theocharides Theocharis ; Artussi Alessandro ; Shafique Muhammad ; Garg Siddharth: INVITED: Building Robust Machine Learning Systems: Current Progress, Research Challenges, and Opportunities. In: *2019 56th ACM/IEEE Design Automation Conference (DAC)*, 2019, p. 1–4

[ZS14] Zimmermann, Werner ; Schmidgall, Ralf: *Bussysteme in der Fahrzeugtechnik*. Wiesbaden : Springer Fachmedien Wiesbaden, 2014. `http://dx.doi.org/10.1007/978-3-658-02419-2`. `http://dx.doi.org/10.1007/978-3-658-02419-2`. – ISBN 978–3–658–02418–5

[ZZL+21] Zhu, Hailong ; Zhou, Wei ; Li, Zhiheng ; Li, Li ; Huang, Tao: Requirements-Driven Automotive Electrical/Electronic Architecture: A Survey and Prospective Trends. In: *IEEE Access* 9 (2021), p. 100096–100112. `http://dx.doi.org/10.1109/ACCESS.2021.3093077`. – DOI 10.1109/ACCESS.2021.3093077

List of Figures

List of Tables

Abbreviations

ABAC	Attribute-based Access Control
ADL	Architecture Description Language
AE	Autoencoder
AI	Artificial Intelligence
API	Application Programming Interface
ASIL	Automotive Safety Integrity Level
AUTOSAR	AUTomotive Open System ARchitecture
BMI	Body Mass Index
BSI	Federal Office for Information Security
CAN	Controller Area Network
CIS	Clinical Information System
CORBA	Common Object Request Broker Architecture
CPR	Cardiopulmonary Resuscitation
CPS	Cyber-Physical System
CRC	Cyclic Redundancy Check
CSV	Comma-Separated Values
CT	Computer Tomography
CoG	Center of Gravity
DDS	Data Distribution Service
DFD	Data Flow Diagram
DICOM-SR	DICOM Structured Reporting
DICOM	Digital Imaging and Communications in Medicine
DoF	Degrees of Freedom
DOOP	Dienst-orientierte OP-Integration
DoS	Denial of Service
ECU	Electronic Control Unit
E/E architecture	Electric/Electronic architecture
EHR	Electronic Health Record
EKF	Extended Kalman Filter
EMR	Electronic Medical Record
EPR	Electronic Patient Record
EUC	Equipment Under Control
EVITA	E-safety vehicle intrusion protected applications
FDA	Food and Drug Administration
FEM	Finite Element Method
FHIR	Fast Healthcare Interoperability Resources
FMEA	Failure Mode and Effects Analysis
FNR	False Negative Rate
FPR	False Positive Rate
FSM	Finite Segment Method
HAL	Hardware Abstraction Layer
HEAVENS	HEAling Vulnerabilities to ENhance Software Security and Safety
HF	High Frequency

HIDS	Host Intrusion Detection System
HIPAA	Health Insurance Portability and Accountability Act
HL7	Health Level Seven
HMI	Human-Machine-Interface
HOR	Hybrid Operating Room
IAM	Identity and Access Management
IC	Integrated Circuit
ICE	Integrated Clinical Environment
ICU	Intensive Care Unit
IDS	Intrusion Detection System
IF	Isolation Forest
IMDRF	International Medical Device Regulators Forum
IMM	Interacting Multiple Model
IMU	Inertial Measurement Unit
IP	Internet Protocol
IQR	Inter Quantile Range
ISAC	Information Sharing and Analysis Center
ISO	International Standards Organization
IT	Information Technology
IoMT	Internet of Medical Things
IoT	Internet of Things
KF	Kalman Filter
KIT	Karlsruhe Institute of Technology
LODA	Lightweight On-line Detector of Anomalies
LReLU	Leaky Rectified Linear Unit
LSTM	Long Short-Term Memory
LTI	Linear Time Invariant
MAC	Message Authentication Code
MAE	Mean Absolute Error
MCPS	Medical Cyber-Physical Systems
MDD	Medical Device Directive
MDIB	Medical Device Information Base
MDPWS	Medical Device Profile for Web Services
MDPnP	Medical Device Plug and Play
MDR	Medical Device Regulation
MDS	Medical Device System
ME Equipment	Medical Electrical Equipment
ME System	Medical Electrical System
MQTT	Message Queue Telemetry Transport
MRDS	Microsoft Robotics Developer Studio
NIDS	Network Intrusion Detection System
NIST	National Institute of Standards and Technology
NLP	Natural Language Processing
NN	Neural Network
OBD	On-Board Diagnostics
OCTAVE	Operationally Critical Threat, Asset, and Vulnerability Evaluation
OEM	Original Equipment Manufacturer
OMG	Object Management Group
OPC UA	Open Platform Communications Unified Architecture

