Supporting Cybersecurity Risk Management for Medical Devices via the SECUMAN Ontology and Shapes

Martin DILLER [a,1], Anne ESSLINGER [b], Piotr GORCZYCA [a], Evi HARTIG [b], Lia KACHOLDT [c], and Hannes STRASS [a]

[a] *Faculty of Computer Science, TU Dresden, Germany*
[b] *Else Kröner Fresenius Center for Digital Health, Dresden, Germany*
[c] *secunet Security Networks AG*

Abstract. We propose the SECUMAN ontology and shapes for representing and analysing cybersecurity risk-management documentation for medical devices. Cybersecurity risks are increasingly relevant for connected medical devices and may have direct consequences for patient safety. Current risk-management files are often maintained as semi-structured natural language text, which makes consistency checking, certification review, and reuse difficult. SECUMAN provides a formal OWL-based vocabulary for modelling security-risk context, assessment, control measures, and residual-risk evaluation, and uses SHACL constraints to check structural completeness and conformity with the intended documentation model. The ontology is aligned with VDE Spec 90025 and the related RISKMAN ontology and shapes, while extending their safety-oriented approach to concepts relevant to cybersecurity risk documentation such as threat scenarios, protection goals, attacker profiles, exposure levels, assets, and secure design arguments. SECUMAN is intended to support automated first-pass validation, traceability, and integration of cybersecurity and safety risk-management documentation.

Keywords. Risk management, Cybersecurity, OWL EL, SHACL, Medical devices

1. Introduction

Medical devices are safety-critical systems whose failures may harm patients, users, or the environment. Manufacturers must therefore provide documented justification that device-related risks have been systematically identified, evaluated, and controlled, as required in the EU by the Medical Device Regulation [1]. In practice, however, risk-management documentation is still often exchanged as semi-structured natural language. This complicates certification and maintenance: notified bodies must inspect large volumes of text, manufacturers must keep risk files consistent across device versions, and reusable parts of risk analyses and risk controls remain difficult to identify and transfer.

To address these problems for safety risk management, the RISKMAN ontology and shapes were proposed in [2]. Building on ISO 14971 and VDE SPEC 90025 [3,

[1] Corresponding Author: Martin Diller, martin.diller@tu-dresden.de

4], RISKMAN encodes central concepts and structures used in safety-oriented risk-management documentation for medical devices, while its SHACL shapes express structural and semantic constraints relevant to such documentation. The approach is not intended to determine whether a concrete control measure is technically adequate; rather, it supports automated first-pass validation, improved navigation, and more systematic reuse of risk-management documentation.

For connected medical devices, including wearable monitors, implantable systems, and networked clinical equipment, safety-oriented documentation alone is insufficient. Such devices are now an integral part of modern healthcare: they enable continuous patient monitoring, real-time physiological data acquisition, and remote management, thereby improving diagnosis, treatment, and healthcare efficiency [5]. However, increasing connectivity also introduces cybersecurity risks that may affect patient safety [6]. Cybersecurity risks in healthcare are therefore not merely theoretical. In Europe, ransomware attacks on hospitals have led to cancelled operations and reduced treatment capacity, as observed in incidents in Spain and the United Kingdom [7,8], while attacks on national health systems have disrupted access to patient data and critical services [9]. In the United States, healthcare data breaches have remained at a sustained high level, with more than 700 large breaches reported annually and over 275 million patient records exposed in 2024 alone [10,11]. At the same time, the cybersecurity threat landscape continues to intensify, with ransomware accounting for more than half of attacks on the health sector [12,13]. These developments show that cybersecurity has become a critical aspect of healthcare systems, with direct implications for patient safety.

Regulatory and guidance documents explicitly reflect this connection. The EU MDR requires software-based systems to be developed in accordance with state-of-the-art principles, including information security principles (MDR Annex I, Sections 3 and 17.2). MDCG 2019-16 likewise treats safety and security risks as interdependent and requires cybersecurity risks and risk controls with potential safety impact to be considered within safety risk-management processes, and vice versa [1,14]. Similar expectations are reflected in international guidance and standards, including IMDRF principles on medical device cybersecurity, IEC 81001-5-1, and AAMI TIR57, which emphasize the need to manage cybersecurity risks in conjunction with patient-safety considerations across the device lifecycle [15,16,17].

A separate ontology for cybersecurity risk management is needed because security risks are conceptualized differently from safety risks. Safety risk management typically starts from device-related hazards and analyzes how they may lead to hazardous situations and harms. Cybersecurity risk management must instead account for threat scenarios shaped by the operating environment, product security properties, attack types, protection goals, vulnerabilities, attacker capabilities, affected assets, and impacts. These concepts are often handled by different expert groups and documented using different terminology, evaluation methods, and data structures [18]. A security ontology must therefore preserve cybersecurity-specific concepts while remaining compatible with safety-oriented documentation, so that security–safety dependencies can be made explicit where needed.

In this work, we provide such an ontology, based on a model for risk-management documentation inspired by VDE SPEC 90025 and RISKMAN, but specialised to cybersecurity using relevant standards and guidance such as AAMI TIR57, AAMI SW96, NIST SP 800-30, and IEC 81001-5-1 [17,19,20,15]. Specifically, our contributions are:

(i) an ontology for representing central concepts and structural elements of cybersecurity risk-management documentation for medical devices; (ii) SHACL shapes expressing constraints on such documentation; and (iii) a RISKMAN-compatible modelling approach that supports traceability from threat context through risk assessment to risk control and residual-risk evaluation. SECUMAN thereby complements RISKMAN by covering the security-risk domain and provides a basis for future integration between cybersecurity and safety risk-management documentation.

2. Related Work

Risk-management ontologies for medical devices. Several ontologies have been proposed for medical-device regulation, risk-related concepts, or adjacent compliance tasks, but only few target risk-management documentation itself – see also the discussion in [2]. Uciteli et al. [21] defined a Risk Identification Ontology embedded in the General Formal Ontology (GFO) [22,23], but their focus was on identifying risks in time periods surrounding surgical procedures. Kim et al. [24] integrated concepts from IEC 60601-1, IEC 62304, and ISO 14971 in an ontology for medical-software development processes; however, their model is primarily intended to support process integration and standards compliance, rather than automated reasoning over risk-management files. Schütz et al. [25] developed an ontology for medical devices in Germany with a broader focus on manufacturers, operators, and legal procedures, yet with the aim of general semantic interoperability, and based on a legal framework that has since been superseded by the EU MDR.

More recent work has explored semantic technologies for neighboring medical-device regulatory tasks: Chattoraj et al. [26] proposed an OWL/RDF knowledge-graph architecture for representing and querying FDA medical-device regulatory rules, while Zhu et al. [27] introduced a methodology for constructing a medical-device risk knowledge graph that combines standards, adverse-event data, and large language models. These works illustrate a continued interest in machine-processable representations for medical-device regulation and risk-related information, but they do not provide an ontology that supports the generation and validation of medical-device risk-management documentation itself. As already mentioned, this work builds on the RISKMAN ontology [2], which formalizes safety-oriented risk-management documentation for medical devices using OWL and SHACL. SECUMAN adopts this documentation-centered approach and is intended to integrate with RISKMAN, for example by mirroring its separation into "context", "assessment", and "control" aspects. However, SECUMAN targets the security domain: it models concepts required for medical-device security-risk management, including security-specific mitigation strategies represented as secure-design arguments, in contrast to the safe-design arguments used in RISKMAN.

Cybersecurity and security-risk ontologies in the medical domain. Ontology-based security modelling covers a wide range of concerns, including security taxonomies, vulnerability and attack representation, threat intelligence, operational technology and industrial Internet of Things security, semantic log analysis, and ontology-supported large-language-model applications, as summarized in surveys on these topics, e.g. [28,29,30, 31,32]. As to work closely related to our setting, Sills et al. [33] constructed a cybersecurity knowledge graph for medical-device cyber threat intelligence by integrating man-

ufacturer bulletins, CISA Industrial Control Systems Cyber Emergency Response Team alerts, Wikidata, and FDA AccessGUDID data; their focus, however, is on cyber-threat-intelligence aggregation and embedding-based retrieval rather than on a documentation ontology for security-risk management. Hannou et al. [34] introduced SafecareOnto, a cyber-physical security ontology for healthcare infrastructures that models hospital assets, dependencies, impacts, and protections in order to reason about incident propagation, whereas SECUMAN targets medical-device security-risk management files and their validation. Dart and Ahmed [35] propose CYBER-AIDD, a Unified Modeling Language-based ontology and governance framework for cybersecurity resilience in large healthcare providers, but this is a high-level organizational governance model rather than a formal OWL and SHACL framework for machine-checking manufacturer risk-management documentation. Liu et al. [36] presented an ontology-based framework for classifying forensic evidence from Internet of Medical Things devices; this work is security-adjacent and medical-domain-specific, but it concerns post-incident forensic evidence prioritization rather than prospective security-risk analysis and control documentation.

Work on Internet of Medical Things vulnerability modelling and ontology-assisted automation is closer to our setting, but still differs in its target artefacts. Bughio et al. [37] developed an Internet of Medical Things cybersecurity ontology for remote patient monitoring that represents devices, vulnerabilities, exploits, attacks, services, vendors, and affected assets, and in related work proposed a knowledge-graph-based framework for Internet of Medical Things vulnerability detection [38]; but they are primarily oriented toward infrastructure, vulnerabilities, and threat intelligence rather than toward documentation and validation. Ghosh et al. [39] presented CVE-LLM, an ontology-assisted large language model system for vulnerability evaluation in post-market medical-device cybersecurity, using existing cybersecurity knowledge such as CWE and CAPEC via the SEPSES knowledge graph to support artefacts such as affected-status classifications, justifications, comments, and environmental CVSS vectors; this complements SECUMAN by showing the usefulness of ontology-enriched automation, but it does not define a formal ontology and SHACL constraint layer for security-risk management files themselves. Finally, Chander et al. [40] proposed an Intelligent Cybersecurity Ontology Framework for Internet of Medical Things-enabled remote patient monitoring that combines ontology-based modelling of devices, dependencies, vulnerabilities, and threats with graph learning, temporal updates, risk scoring, and adaptive mitigation, thereby addressing operational threat detection and response rather than regulatory or engineering documentation.

Overall, the closest related works are those on medical-device cyber threat intelligence, Internet of Medical Things vulnerability modelling, and ontology-assisted vulnerability evaluation [33,37,38,39]; these approaches are complementary to SECUMAN, since they structure vulnerability, threat, asset, or operational-security knowledge, while SECUMAN focuses on a lightweight OWL ontology and SHACL shapes for representing, reasoning over, and validating medical-device security-risk management documentation.

3. Background

The SECUMAN ontology follows the RISKMAN ontology in being based on the recent VDE Spec 90025 [4], which proposes a structured format for digitalizing risk manage-

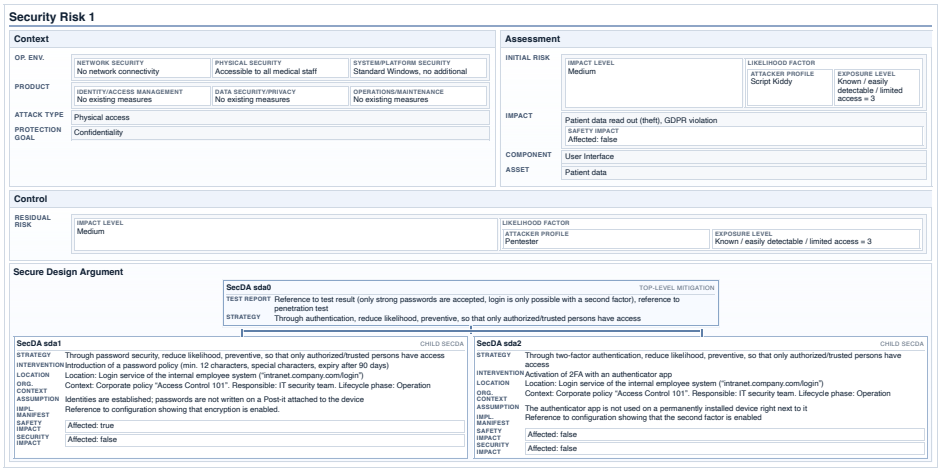

Figure 1. HTML rendering of a security risk.

ment files, as well as a machine-readable exchange format using HTML with RDFa [41], which annotates selected HTML tags with Resource Description Framework (RDF) triples. However, whereas VDE Spec 90025 and RISKMAN focus on safety aspects of medical devices, primarily following ISO 14971 [3], SECUMAN formalises a model for cybersecurity risk management documentation drawing on cybersecurity-relevant standards and guidance, such as AAMI TIR57, AAMI SW96, NIST SP 800-30, and IEC 81001-5-1 [17,19,20,15]. At the same time, SECUMAN follows the same philosophy as VDE Spec 90025 and RISKMAN with respect to the digitalisation and validation of risk management files, and is intended to support integration with safety-focused risk management documentation.

In contrast to vulnerability-centric approaches, which assess isolated technical weaknesses, the model formalised by SECUMAN is built around contextualized threat scenarios, where risk emerges from the interaction of the operating environment, device-specific properties, attack types, and attacker capabilities. The model is structured into three sections: Context, Assessment, and Control, mirroring the conceptual structure of safety risk management as formalised in VDE Spec 90025. This separation enables consistent reuse of context attributes across multiple risk entries, thereby reducing the workload involved in creating a security risk management file, and supports traceability from threat description to risk control. The model is designed to support structured documentation of security risks and to enable alignment and systematic integration with safety risk management.

Figure 1 shows the rendering of an HTML file containing a cybersecurity risk structured according to the model formalised by SECUMAN. The context of the risk describes the conditions under which a threat can occur. It is defined by the combination of information pertaining to the operating environment, product, attack type, and protection goal. These elements represent reusable attributes that are not specific to a single risk, but can be applied across multiple risk scenarios and even across multiple devices employed in similar contexts.

Specifically, the operating environment captures external conditions relevant to cybersecurity. It includes physical security, network security, and system and platform security or hardening. The product describes security-relevant properties of the device itself, including identity and access management, data security and privacy, and operations and maintenance. The protection goal is one of confidentiality, integrity or availability.

The assessment aspects describe how the threat affects the system, its operational environment, and relevant stakeholders. In particular, the assessment refers to the context and extends it with the component through which the attack occurs, the affected asset, the impact, and the initial security risk level. The component represents the high-level system element that serves as the entry point of the attack, while the asset represents the physical or digital entity that has value to an individual, an organization, or a government and that is affected by the attack. The impact may include different types of consequences, such as harm to people, financial loss, legal implications, or operational disruption. It also explicitly records whether an impact on safety can occur. The initial security risk level, i.e., the security risk level before any mitigation has been proposed, is composed of an exposure level and an attacker profile, which together form the likelihood factor, as well as an impact level.

The control section represents the risk after the application of control measures. It refers to the assessment aspects and includes a secure design argument and the residual security risk level. The latter has the same structure as the initial security risk level, but represents the risk level after mitigation measures have been applied.

Cybersecurity risk control in the proposed model is represented through secure design arguments (SecDAs). SecDAs are a variant of the safety design arguments proposed in VDE Spec 90025 and formalised in RISKMAN – which in turn are a simplified form of assurance cases as defined in standards such as ISO/IEC/IEEE 15026 and ISO/TS 81001-2-1 [42,43] – specialised for mitigation strategies addressing security risks.

SecDAs are structured as trees, with the top-level SecDA representing the claim that a given cybersecurity risk is adequately controlled. Sub-arguments, or sub-SecDAs, contribute to this claim by refining the control strategy into more specific measures. The tree structure enforces a clear separation of abstraction levels: the top-level SecDA defines the overall strategy and includes a test report for effectiveness verification; intermediate SecDAs refine the strategy without implementation detail; and leaf SecDAs define concrete control measures. The strategy defines the overarching approach to risk mitigation, including the control type, i.e., preventive, detective, or corrective, the rationale for the control, and its intended effect on risk parameters, e.g., the likelihood or impact factor. In addition to the strategy, each leaf SecDA contains attributes describing the intervention method, the location of intervention, the organisational context of intervention, assumptions, and an assessment of the impact on safety and security.

Specifically, the intervention method describes the mechanism by which the control is applied. It is the most concrete description of the control and specifies what must be implemented. It may include technical, operational, or administrative measures. The location of intervention specifies where the control is applied within the system, including device internals, connected infrastructure, or, where relevant, the deployment environment. The organisational context of intervention describes the operational setting of the control, including the device lifecycle phase, responsible stakeholders, and relevant policies or practices. The assumptions capture conditions that must hold for the control to be effective, such as the availability of infrastructure or expected user behaviour. Finally,

the assessment of the impact on safety and security indicates whether the control measure may itself introduce a safety impact or an additional security impact that requires mitigation.

SecDAs inherit the subcategorisation of safe design arguments from VDE Spec 90025 and RISKMAN into implementation-specific arguments and assurance arguments. Implementation-specific SecDAs include an implementation manifest, which provides evidence of how the control measure has been realised within the medical device; in particular, leaf SecDAs must always include an implementation manifest. Assurance SecDAs, on the other hand, additionally reference established state-of-the-art practices, such as standards, regulatory guidance, or accepted security frameworks. These references support the justification and traceability of control strategies. In contrast to safety, where standards often define the state of the art more comprehensively, cybersecurity assurance may also rely on evolving best practices and accepted security baselines.

4. Design & Overview

4.1. Design

The SECUMAN ontology was developed based on requirements and conceptual design decisions defined by a consortium of experts in medical devices, regulatory affairs, security risk management, ontology engineering, and software development. We followed the Linked Open Terms (LOT) methodology [44], which builds on the NeOn methodology [45]. The development process comprised three steps: (1) Requirement specification: We studied relevant norms and standards and examined real medical device security risk management files to establish a shared understanding of the required terms. This was done through regular consortium meetings, workshops, and meetings with manufacturers and security risk experts. The result was a conceptual model of security risk documentation that formed the basis of the SECUMAN ontology. (2) Implementation: We encoded the ontology and shapes in OWL and SHACL. During development, we primarily used the HermiT reasoner [46] and the OntOlogy Pitfalls Scanner (OOPS!) [47] to support validity and consistency checks. (3) Publication and maintenance: The SECUMAN ontology and shapes[2] as well as the associated validation pipeline[3] are maintained in separate GitHub repositories for version control, issue tracking, and open online access.

4.2. Formal Background

The SECUMAN ontology is encoded in the $\mathscr{EL}$ profile of OWL [48], while the shapes are specified in SHACL [49]. For ease of presentation, we describe the ontology here using the description logic $\mathscr{EL}^{++}$ [50,51] that underlies the $\mathscr{EL}$ profile of OWL. For shapes, we use the abstract SHACL syntax introduced by Corman et al. [52,53]. Before giving an overview of SECUMAN, we therefore briefly recall the syntax, semantics, and aspects of reasoning relevant to our approach of $\mathscr{EL}^{++}$ as well as SHACL in the sense of Corman et al.

[2] `https://github.com/cl-tud/secuman`

[3] `https://github.com/cl-tud/semeco-q2-validation-pipeline`

The Description Logic $\mathcal{EL}^{++}$ $\mathcal{EL}^{++}$'s concept constructors and their semantics are recalled in the upper part of Table 1; the middle (lower) part shows the constructs allowed in a TBox (ABox). As usual for description logics, the semantics of $\mathcal{EL}^{++}$ is defined

Table 1. Syntax and semantics of concept, TBox, and ABox expressions of $\mathcal{EL}^{++}$.

Name	Syntax	Semantics
individual name	$\mathsf{a} \in \mathsf{N_I}$	$\mathsf{a}^{\mathscr{I}} \in \Delta^{\mathscr{I}}$
concept name	$\mathsf{A} \in \mathsf{N_C}$	$\mathsf{A}^{\mathscr{I}} \subseteq \Delta^{\mathscr{I}}$
role name	$\mathsf{R} \in \mathsf{N_R}$	$\mathsf{R}^{\mathscr{I}} \subseteq \Delta^{\mathscr{I}} \times \Delta^{\mathscr{I}}$
top	$\top$	$\Delta^{\mathscr{I}}$
bottom	$\bot$	$\emptyset$
nominal	$\{\mathsf{a}\}$	$\{\mathsf{a}^{\mathscr{I}}\}$
conjunction	$C \sqcap D$	$C^{\mathscr{I}} \cap D^{\mathscr{I}}$
existential restriction	$\exists \mathsf{R}.C$	$\{x \in \Delta^{\mathscr{I}} \mid \exists y \in \Delta^{\mathscr{I}} \colon (x,y) \in \mathsf{R}^{\mathscr{I}} \;\&\; y \in C^{\mathscr{I}}\}$
domain restriction	$dom(\mathsf{R}) \sqsubseteq \mathsf{A}$	$\mathsf{R}^{\mathscr{I}} \subseteq \mathsf{A}^{\mathscr{I}} \times \Delta^{\mathscr{I}}$
range restriction	$ran(\mathsf{R}) \sqsubseteq \mathsf{A}$	$\mathsf{R}^{\mathscr{I}} \subseteq \Delta^{\mathscr{I}} \times \mathsf{A}^{\mathscr{I}}$
general concept inclusion	$C \sqsubseteq D$	$C^{\mathscr{I}} \subseteq D^{\mathscr{I}}$
role inclusion axiom	$\mathsf{R}_1 \circ \cdots \circ \mathsf{R}_k \sqsubseteq \mathsf{R}$	${\mathsf{R}_1}^{\mathscr{I}} \circ \cdots \circ {\mathsf{R}_k}^{\mathscr{I}} \subseteq \mathsf{R}^{\mathscr{I}}$
concept assertion	$\mathsf{A}(\mathsf{a})$	$\mathsf{a}^{\mathscr{I}} \in \mathsf{A}^{\mathscr{I}}$
role assertion	$\mathsf{R}(\mathsf{a},\mathsf{b})$	$(\mathsf{a}^{\mathscr{I}}, \mathsf{b}^{\mathscr{I}}) \in \mathsf{R}^{\mathscr{I}}$

via *interpretations* $\mathscr{I} = (\Delta^{\mathscr{I}}, \cdot^{\mathscr{I}})$ with a non-empty *domain* $\Delta^{\mathscr{I}}$ and an *interpretation function* $\cdot^{\mathscr{I}}$. An interpretation $\mathscr{I}$ is a *model* of an ABox $\mathscr{A}$ (TBox $\mathscr{T}$) if it satisfies all elements of $\mathscr{A}$ ($\mathscr{T}$) as per Table 1. An assertion α is *entailed* by $\mathscr{T} \cup \mathscr{A}$, written $\mathscr{T} \cup \mathscr{A} \models \alpha$, if every model of $\mathscr{T} \cup \mathscr{A}$ is a model of α.