OPF	Open Platform Foundation
OR table	Operating Room Table
ORiN	Open Robot/Resource interface for the Network
OR	Operating Room
OSEK	Offene Systeme und deren Schnittstellen für die Elektronik im Kraftfahrzeug
OSI	Open Systems Interconnection
OS	Operating System
OTA	Over The Air
PACS	Picture Archiving and Communication System
PASTA	Process for Attack Simulation and Threat Analysis
PDMS	Patient Data Management System
PESS	Programmable Electrical Subsystem
PHD	Personal Health Device
PHI	Protected Health Information
PLC	Programmable Logic Controller
POC	Point of Care
POSIX	Portable Operating System Interface
QoS	Quality of Service
R/R	Request/Response
REST	REpresentational State Transfer
RIM	Reference Information Model
RMSE	Root Mean Square Error
RNN	Recurrent Neural Networks
ROS	Robot Operating System
RTE	Real-Time Ethernet
RTOS	Real-Time Operating System
ReLU	Rectified Linear Unit
SB/SO-GW	Signal-Based/Service-Oriented Gateway
SCADA	Supervisory Control and Data Acquisition
SCOT	Smart Cyber Operating Theater®
SCO	Service and Control Object
SDC	Service-oriented Device Connectivity
SIEM	Security Information and Event Management
SIL	Safety Integrity Level
SOA	Service Oriented Architecture
SODA	Service-Oriented Device Architecture
SOMDA	Service-Oriented Medical Device Architecture
SOME/IP-SD	SOME/IP Service Discovery
SOME/IP	Scalable service-oriented Middleware over IP
SPI	Serial Peripheral Interface
SRTB	Surgical Real-Time Bus
SSC	Software Safety Classification
STRIDE	Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege
SiL	Software-in-the-Loop
SoS	System of Systems
TARA	Threat Analysis and Risk Assessment
TCP	Transfer Control Protocol

TLS	Transport Layer Security
TMR	Triple Modular Redundancy
TSN	Time-Sensitive Networking
Trend	Trendelenburg
UDP	User Datagram Protocol
UI	User Interface
UKF	Unscented Kalman Filter
URDF	Unified Robot Description Format
USB	Universal Serial Bus
VCID	virtual CAN network ID
VLAN	Virtual Local Area Network
VMD	Virtual Medical Device
VPN	Virtual Private Network
W3C	World Wide Web Consortium
WADO	Web Access to DICOM Persistent Objects
XML	Extensible Markup Language
YAML	YAML Ain't Markup Language
CAN-FD	CAN with Flexible Data Rate
eSOC	embedded Service-Oriented Communication
IP	Internet Protocol
LIN	Local Interconnect Network
MES	Manufacturing Execution System
PEMS	Programmable Electrical Medical System
TCP	Transfer Control Protocol
UDP	User Datagram Protocol
vMR	Virtual Medical Record
1DCNN	1D Convolutional Neural Network
DBN	Dynamic Bayesian Network

Andreas Puder

ANOMALY **DETECTION** FOR **INTEROPERABLE** AND **MODULAR** **OPERATING ROOM** TABLES

Technologies that facilitate connectivity, interoperability and automation have the potential to enhance the quality and efficiency of patient care. However, the integration of these technologies into medical devices is progressing at a gradual pace. This work therefore uses the example of operating room tables to investigate the necessary adaptations to the software and E/E architectures of medical devices in the operating room to achieve these goals without compromising safety and security.

The author proposes a hybrid architecture with signal-based and service-oriented communication that fulfills the flexibility requirements for modular and interoperable systems, while also considering legacy modules. Particular attention is paid to the detection of anomalies, for example in the movement of the operating room table, by monitoring its positions. Therefore, a novel approach that uses hybrid plausibility checks is presented, combining data- and model-based algorithms for anomaly detection.

ISBN 978-3-7315-1419-0

Gedruckt auf FSC-zertifiziertem Papier