SHACL To define SHACL, we conveniently re-use description logic vocabulary, viz., pairwise disjoint sets $\mathsf{N_C}$ of *classes*, $\mathsf{N_R}$ of *properties*, and $\mathsf{N_I}$ of *individuals*. A finite set $\mathscr{A}$ of assertions (an ABox) can then be seen as representing a labelled *graph*, with individuals acting as nodes, classes labelling nodes, and properties labelling edges. The syntax of *shape expressions* ϕ and *path expressions* E is shown in Table 2 (top). For the semantics, a given graph (ABox) $\mathscr{A}$ with nodes (individuals) $\mathsf{N_I}(\mathscr{A})$ defines an evaluation function $[\![\cdot]\!]^{\mathscr{A}}$ that assigns to each path expression E a binary relation $[\![E]\!]^{\mathscr{A}} \subseteq \mathsf{N_I}(\mathscr{A}) \times \mathsf{N_I}(\mathscr{A})$, and to each shape expression ϕ a set $[\![\phi]\!]^{\mathscr{A}} \subseteq \mathsf{N_I}(\mathscr{A})$ via induction as shown in Table 2 (bottom).

A *shape constraint* is an expression of the form $\mathsf{A} \leftarrow \phi$, with $\mathsf{A} \in \mathsf{N_C}$ and ϕ a shape expression. A *shape schema* is a pair $(\mathscr{C}, \mathscr{B})$ where $\mathscr{C}$ is a set of shape constraints and $\mathscr{B}$ is a set of *target* concept assertions. Intuitively, a target $\mathsf{A}(\mathsf{a})$ expresses the requirement that a be labelled by A. Formally, an ABox $\mathscr{A}$ is a *model* for a set $\mathscr{C}$ of constraints if $[\![\phi]\!]^{\mathscr{A}} \subseteq [\![\mathsf{A}]\!]^{\mathscr{A}}$ for all $\mathsf{A} \leftarrow \phi \in \mathscr{C}$. An ABox $\mathscr{A}$ is *validated* against a schema $(\mathscr{C}, \mathscr{B})$ if there exists a set $\mathscr{B}'$ of concept assertions such that (1) $\mathscr{B} \subseteq \mathscr{B}'$, (2) $\mathsf{N_I}(\mathscr{B}') \subseteq \mathsf{N_I}(\mathscr{A})$, and (3) $\mathscr{A} \cup \mathscr{B}'$ is a model for $\mathscr{C}$. For brevity, in the following we also use the syntactic abbreviations $\leq_n E.\phi$ for $\neg(\geq_{n+1} E.\phi)$ and $=_1 E.\phi$ for $\geq_1 E.\phi \wedge \leq_1 E.\phi$.

Reasoning For reasoning with respect to the SECUMAN ontology and shapes, we adopt a materialisation-based approach. First, all relevant logical consequences of a given ABox together with the $\mathcal{EL}^{++}$ ontology are computed; more precisely, we derive all entailed concept and role assertions involving concept, role, and individual names occur-

Table 2. Syntax and semantics of path and shape expressions.

The syntax of path expressions E and shape expressions ϕ is given by the grammars

$$E ::= \mathsf{R} \mid \mathsf{R}^- \mid E \cup E \mid E \cdot E \mid E^* \text{ and } \phi ::= \top \mid \mathsf{A} \mid \mathsf{a} \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid \geq_n E.\phi \mid \forall E.\phi \mid E = E$$

where $n \in \mathbb{N}^+$, $\mathsf{A} \in \mathsf{N_C}$, $\mathsf{a} \in \mathsf{N_I}$, and $\mathsf{R} \in \mathsf{N_R}$ with R^- indicating the inverse of R.

$$[\![\mathsf{R}]\!]^{\mathscr{A}} = \{(a,b) \mid \mathsf{R}(a,b) \in \mathscr{A}\} \qquad [\![E_1 \cdot E_2]\!]^{\mathscr{A}} = [\![E_1]\!]^{\mathscr{A}} \circ [\![E_2]\!]^{\mathscr{A}}$$

$$[\![\mathsf{R}^-]\!]^{\mathscr{A}} = \{(b,a) \mid \mathsf{R}(a,b) \in \mathscr{A}\} \qquad [\![E_1 \cup E_2]\!]^{\mathscr{A}} = [\![E_1]\!]^{\mathscr{A}} \cup [\![E_2]\!]^{\mathscr{A}} \qquad [\![E^*]\!]^{\mathscr{A}} = \left([\![E]\!]^{\mathscr{A}}\right)^*$$

$$[\![\top]\!]^{\mathscr{A}} = \mathsf{N_I}(\mathscr{A}) \qquad [\![\mathsf{A}]\!]^{\mathscr{A}} = \{a \mid \mathsf{A}(a) \in \mathscr{A}\} \qquad [\![\mathsf{a}]\!]^{\mathscr{A}} = \{\mathsf{a}\}$$

$$[\![\phi_1 \wedge \phi_2]\!]^{\mathscr{A}} = [\![\phi_1]\!]^{\mathscr{A}} \cap [\![\phi_2]\!]^{\mathscr{A}} \qquad [\![\neg\phi]\!]^{\mathscr{A}} = \mathsf{N_I}(\mathscr{A}) \setminus [\![\phi]\!]^{\mathscr{A}}$$

$$[\![\forall E.\phi]\!]^{\mathscr{A}} = \left\{a \;\middle|\; \forall b : (a,b) \in [\![E]\!]^{\mathscr{A}} \text{ implies } b \in [\![\phi]\!]^{\mathscr{A}}\right\}$$

$$[\![\geq_n E.\phi]\!]^{\mathscr{A}} = \left\{a \;\middle|\; \left|\left\{(a,b) \in [\![E]\!]^{\mathscr{A}} \text{ and } b \in [\![\phi]\!]^{\mathscr{A}}\right\}\right| \geq n\right\}$$

$$[\![E_1 = E_2]\!]^{\mathscr{A}} = \left\{a \;\middle|\; \forall b : (a,b) \in [\![E_1]\!]^{\mathscr{A}} \text{ iff } (a,b) \in [\![E_2]\!]^{\mathscr{A}}\right\}$$

ring in the input ABox. Second, the resulting completed ABox is validated against the SHACL constraints.

The feasibility of this approach in our setting is ensured by three observations. First, the role inclusion axioms of SECUMAN satisfy the syntactic restriction imposed by Baader et al. [51, Section 3], and hence entailment of assertions can be decided in polynomial time. Second, SECUMAN uses existential quantification only on the left-hand side of general concept inclusion axioms. As a consequence, materialisation does not introduce fresh individuals, and thus remains finite. Finally, under the formal semantics of Corman et al., SHACL validation is NP-complete in general, whereas certain positive fragments are PTIME-complete [52].

4.3. The SECUMAN *Ontology & Shapes*

Common Aspects Figure 3 presents the GCIs for classes shared across the assessment and control aspects of security risk management documentation. In particular, the central class SecurityRisk is characterized compositionally: every individual with an attacker profile and an exposure level is a LikelihoodFactor, every individual with an impact level and a likelihood factor is a SecurityRiskLevel, and every individual with a security risk level is a SecurityRisk. Finally, there is the GCI that introduces the auxiliary class ImpactAssessment, which captures an assessment of whether a security risk or a control measure may induce further risks, in particular in the safety or security dimension. Concretely, it states that every individual with an isAffected value is an ImpactAssessment.

$$\exists \mathsf{isAffected}.\top \sqsubseteq \mathsf{ImpactAssessment} \tag{1}$$

$$\exists \mathsf{hasAttackerProfile}.\top \sqcap \exists \mathsf{hasExposureLevel}.\top \sqsubseteq \mathsf{LikelihoodFactor} \tag{2}$$

$$\exists \mathsf{hasSecurityRiskLevel}.\top \sqsubseteq \mathsf{SecurityRisk} \tag{3}$$

$$\exists \mathsf{hasImpactLevel}.\top \sqcap \exists \mathsf{hasLikelihoodFactor}.\top \sqsubseteq \mathsf{SecurityRiskLevel} \tag{4}$$

Figure 3. General concept inclusions for common aspects.

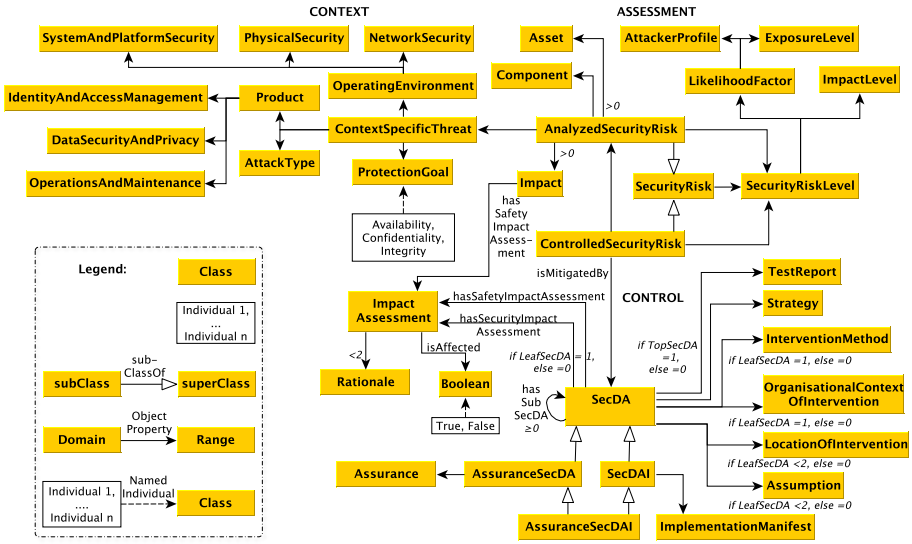

Figure 2. Schema diagram of the SECUMAN classes and properties divided into context, assessment, and control aspects. Any arrow representing a property without an explicit name means that the property has the name hasX where X is the class in the range, e.g. hasAsset for the property with domain AnalyzedSecurityRisk and range Asset. Property arrows are also labelled with information pertaining to the Shape constraints, e.g. an AnalyzedSecurityRisk can have 1 or more assets or a SecDA only has a TestReport if it is a top SecDA (i.e. is not the child of any other SecDA), and then it must have exactly one TestReport. If there are no such labels, then each instance in the domain must be related to exactly one instance in the range. Moreover, any two classes without subclass relationship are disjoint.

The domain and range axioms in Figures 4 and 5 complement these GCIs by fixing the intended source and target classes of the relevant properties, including hasAttackerProfile, hasExposureLevel, hasImpactLevel, hasLikelihoodFactor, hasSecurityRiskLevel, hasRationale, hasSafetyImpactAssessment, hasSecurityImpactAssessment, and isAffected. In particular, both hasSafetyImpactAssessment and hasSecurityImpactAssessment have domain ImpactAssessmentDomain. As shown by further GCIs introduced below, this makes it possible to associate impact assessments both with risks and control measures (security design arguments). Moreover, the range of isAffected is a Boolean, while an ImpactAssessment individual may also have associated a Rationale, which captures, as its name suggests, the rationale underlying a particular impact assessment.

$$dom(\mathsf{hasAttackerProfile}) \sqsubseteq \mathsf{LikelihoodFactor} \tag{5}$$
$$ran(\mathsf{hasAttackerProfile}) \sqsubseteq \mathsf{AttackerProfile} \tag{6}$$
$$dom(\mathsf{hasImpactLevel}) \sqsubseteq \mathsf{SecurityRiskLevel} \tag{7}$$
$$ran(\mathsf{hasImpactLevel}) \sqsubseteq \mathsf{ImpactLevel} \tag{8}$$

Figure 4. Range and domain restrictions for common aspects, part I.

$$dom(\mathsf{hasLikelihoodFactor}) \sqsubseteq \mathsf{SecurityRiskLevel} \tag{9}$$
$$ran(\mathsf{hasLikelihoodFactor}) \sqsubseteq \mathsf{LikelihoodFactor} \tag{10}$$
$$dom(\mathsf{hasRationale}) \sqsubseteq \mathsf{ImpactAssessment} \tag{11}$$
$$ran(\mathsf{hasRationale}) \sqsubseteq \mathsf{Rationale} \tag{12}$$
$$dom(\mathsf{hasSafetyImpactAssessment}) \sqsubseteq \mathsf{ImpactAssessmentDomain} \tag{13}$$
$$ran(\mathsf{hasSafetyImpactAssessment}) \sqsubseteq \mathsf{ImpactAssessment} \tag{14}$$
$$dom(\mathsf{hasSecurityImpactAssessment}) \sqsubseteq \mathsf{ImpactAssessmentDomain} \tag{15}$$
$$ran(\mathsf{hasSecurityImpactAssessment}) \sqsubseteq \mathsf{ImpactAssessment} \tag{16}$$
$$dom(\mathsf{hasSecurityRiskLevel}) \sqsubseteq \mathsf{SecurityRisk} \tag{17}$$
$$ran(\mathsf{hasSecurityRiskLevel}) \sqsubseteq \mathsf{SecurityRiskLevel} \tag{18}$$
$$dom(\mathsf{hasExposureLevel}) \sqsubseteq \mathsf{LikelihoodFactor} \tag{19}$$
$$ran(\mathsf{hasExposureLevel}) \sqsubseteq \mathsf{ExposureLevel} \tag{20}$$
$$dom(\mathsf{isAffected}) \sqsubseteq \mathsf{ImpactAssessment} \tag{21}$$
$$ran(\mathsf{isAffected}) \sqsubseteq \mathsf{Boolean} \tag{22}$$

Figure 5. Range and domain restrictions for common aspects, part II.

At the constraint level, the SHACL shapes in Figure 6 make these dependencies explicit for instance data. In particular, each LikelihoodFactor must have exactly one AttackerProfile and exactly one ExposureLevel, each SecurityRisk must have exactly one SecurityRiskLevel, and each SecurityRiskLevel must have exactly one ImpactLevel and exactly one LikelihoodFactor. Moreover, each ImpactAssessment must specify exactly one isAffected value and may have at most one Rationale.

$$\mathsf{ImpactAssessment} \leftarrow =_1 \mathsf{isAffected}.\top \wedge \leq_1 \mathsf{hasRationale}.\top \tag{23}$$
$$\mathsf{LikelihoodFactor} \leftarrow =_1 \mathsf{hasAttackerProfile}.\top \wedge =_1 \mathsf{hasExposureLevel}.\top \tag{24}$$
$$\mathsf{SecurityRisk} \leftarrow =_1 \mathsf{hasSecurityRiskLevel}.\top \tag{25}$$
$$\mathsf{SecurityRiskLevel} \leftarrow =_1 \mathsf{hasImpactLevel}.\top \wedge =_1 \mathsf{hasLikelihoodFactor}.\top \tag{26}$$

Figure 6. Shape constraints for common aspects.

Context The GCIs in Figure 7, together with the domain and range restrictions in Figure 8 capture the context-specific information used to characterize a security concern more precisely. The central class ContextSpecificThreat brings together four dimensions: the operating environment, the product, the attack type, and the protection goal. The two classes OperatingEnvironment and Product further structure this information: OperatingEnvironment groups physical, network, and system/platform security aspects, whereas Product groups identity and access management, data security and privacy, and operations and maintenance. The protection goals are given by the distinguished values availability, confidentiality, and integrity.

$$\exists \mathsf{hasOperatingEnvironment}.\top \sqcap \exists \mathsf{hasProduct}.\top \sqcap \exists \mathsf{hasAttackType}.\top \sqcap \exists \mathsf{hasProtectionGoal}.\top \sqsubseteq \mathsf{ContextSpecificThreat} \tag{27}$$

$$\exists \mathsf{hasPhysicalSecurity}.\top \sqcap \exists \mathsf{hasNetworkSecurity}.\top \sqcap \exists \mathsf{hasSystemAndPlatformSecurity}.\top \sqsubseteq \mathsf{OperatingEnvironment} \tag{28}$$

$$\exists \mathsf{hasIdentityAndAccessManagement}.\top \sqcap \exists \mathsf{hasDataSecurityAndPrivacy}.\top \sqcap \exists \mathsf{hasOperationsAndMaintenance}.\top \sqsubseteq \mathsf{Product} \tag{29}$$

$$\mathsf{ProtectionGoal}(\mathsf{availability}) \quad \mathsf{ProtectionGoal}(\mathsf{confidentiality}) \quad \mathsf{ProtectionGoal}(\mathsf{integrity}) \tag{30}$$

Figure 7. General concept inclusion axioms and concept assertions for context aspects.

$$dom(\mathsf{hasAttackType}) \sqsubseteq \mathsf{ContextSpecificThreat} \tag{31}$$
$$ran(\mathsf{hasAttackType}) \sqsubseteq \mathsf{AttackType} \tag{32}$$
$$dom(\mathsf{hasDataSecurityAndPrivacy}) \sqsubseteq \mathsf{Product} \tag{33}$$
$$ran(\mathsf{hasDataSecurityAndPrivacy}) \sqsubseteq \mathsf{DataSecurityAndPrivacy} \tag{34}$$
$$dom(\mathsf{hasIdentityAndAccessManagement}) \sqsubseteq \mathsf{Product} \tag{35}$$
$$ran(\mathsf{hasIdentityAndAccessManagement}) \sqsubseteq \mathsf{IdentityAndAccessManagement} \tag{36}$$
$$dom(\mathsf{hasNetworkSecurity}) \sqsubseteq \mathsf{OperatingEnvironment} \tag{37}$$
$$ran(\mathsf{hasNetworkSecurity}) \sqsubseteq \mathsf{NetworkSecurity} \tag{38}$$
$$dom(\mathsf{hasOperationsAndMaintenance}) \sqsubseteq \mathsf{Product} \tag{39}$$
$$ran(\mathsf{hasOperationsAndMaintenance}) \sqsubseteq \mathsf{OperationsAndMaintenance} \tag{40}$$
$$dom(\mathsf{hasOperatingEnvironment}) \sqsubseteq \mathsf{ContextSpecificThreat} \tag{41}$$
$$ran(\mathsf{hasOperatingEnvironment}) \sqsubseteq \mathsf{OperatingEnvironment} \tag{42}$$
$$dom(\mathsf{hasPhysicalSecurity}) \sqsubseteq \mathsf{OperatingEnvironment} \tag{43}$$
$$ran(\mathsf{hasPhysicalSecurity}) \sqsubseteq \mathsf{PhysicalSecurity} \tag{44}$$
$$dom(\mathsf{hasProduct}) \sqsubseteq \mathsf{ContextSpecificThreat} \tag{45}$$
$$ran(\mathsf{hasProduct}) \sqsubseteq \mathsf{Product} \tag{46}$$
$$dom(\mathsf{hasProtectionGoal}) \sqsubseteq \mathsf{ContextSpecificThreat} \tag{47}$$
$$ran(\mathsf{hasProtectionGoal}) \sqsubseteq \mathsf{ProtectionGoal} \tag{48}$$
$$dom(\mathsf{hasSystemAndPlatformSecurity}) \sqsubseteq \mathsf{OperatingEnvironment} \tag{49}$$
$$ran(\mathsf{hasSystemAndPlatformSecurity}) \sqsubseteq \mathsf{SystemAndPlatformSecurity} \tag{50}$$

Figure 8. Range and domain restrictions for context aspects.

The corresponding SHACL shapes in Figure 9 require these structures to be present explicitly in the data. Thus, each ContextSpecificThreat must specify exactly one value for each of its four dimensions, while each OperatingEnvironment and each Product must provide exactly one value for each of their respective subdimensions. In addition, every ProtectionGoal value must belong to {availability,confidentiality,integrity}.

$$\mathsf{ContextSpecificThreat} \leftarrow =_1 \mathsf{hasOperatingEnvironment}.\top \sqcap =_1 \mathsf{hasProduct}.\top \sqcap =_1 \mathsf{hasAttackType}.\top \sqcap =_1 \mathsf{hasProtectionGoal}.\top \tag{51}$$

$$\mathsf{OperatingEnvironment} \leftarrow =_1 \mathsf{hasPhysicalSecurity}.\top \sqcap =_1 \mathsf{hasNetworkSecurity}.\top \sqcap =_1 \mathsf{hasSystemAndPlatformSecurity}.\top \tag{52}$$

$$\mathsf{Product} \leftarrow =_1 \mathsf{hasIdentityAndAccessManagement}.\top \sqcap =_1 \mathsf{hasDataSecurityAndPrivacy}.\top \sqcap =_1 \mathsf{hasOperationsAndMaintenance}.\top \tag{53}$$

$$\mathsf{ProtectionGoal} \leftarrow \{\mathsf{availability}, \mathsf{confidentiality}, \mathsf{integrity}\} \tag{54}$$

Figure 9. Shape constraints for context aspects.

Assessment The GCIs in Figure 10 capture the essential structure of analyzed security risks. In particular, every individual that is related to a context-specific threat, an impact, an initial security risk level, a component, and at least one asset is characterized as an AnalyzedSecurityRisk. Moreover, the property hasInitialSecurityRiskLevel is specified as a subproperty of hasSecurityRiskLevel, which implies that every AnalyzedSecurityRisk is also a SecurityRisk (see Figure 3), while Impact is placed under ImpactAssessmentDomain, thereby linking impacts – and indirectly also analyzed risks – to the common impact-assessment pattern introduced above.

$$\mathsf{hasInitialSecurityRiskLevel} \sqsubseteq \mathsf{hasSecurityRiskLevel} \tag{55}$$

$$\mathsf{Impact} \sqsubseteq \mathsf{ImpactAssessmentDomain} \tag{56}$$

$$\exists \mathsf{hasContextSpecificThreat}.\top \sqcap \exists \mathsf{hasImpact}.\top \sqcap \exists \mathsf{hasInitialSecurityRiskLevel}.\top \sqcap \exists \mathsf{hasComponent}.\top \sqcap \exists \mathsf{hasAsset}.\top \sqsubseteq \mathsf{AnalyzedSecurityRisk} \tag{57}$$

Figure 10. General concept and role inclusion axioms for assessment aspects.

Complementing this conceptual characterization, the domain and range restrictions in Figure 11 determine the intended typing of these relations by assigning AnalyzedSecurityRisk as their domain and ContextSpecificThreat, Impact, SecurityRiskLevel, Component, and Asset as the corresponding ranges.

$$dom(\mathsf{hasAsset}) \sqsubseteq \mathsf{AnalyzedSecurityRisk} \tag{58}$$
$$ran(\mathsf{hasAsset}) \sqsubseteq \mathsf{Asset} \tag{59}$$
$$dom(\mathsf{hasComponent}) \sqsubseteq \mathsf{AnalyzedSecurityRisk} \tag{60}$$
$$ran(\mathsf{hasComponent}) \sqsubseteq \mathsf{Component} \tag{61}$$
$$dom(\mathsf{hasContextSpecificThreat}) \sqsubseteq \mathsf{AnalyzedSecurityRisk} \tag{62}$$
$$ran(\mathsf{hasContextSpecificThreat}) \sqsubseteq \mathsf{ContextSpecificThreat} \tag{63}$$
$$dom(\mathsf{hasImpact}) \sqsubseteq \mathsf{AnalyzedSecurityRisk} \tag{64}$$
$$ran(\mathsf{hasImpact}) \sqsubseteq \mathsf{Impact} \tag{65}$$
$$dom(\mathsf{hasInitialSecurityRiskLevel}) \sqsubseteq \mathsf{AnalyzedSecurityRisk} \tag{66}$$
$$ran(\mathsf{hasInitialSecurityRiskLevel}) \sqsubseteq \mathsf{SecurityRiskLevel} \tag{67}$$

Figure 11. Range and domain restrictions for assessment aspects.

The ontological axioms are mirrored at the constraint level by the SHACL shape in Figure 12, which makes the intended dependencies explicit for data: each AnalyzedSecurityRisk must specify exactly one context-specific threat, exactly one initial security risk level, and exactly one component, while requiring at least one impact and at least one asset.

$$\begin{aligned}\mathsf{AnalyzedSecurityRisk} \leftarrow {} & =_1 \mathsf{hasContextSpecificThreat}.\top \wedge \geq_1 \mathsf{hasImpact}.\top \\ & \wedge =_1 \mathsf{hasInitialSecurityRiskLevel}.\top \wedge =_1 \mathsf{hasComponent}.\top \\ & \wedge \geq_1 \mathsf{hasAsset}.\top\end{aligned} \tag{68}$$

Figure 12. Shape constraints for assessment aspects.

Control The main GCI in Figure 13 defines controlled security risks: every individual that is related to an analyzed security risk, a residual security risk level, and a secure design argument is characterized as a ControlledSecurityRisk. In addition, and analogously to hasInitialSecurityRiskLevel for analyzed risks, hasResidualSecurityRiskLevel is introduced as a subproperty of hasSecurityRiskLevel, from which it follows that controlled security risks are also security risks.

$$\mathsf{hasResidualSecurityRiskLevel} \sqsubseteq \mathsf{hasSecurityRiskLevel} \tag{69}$$
$$\begin{aligned}\exists \mathsf{hasAnalyzedSecurityRisk}.\top \sqcap \exists \mathsf{hasResidualSecurityRiskLevel}.\top \\ \sqcap \exists \mathsf{isMitigatedBy}.\top \sqsubseteq \mathsf{ControlledSecurityRisk}\end{aligned} \tag{70}$$

Figure 13. General concept and role inclusion for control aspects.

The domain and range restrictions in Figure 14 again fix the expected typing of the associated relations, with ControlledSecurityRisk as domain and AnalyzedSecurityRisk, SecurityRiskLevel, and SecDA as the corresponding ranges.

$$\mathit{dom}(\mathsf{hasAnalyzedSecurityRisk}) \sqsubseteq \mathsf{ControlledSecurityRisk} \tag{71}$$
$$\mathit{ran}(\mathsf{hasAnalyzedSecurityRisk}) \sqsubseteq \mathsf{AnalyzedSecurityRisk} \tag{72}$$
$$\mathit{dom}(\mathsf{hasResidualSecurityRiskLevel}) \sqsubseteq \mathsf{ControlledSecurityRisk} \tag{73}$$
$$\mathit{ran}(\mathsf{hasResidualSecurityRiskLevel}) \sqsubseteq \mathsf{SecurityRiskLevel} \tag{74}$$
$$\mathit{dom}(\mathsf{isMitigatedBy}) \sqsubseteq \mathsf{ControlledSecurityRisk} \tag{75}$$
$$\mathit{ran}(\mathsf{isMitigatedBy}) \sqsubseteq \mathsf{SecDA} \tag{76}$$

Figure 14. Range and domain restrictions for control aspects.

Finally, the SHACL shape in Figure 15 formulates the corresponding constraints at the data level by requiring each ControlledSecurityRisk to specify exactly one analyzed security risk, exactly one residual security risk level, and exactly one secure design argument.

$$\begin{aligned}\mathsf{ControlledSecurityRisk} \leftarrow {} & =_1 \mathsf{hasAnalyzedSecurityRisk}.\top \wedge =_1 \mathsf{hasResidualSecurityRiskLevel}.\top \\ & \wedge =_1 \mathsf{isMitigatedBy}.\top\end{aligned} \tag{77}$$

Figure 15. Shape constraints for control aspects.

Secure Design Arguments The GCIs in Figure 16 define the central structure of secure design arguments and their refinements. In particular, every individual that is related to a strategy is characterized as a SecDA, while every SecDA with an implementation manifest is classified as a SecDAI. The remaining GCIs introduce the assurance-specific specializations AssuranceSecDA and AssuranceSecDAI. In addition, the axiom $\mathsf{SecDA} \sqsubseteq \mathsf{ImpactAssessmentDomain}$ places secure design arguments into the common impact-assessment pattern introduced earlier.

$$\mathsf{SecDA} \sqcap \exists \mathsf{hasAssurance}.\top \sqsubseteq \mathsf{AssuranceSecDA} \tag{78}$$
$$\mathsf{SecDAI} \sqcap \mathsf{AssuranceSecDA} \sqsubseteq \mathsf{AssuranceSecDAI} \tag{79}$$
$$\mathsf{SecDA} \sqsubseteq \mathsf{ImpactAssessmentDomain} \tag{80}$$
$$\exists \mathsf{hasStrategy}.\top \sqsubseteq \mathsf{SecDA} \tag{81}$$
$$\mathsf{SecDA} \sqcap \exists \mathsf{hasImplementationManifest}.\top \sqsubseteq \mathsf{SecDAI} \tag{82}$$

Figure 16. General concept and role inclusion axioms for secure design argument aspects.

The domain and range restrictions in Figure 17 fix the expected typing of the relations involving secure design arguments, with SecDA as domain and, depending on the property, assumptions, assurances, intervention methods, implementation manifests, locations and organisational contexts of intervention, strategies, subordinate secure design arguments, and test reports as corresponding ranges.

$$dom(\mathsf{hasAssumption}) \sqsubseteq \mathsf{SecDA} \tag{83}$$
$$ran(\mathsf{hasAssumption}) \sqsubseteq \mathsf{Assumption} \tag{84}$$
$$dom(\mathsf{hasAssurance}) \sqsubseteq \mathsf{SecDA} \tag{85}$$
$$ran(\mathsf{hasAssurance}) \sqsubseteq \mathsf{Assurance} \tag{86}$$
$$dom(\mathsf{hasInterventionMethod}) \sqsubseteq \mathsf{SecDA} \tag{87}$$
$$ran(\mathsf{hasInterventionMethod}) \sqsubseteq \mathsf{InterventionMethod} \tag{88}$$
$$dom(\mathsf{hasImplementationManifest}) \sqsubseteq \mathsf{SecDA} \tag{89}$$
$$ran(\mathsf{hasImplementationManifest}) \sqsubseteq \mathsf{ImplementationManifest} \tag{90}$$
$$dom(\mathsf{hasLocationOfIntervention}) \sqsubseteq \mathsf{SecDA} \tag{91}$$
$$ran(\mathsf{hasLocationOfIntervention}) \sqsubseteq \mathsf{LocationOfIntervention} \tag{92}$$
$$dom(\mathsf{hasOrganisationalContextOfIntervention}) \sqsubseteq \mathsf{SecDA} \tag{93}$$
$$ran(\mathsf{hasOrganisationalContextOfIntervention}) \sqsubseteq \mathsf{OrganisationalContextOfIntervention} \tag{94}$$
$$dom(\mathsf{hasStrategy}) \sqsubseteq \mathsf{SecDA} \tag{95}$$
$$ran(\mathsf{hasStrategy}) \sqsubseteq \mathsf{Strategy} \tag{96}$$
$$dom(\mathsf{hasSubSecDA}) \sqsubseteq \mathsf{SecDA} \tag{97}$$
$$ran(\mathsf{hasSubSecDA}) \sqsubseteq \mathsf{SecDA} \tag{98}$$
$$dom(\mathsf{hasTestReport}) \sqsubseteq \mathsf{SecDA} \tag{99}$$
$$ran(\mathsf{hasTestReport}) \sqsubseteq \mathsf{TestReport} \tag{100}$$

Figure 17. Range and domain restrictions for secure design argument aspects.

Once more, the SHACL constraints in Figure 18, which include several helper constraints, specify the structural constraints of secure design arguments at the data level. They require each SecDA to specify exactly one strategy, while allowing at most one location of intervention, at most one assumption, at most one implementation manifest, and at most one assurance. Moreover, the recursive organization of secure design arguments is constrained explicitly: every SecDA must have at least one implementation-level argument reachable via the hasSubSecDA relation, which implies in particular that every leaf secure design argument must have an implementation manifest. In addition, top-level arguments must have exactly one test report, whereas non-top-level arguments must have none, and only leaf arguments may carry impact and intervention related information as well as assumptions. For assurance-specific arguments, the corresponding SHACL shape further requires exactly one assurance and propagates the assurance character recursively to subordinate arguments. Moreover, the shapes for SecDAI and AssuranceSecDAI require exactly one implementation manifest. In this respect, the subdivision of secure design arguments into assurance-specific and implementation-specific arguments, as well as the corresponding constraints, parallels that of safe design arguments from [2]; for convenience, however, these structures are duplicated explicitly in the SECUMAN shapes.

$$\mathsf{TopSecDA} \leftarrow \leq_0 \mathsf{hasSubSecDA}^-.\top \tag{101}$$

$$\mathsf{NotTopSecDA} \leftarrow \geq_1 \mathsf{hasSubSecDA}^-.\top \tag{102}$$

$$\mathsf{LeafSecDA} \leftarrow \leq_0 \mathsf{hasSubSecDA}.\top \tag{103}$$

$$\mathsf{NotLeafSecDA} \leftarrow \geq_1 \mathsf{hasSubSecDA}.\top \tag{104}$$

$$\mathsf{TopSecDAAndOneTestReport} \leftarrow \mathsf{TopSecDA} \wedge =_1 \mathsf{hasTestReport}.\top \tag{105}$$

$$\mathsf{NotTopSecDAAndNoTestReport} \leftarrow \mathsf{NotTopSecDA} \wedge \leq_0 \mathsf{hasTestReport}.\top \tag{106}$$

$$\begin{aligned}\mathsf{LeafSecDAAndOneA} \leftarrow\ & \mathsf{LeafSecDA} \wedge =_1 h.\top \\ & \text{for all } (A,h) \text{ in} \\ & \{(\mathsf{InterventionMethod}, \mathsf{hasInterventionMethod}), \\ & (\mathsf{OrganisationalContextOfIntervention}, \\ & \mathsf{hasOrganisationalContextOfIntervention}), \\ & (\mathsf{SafetyImpactAssessment}, \mathsf{hasSafetyImpactAssessment}), \\ & (\mathsf{SecurityImpactAssessment}, \mathsf{hasSecurityImpactAssessment})\}\end{aligned} \tag{107}$$

$$\begin{aligned}\mathsf{NotLeafSecDAAndNoA} \leftarrow\ & \mathsf{NotLeafSecDA} \wedge \leq_0 h.\top \\ & \text{for all } (A,h) \text{ in} \\ & \{(\mathsf{Assumption}, \mathsf{hasAssumption}), \\ & (\mathsf{InterventionMethod}, \mathsf{hasInterventionMethod}), \\ & (\mathsf{LocationOfIntervention}, \mathsf{hasLocationOfIntervention}), \\ & (\mathsf{OrganisationalContextOfIntervention}, \\ & \mathsf{hasOrganisationalContextOfIntervention}), \\ & (\mathsf{SafetyImpactAssessment}, \mathsf{hasSafetyImpactAssessment}), \\ & (\mathsf{SecurityImpactAssessment}, \mathsf{hasSecurityImpactAssessment})\}\end{aligned} \tag{108}$$

$$\begin{aligned}\mathsf{AssuranceSecDA} \leftarrow\ & \forall \mathsf{hasSubSecDA}.\mathsf{AssuranceSecDA} \\ & \wedge =_1 \mathsf{hasAssurance}.\top\end{aligned} \tag{109}$$

$$\begin{aligned}\mathsf{AssuranceSecDAI} \leftarrow\ & =_1 \mathsf{hasAssurance}.\top \\ & \wedge =_1 \mathsf{hasImplementationManifest}.\top\end{aligned} \tag{110}$$

$$\begin{aligned}\mathsf{SecDA} \leftarrow\ & =_1 \mathsf{hasStrategy}.\top \\ & \wedge \leq_1 h.\top \text{ for all } h \text{ in} \\ & \quad \{\mathsf{hasLocationOfIntervention}, \mathsf{hasAssumption}, \\ & \quad \mathsf{hasImplementationManifest}, \mathsf{hasAssurance}\} \\ & \wedge \exists \mathsf{hasSubSecDA}^*.\mathsf{SecDAI} \\ & \wedge (\mathsf{NotTopSecDA} \vee \mathsf{TopSecDAAndOneTestReport}) \\ & \wedge (\mathsf{TopSecDA} \vee \mathsf{NotTopSecDAAndNoTestReport}) \\ & \wedge (\mathsf{NotLeafSecDA} \vee \mathsf{LeafSecDAAndOneA}) \\ & \text{for all } A \text{ in} \\ & \{\mathsf{InterventionMethod}, \\ & \mathsf{OrganisationalContextOfIntervention}, \\ & \mathsf{SafetyImpactAssessment}, \\ & \mathsf{SecurityImpactAssessment}\} \\ & \wedge (\mathsf{LeafSecDA} \vee \mathsf{NotLeafSecDAAndNoA}) \\ & \text{for all } A \text{ in} \\ & \{\mathsf{Assumption}, \\ & \mathsf{InterventionMethod}, \\ & \mathsf{LocationOfIntervention}, \\ & \mathsf{OrganisationalContextOfIntervention}, \\ & \mathsf{SafetyImpactAssessment}, \\ & \mathsf{SecurityImpactAssessment}\}\end{aligned} \tag{111}$$

$$\mathsf{SecDAI} \leftarrow =_1 \mathsf{hasImplementationManifest}.\top \tag{112}$$

Figure 18. Shape constraints for secure design argument aspects.

Disjointness Finally, the disjointness axioms in Figure 19 operate at two conceptual levels. First, pairwise disjointness is imposed on the ontology's primitive classes, i.e., classes that are not introduced as subclasses of other domain classes and therefore serve as basic modeling categories. Second, further disjointness axioms are stated for specialized classes that refine more general concepts. In particular, AnalyzedSecurityRisk and ControlledSecurityRisk, both subsumed by SecurityRisk, are required to be disjoint, and Impact and SecDA, both subsumed by ImpactAssessmentDomain, are likewise required to be disjoint. Together, these axioms exclude unintended class overlap both among primitive categories and among their specialized refinements.

$$A_i \sqcap A_j \sqsubseteq \bot \text{ for all distinct } A_i, A_j \text{ in} \\ \{\mathsf{Assumption}, \mathsf{Assurance}, \mathsf{Asset}, \\ \mathsf{AttackType}, \mathsf{AttackerProfile}, \mathsf{Component}, \\ \mathsf{ContextSpecificThreat}, \mathsf{DataSecurityAndPrivacy}, \\ \mathsf{IdentityAndAccessManagement}, \mathsf{ImpactAssessment}, \\ \mathsf{ImpactAssessmentDomain}, \mathsf{ImpactLevel}, \\ \mathsf{ImplementationManifest}, \mathsf{InterventionMethod}, \\ \mathsf{LikelihoodFactor}, \mathsf{LocationOfIntervention}, \\ \mathsf{NetworkSecurity}, \mathsf{OperatingEnvironment}, \\ \mathsf{OperationsAndMaintenance}, \\ \mathsf{OrganisationalContextOfIntervention}, \\ \mathsf{PhysicalSecurity}, \mathsf{Product}, \mathsf{ProtectionGoal}, \mathsf{Rationale}, \\ \mathsf{SecurityRisk}, \mathsf{SecurityRiskLevel}, \mathsf{Strategy}, \\ \mathsf{SystemAndPlatformSecurity}, \mathsf{TestReport}, \\ \mathsf{ExposureLevel}\} \tag{113}$$

$$\mathsf{AnalyzedSecurityRisk} \sqcap \mathsf{ControlledSecurityRisk} \sqsubseteq \bot \tag{114}$$

$$\mathsf{Impact} \sqcap \mathsf{SecDA} \sqsubseteq \bot \tag{115}$$

Figure 19. Disjointness axioms.

5. Usage & Extensibility

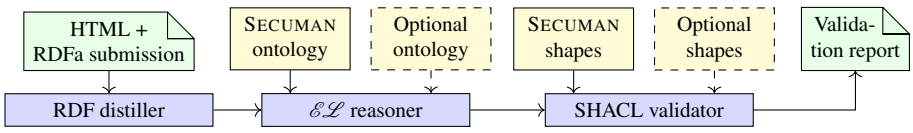

Figure 20. Overview of the SECUMAN validation pipeline.

Figure 20 shows the validation pipeline for risk management documentation of medical devices proposed in [2], which we adapt here for risk management documentation concerning security aspects. As noted above, the approach assumes that risk documentation is submitted as HTML files containing risk management data encoded as RDF triples and expressed using the terminology of the SECUMAN ontology. An *RDF distiller* extracts

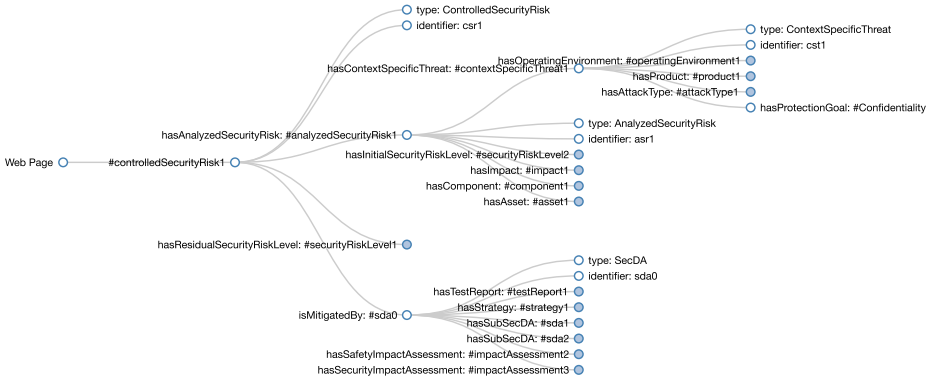

Figure 21. Partial graphical representation of the RDF for the security risk from Figure 1.

the RDF graph from the HTML by removing the markup and isolating the RDF triples – see Figure 21 for a graphical depiction of the RDF encoding the security risk rendered via HTML in Figure 1. The RDF triples are then used as input to an OWL ($\mathscr{EL}$ profile) *reasoner* together with SECUMAN and, where needed, additional ontologies. As a result, a materialized knowledge base is obtained and subsequently validated against the SECUMAN SHACL constraints, and potentially additional constraints, using a *SHACL validator*. The result is then communicated in the form of a human-readable *validation report*.

In our extension of the validation pipeline for RISKMAN (see the URL in Section 4.1), we in particular use the HermiT reasoner [46] for materialization, although additional options are provided through a realization wrapper that we also make available online[4]. For SHACL validation, we use PySHACL [54].

Like RISKMAN, the SECUMAN ontology and shapes are intentionally lightweight so that the validation pipeline can be adapted easily to different modelling needs by incorporating additional ontologies and shapes, as already indicated above. One area in which the ontology will clearly require extension is the modelling of security risk levels, for example attacker profiles, exposure levels, and impact levels, as well as the way in which these are combined to determine whether a risk level is acceptable or unacceptable. A minimalist "plugin" for adding such levels to the SECUMAN ontology is also provided in our validation pipeline. In this setting, the user may enrich the SECUMAN ontology with an ontology $\mathscr{K}^{a\text{-}v\text{-}i}_{\pi,\sigma,\tau} := \mathscr{T}_{\mathsf{gt}} \cup \mathscr{A}_{\pi,\sigma,\tau}$ for π attacker profiles, σ exposure levels, and τ impact levels ($\pi, \sigma, \tau \in \mathbb{N}$). This ontology not only introduces individual profiles and levels represented by nominals, but also an ordering gt over these profiles and levels:

[4] `https://github.com/cl-tud/realization-wrapper`

$$
\begin{aligned}
\mathscr{A}_{\pi,\sigma,\tau} = {} & \{\,\mathsf{AttackerProfile}(\mathsf{a}_i) \mid 1 \le i \le \pi\} \cup \\
& \{\mathsf{ExposureLevel}(\mathsf{e}_i) \mid 1 \le i \le \sigma\} \cup \\
& \{\mathsf{ImpactLevel}(\mathsf{s}_i) \mid 1 \le i \le \tau\} \cup \\
& \{\mathsf{gt}(\mathsf{a}_{i+1}, \mathsf{a}_i) \mid 1 \le i < \pi\} \cup \\
& \{\mathsf{gt}(\mathsf{e}_{i+1}, \mathsf{e}_i) \mid 1 \le i < \sigma\} \cup \\
& \{\mathsf{gt}(\mathsf{i}_{i+1}, \mathsf{i}_i) \mid 1 \le i < \tau\} \\
\mathscr{T}_{\mathsf{gt}} = {} & \{\mathit{tra}(\mathsf{gt})\}
\end{aligned}
$$

To encode a concrete security risk matrix – that is, the way in which risk levels are derived from combinations of attacker profile, exposure level, and impact level – further axioms would need to be added to $\mathscr{T}_{\mathsf{gt}}$, since such matrices are typically manufacturer-specific and therefore do not form part of SECUMAN. As an example, the following axioms first identify critical attacker profiles, exposure levels, and impact levels, and then use these classifications in the last two GCIs to determine critical likelihood factors and, subsequently, critical security levels:

$$\exists \mathsf{gt}.\{\mathsf{a}_2\} \sqsubseteq \mathsf{CriticalAP}$$

$$\exists \mathsf{gt}.\{\mathsf{e}_3\} \sqsubseteq \mathsf{CriticalEL}$$

$$\exists \mathsf{gt}.\{\mathsf{i}_2\} \sqsubseteq \mathsf{CriticalIL}$$

$$\exists \mathsf{hasAttackerProfile}.\mathsf{CriticalAP} \sqcap \exists \mathsf{hasExposureLevel}.\mathsf{CriticalEL} \sqsubseteq \mathsf{CriticalLF}$$

$$\exists \mathsf{hasLikelihoodFactor}.\mathsf{CriticalLF} \sqcap \exists \mathsf{hasImpactLevel}.\mathsf{CriticalIL} \sqsubseteq \mathsf{CriticalRiskLevel}$$

Checking for controlled security risks with critical residual risk levels can then be achieved by adding the SHACL constraint

$$\mathsf{ControlledSecurityRisk} \leftarrow \neg(\exists \mathsf{hasResidualSecurityRiskLevel}.\mathsf{CriticalRiskLevel})$$

to the SECUMAN shapes. Of course, more complex conditions for deriving and detecting critical security risk levels can likewise be expressed by adding further axioms and shapes.

6. Conclusion

We presented the SECUMAN ontology and accompanying SHACL shapes for representing and analysing cybersecurity risk-management documentation for medical devices. Analysed risks and their mitigations are represented as an $\mathscr{EL}^{++}$ ABox; the SECUMAN ontology is used together with an $\mathscr{EL}$ reasoner to infer implicit knowledge; and SHACL constraints are used to check whether the resulting data conform to the intended documentation model. The ontology, shapes, and a reference implementation of the complete validation pipeline are freely available.

The SECUMAN ontology complements the existing RISKMAN ontology by targeting security rather than safety risks of medical devices. It was designed with the ultimate objective of enabling the combined analysis of safety and security risk-management documentation. This objective is reflected not only in the similar structure of the RISKMAN and SECUMAN ontologies, but also, for instance, in the explicit safety-

impact assessments associated with both analysed security risks and SecDAs in the SECUMAN ontology. The next step is therefore to develop a bridging ontology for safety and security aspects. Such an ontology should make it possible, for example, to propagate a safety impact identified for a security control measure into the safety-risk-management domain, where the resulting risk can then be analysed and controlled.

Both RISKMAN and SECUMAN are lightweight ontologies intended to support first-pass validation and, in particular, to facilitate the analysis, navigation, and reuse of medical-device risk-management documentation. More substantive automated analysis will require integration with additional domain ontologies and structured cybersecurity resources; see, e.g., [55,56,57,58,59,60,61], as well as the existing medical-device-specific cybersecurity ontologies and knowledge graphs discussed in Section 2. Quality assurance is a particularly significant challenge [62,63]; possible techniques include dialogue-inspired explication methods from computational argumentation [64,65], as well as other approaches for providing justifications in logic-based knowledge-representation formalisms [66].

Finally, the systematic reuse of risk-management documentation remains an important open challenge. We believe that a shared, Semantic Web-based repository of risk-management documentation, including reusable secure and safe design arguments, could substantially accelerate both the creation and the evaluation of such documentation. This potential is especially promising when combined with large language models in retrieval-augmented generation setups [67,68]. Realising this vision, however, will require appropriate incentives for sharing risk-management knowledge, as well as infrastructure that supports controlled access and reuse without exposing manufacturer secrets.

Acknowledgements.

This work was supported by funding from BMFTR (Federal Ministry of Research, Technology and Space) within the project SEMECO (grant no. 03ZU1210BG).

References

[1] Regulation (EU) 2017/745 of the European Parliament and of the Council of 5 April 2017 on medical devices; 2017. Available from: `http://data.europa.eu/eli/reg/2017/745/oj`.

[2] Gorczyca P, Arndt D, Diller M, Hampe J, Heidenreich G, Kettmann P, et al. Supporting risk management for medical devices via the RISKMAN ontology and shapes. In: SEMANTiCS. Studies on the Semantic Web. IOS Press; 2025. p. 226-46.

[3] International Organization for Standardization. ISO 14971:2019: Medical devices – application of risk management to medical devices; 2019. Published by ISO, Geneva, Switzerland. Available from: `https://www.iso.org/standard/72704.html`.

[4] Arndt D, Bank P, Gorczyca P, Heidenreich G, Kettmann P, Kobel I, et al.. VDE SPEC 90025 V1.0 (en): Medical devices – risk management: Framework of a computerized risk analysis format for transmission and submission (MD-CRAFTS); 2024. VDE Verband der Elektrotechnik Elektronik Informationstechnik e.V. Available from: `https://www.vde.com/resource/blob/2323050/6df908c47e69cd01126530ab07e78a52/vde-spec-90025-v1-0--en--data.pdf`.

[5] Kang M, Park E, Cho BH, Lee KS. Recent patient health monitoring platforms incorporating internet of things-enabled smart devices. International neurourology journal. 2018;22(Suppl 2):S76.

[6] Schwartz S, Ross A, Carmody S, Chase P, Coley SC, Connolly J, et al. The evolving state of medical device cybersecurity. Biomedical instrumentation & technology. 2018;52(2):103-11.

[7] AP News. Cyberattack hits major hospital in Spanish city of Barcelona; 2023. Accessed: April 27, 2026. Available from: `https://apnews.com/article/barcelona-hospital-cyberattack-ransomware-37e0fee33798c56459e63866ca8b449f`.

[8] Campbell D, Hern A. Services disrupted as London hospitals hit by cyber-attack; 2024. The Guardian. Available from: https://www.theguardian.com/society/article/2024/jun/04/cyber-attack-london-hospitals.
[9] Mashinchi MI, Acton T, Datta PM. When healthcare becomes sick: Recovering from ransomware. Journal of Information Technology Teaching Cases. 2024:20438869241279443.
[10] Alder S. 2024 healthcare data breach report; 2025. The HIPAA Journal. Available from: https://www.hipaajournal.com/2024-healthcare-data-breach-report/.
[11] Alder S. Healthcare data breach statistics; 2026. The HIPAA Journal. Available from: https://www.hipaajournal.com/healthcare-data-breach-statistics/.
[12] ENISA. Health; 2025. Accessed: April 27, 2026. Available from: https://www.enisa.europa.eu/topics/cybersecurity-of-critical-sectors/health.
[13] European Commission. Cybersecurity in healthcare; 2026. Accessed: April 27, 2026. Available from: https://commission.europa.eu/topics/digital-economy-and-society/cybersecurity-healthcare_en.
[14] Medical Device Coordination Group. MDCG 2019-16: Guidance on cybersecurity for medical devices; 2020. MDCG 2019-16.
[15] International Electrotechnical Commission. IEC 81001-5-1:2021: Health software and health IT systems safety, effectiveness and security – part 5-1: Security – activities in the product life cycle; 2021. IEC 81001-5-1:2021. Available from: https://www.iso.org/standard/76097.html.
[16] International Medical Device Regulators Forum. IMDRF/CYBER WG/N60FINAL:2020: Principles and practices for medical device cybersecurity; 2020. IMDRF/CYBER WG/N60FINAL:2020.
[17] Association for the Advancement of Medical Instrumentation. AAMI TIR57:2016/(R)2023: Principles for medical device security – risk management; 2016. Reaffirmed 2023.
[18] Martinez J, Godot J, Ruiz A, Balbis A, Ruiz Nolasco R. Safety and security interference analysis in the design stage. In: International Conference on Computer Safety, Reliability, and Security. Springer; 2020. p. 54-68.
[19] Association for the Advancement of Medical Instrumentation. ANSI/AAMI SW96:2023: Standard for medical device security – security risk management for device manufacturers; 2023. ANSI/AAMI SW96:2023.
[20] Joint Task Force Transformation Initiative. NIST Special Publication 800-30 Revision 1: Guide for conducting risk assessments. National Institute of Standards and Technology; 2012. NIST SP 800-30 Rev. 1.
[21] Uciteli A, Neumann J, Tahar K, Saleh K, Stucke S, Faulbrück-Röhr S, et al. Ontology-based specification, identification and analysis of perioperative risks. J Biomed Semant. 2017;8(1):36:1-36:14.
[22] Herre H. General formal ontology (GFO): A foundational ontology for conceptual modelling. In: Poli R, Healy M, Kameas A, editors. Theory and Applications of Ontology: Computer Applications. Dordrecht: Springer Netherlands; 2010. p. 297-345.
[23] Herre H, Heller B, Burek P, Hoehndorf R, Loebe F, Michalek H. General formal ontology (GFO): A foundational ontology integrating objects and processes. Part I: Basic principles (version 1.0). University of Leipzig: Research Group Ontologies in Medicine (Onto-Med); 2006. Onto-Med Report.
[24] Kim DY, Park YS, Lee B, Lee JW. Ontology-based process integration incorporating reference associations between medical standards from the perspective of medical software developers. Journal of Ambient Intelligence and Humanized Computing. 2019.
[25] Schütz AE, Fertig T, Weber K. Defining a core ontology for medical devices in Germany to ensure semantic interoperability. In: Simian D, Stoica LF, editors. Modelling and Development of Intelligent Systems – 7th International Conference, MDIS 2020, Sibiu, Romania, October 22-24, 2020, Revised Selected Papers. vol. 1341 of Communications in Computer and Information Science. Springer; 2020. p. 394-410.
[26] Chattoraj S, Walid R, Joshi KP. Semantically rich approach to automating regulations of medical devices. In: ICDH. IEEE; 2024. p. 132-7.
[27] Zhu W, Zhang P, Xia W, Gao Z, Li W, Tian R, et al. AI-driven medical device risk management: A new paradigm integrating large language models and prompt engineering for standard-risk knowledge graph construction and application. Risk Management and Healthcare Policy. 2026;19:571156. Available from: https://www.tandfonline.com/doi/abs/10.2147/RMHP.S571156.
[28] Rosa FdF, Bonacin R, Jino M. The security assessment domain: A survey of taxonomies and ontologies. arXiv preprint arXiv:170609772. 2017.

[29] Adach M, Hänninen K, Lundqvist K. Security ontologies: A systematic literature review. In: International Conference on Enterprise Design, Operations, and Computing. Springer; 2022. p. 36-53.
[30] Hollerer S, Sauter T, Kastner W. A survey of ontologies considering general safety, security, and operation aspects in OT. IEEE Open Journal of the Industrial Electronics Society. 2024;5:861-85.
[31] Jarwar MA, Freng JWC, Ali S. Modelling industrial IoT security using ontologies: A systematic review. IEEE Open Journal of the Communications Society. 2025.
[32] Lourenço B, Adão P, Ferreira JF, Marques MM, Vaz C. Structuring security: A survey of cybersecurity ontologies, semantic log processing, and LLMs application. CoRR. 2025;abs/2510.16610.
[33] Sills M, Ranade P, Mittal S. Cybersecurity threat intelligence augmentation and embedding improvement – a healthcare use case. In: IEEE International Conference on Intelligence and Security Informatics (ISI). IEEE; 2020. p. 1-6.
[34] Hannou F, Atigui F, Lammari N, Cherfi SS. SafecareOnto: A cyber-physical security ontology for healthcare systems. In: DEXA (2). Lecture Notes in Computer Science. Springer; 2021. p. 22-34.
[35] Dart M, Ahmed M. CYBER-AIDD: A novel approach to implementing improved cybersecurity resilience for large Australian healthcare providers using a Unified Modelling Language ontology. Digital Health. 2023;9:20552076231191095.
[36] Liu J, Sasaki R, Uehara T. An ontology-based framework for medical IoT forensic evidence. In: QRS Companion. IEEE; 2023. p. 863-4.
[37] Bughio KS, Cook DM, Shah SAA. Developing a novel ontology for cybersecurity in Internet of Medical Things-enabled remote patient monitoring. Sensors. 2024;24(9):2804.
[38] Bughio KS, Cook DM, Shah SAA. Novel knowledge graph-based modeling for vulnerability detection in the Internet of Medical Things. In: Recent Challenges in Intelligent Information and Database Systems. Communications in Computer and Information Science. Springer; 2024. p. 314-25.
[39] Ghosh R, von Stockhausen H, Schmitt M, Marica VG, Karn SK, Farri O. CVE-LLM: Ontology-assisted automatic vulnerability evaluation using large language models. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39. AAAI Press; 2025. p. 28757-65.
[40] Chander AS, Harshavardhana M, Sathyanarayana N, Hemalatha K, Gagana B. Intelligent cybersecurity ontology framework for Internet of Medical Things-enabled remote patient monitoring. In: ITM Web of Conferences. vol. 79. EDP Sciences; 2025. p. 01030.
[41] Adida B, Birbeck M, McCarron S, Herman I. RDFa core 1.1 – third edition, 17 March 2015; 2015. W3C Recommendation. Available from: `http://www.w3.org/TR/rdfa-core/`.
[42] International Organization for Standardization. ISO/IEC/IEEE 15026-1:2019: Systems and software engineering – systems and software assurance – part 1: Concepts and vocabulary; 2019. ISO/IEC/IEEE 15026-1:2019.
[43] International Organization for Standardization. ISO/TS 81001-2-1:2025: Health software and health IT systems safety, effectiveness and security – part 2-1: Coordination – guidance and requirements for the use of assurance cases for safety and security; 2025. ISO/TS 81001-2-1:2025. Available from: `https://www.iso.org/standard/82210.html`.
[44] Poveda-Villalón M, Fernández-Izquierdo A, Fernández-López M, García-Castro R. LOT: An industrial-oriented ontology engineering framework. Engineering Applications of Artificial Intelligence. 2022 may;111:104755.
[45] Suárez-Figueroa MC, Gómez-Pérez A, Fernández-López M. The NeOn methodology framework: A scenario-based methodology for ontology development. Appl Ontology. 2015;10:107-45. Available from: `https://api.semanticscholar.org/CorpusID:5094899`.
[46] Shearer RD, Motik B, Horrocks I. HermiT: A highly efficient OWL reasoner. In: Owled. vol. 432; 2008. p. 91.
[47] Poveda-Villalón M, Gómez-Pérez A, Suárez-Figueroa MC. OOPS! (OntOlogy Pitfall Scanner!): An online tool for ontology evaluation. International Journal on Semantic Web and Information Systems (IJSWIS). 2014;10(2):7-34.
[48] Motik B, Grau BC, Horrocks I, Wu Z, Fokoue A, Lutz C. OWL 2 web ontology language profiles (second edition); 2012. W3C Recommendation. Available from: `https://www.w3.org/TR/owl2-profiles/`.
[49] W3C. Shapes constraint language (SHACL): Technical report; 2017. Available from: `https://www.w3.org/TR/shacl`.
[50] Baader F, Brandt S, Lutz C. Pushing the EL envelope. In: Kaelbling LP, Saffiotti A, editors. IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh,

Scotland, UK, July 30 – August 5, 2005. Professional Book Center; 2005. p. 364-9. Available from: http://ijcai.org/Proceedings/05/Papers/0372.pdf.

[51] Baader F, Lutz C, Brandt S. Pushing the EL envelope further. In: Clark K, Patel-Schneider PF, editors. Proceedings of the Fourth OWLED Workshop on OWL: Experiences and Directions, Washington, DC, USA, 1–2 April 2008. vol. 496 of CEUR Workshop Proceedings. CEUR-WS.org; 2008. Available from: https://ceur-ws.org/Vol-496/owled2008dc_paper_3.pdf.

[52] Corman J, Reutter J, Savkovic O. Semantics and validation of recursive SHACL. In: Vrandecić D, Bontcheva K, Suárez-Figueroa MC, Presutti V, Celino I, Sabou M, et al., editors. The Semantic Web – ISWC 2018 – 17th International Semantic Web Conference, Proceedings, Part I. vol. 11136 of Lecture Notes in Computer Science. Springer; 2018. p. 318-36.

[53] Andresel M, Corman J, Ortiz M, Reutter J, Savkovic O, Šimkus M. Stable model semantics for recursive SHACL. In: Huang Y, King I, Liu T, van Steen M, editors. Proceedings of The Web Conference 2020. WWW'20. New York, NY, USA: ACM / IW3C2; 2020. p. 1570-80.

[54] Sommer A, Car N. pySHACL. Zenodo; 2024.

[55] Syed Z, Padia A, Mathews ML, Finin T, Joshi A. UCO: A Unified Cybersecurity Ontology. In: Proceedings of the AAAI Workshop on Artificial Intelligence for Cyber Security. AAAI Press; 2016. p. 195-202.

[56] Iannacone M, Bohn S, Nakamura G, Gerth J, Huffer K, Bridges R, et al. Developing an Ontology for Cyber Security Knowledge Graphs. In: Proceedings of the 10th Annual Cyber and Information Security Research Conference. CISRC '15. Association for Computing Machinery; 2015. .

[57] Kiesling E, Ekelhart A, Kurniawan K, Ekaputra F. The SEPSES Knowledge Graph: An Integrated Resource for Cybersecurity. In: The Semantic Web – ISWC 2019. vol. 11779 of Lecture Notes in Computer Science. Springer; 2019. p. 198-214.

[58] Hemberg E, Kelly J, Shlapentokh-Rothman M, Reinstadler B, Xu K, Rutar N, et al.. Linking Threat Tactics, Techniques, and Patterns with Defensive Weaknesses, Vulnerabilities and Affected Platform Configurations for Cyber Hunting; 2021.

[59] MITRE. CAPEC: Common Attack Pattern Enumeration and Classification;. Available from: https://capec.mitre.org/.

[60] MITRE. CWE: Common Weakness Enumeration;. Available from: https://cwe.mitre.org/.

[61] National Institute of Standards and Technology. National Vulnerability Database;. Available from: https://nvd.nist.gov/.

[62] de la Vara J, Jiménez G, Mendieta R, Parra E. Assessment of the quality of safety cases: A research preview. In: Knauss E, Goedicke M, editors. Requirements Engineering: Foundation for Software Quality – 25th International Working Conference, REFSQ 2019, Essen, Germany, March 18-21, 2019, Proceedings. vol. 11412 of Lecture Notes in Computer Science. Springer; 2019. p. 124-31.

[63] Foster S, Nemouchi Y, Gleirscher M, Wei R, Kelly T. Integration of formal proof into unified assurance cases with Isabelle/SACM. Formal Aspects Comput. 2021;33(6):855-84.

[64] Diller M, Gaggl S, Gorczyca P. Flexible dispute derivations with forward and backward arguments for assumption-based argumentation. In: Baroni P, Benzmüller C, Wáng YN, editors. Logic and Argumentation – 4th International Conference, CLAR 2021, Hangzhou, China, October 20-22, 2021, Proceedings. vol. 13040 of Lecture Notes in Computer Science. Springer; 2021. p. 147-68.

[65] Diller M, Gorczyca P. ABA Disputes in ASP: Advancing Argument Games Through Multi-shot Solving. In: PRIMA. vol. 16366 of Lecture Notes in Computer Science. Springer; 2025. p. 566-84.

[66] Denecker M, Brewka G, Strass H. A formal theory of justifications. In: Calimeri F, Ianni G, Truszczyński M, editors. Logic Programming and Nonmonotonic Reasoning – 13th International Conference, LPNMR 2015, Lexington, KY, USA, September 27-30, 2015. Proceedings. vol. 9345 of Lecture Notes in Computer Science. Springer; 2015. p. 250-64.

[67] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In: Advances in Neural Information Processing Systems. vol. 33; 2020. p. 9459-74.

[68] Fan W, Ding Y, Ning L, Wang S, Li H, Yin D, et al. A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery; 2024. p. 6491-501.