# The Etiology of Cybersecurity

**Abstract.** The objective of this research is to lay the foundations for the development of a scientific theory that determines (all and only) the possible insecure and secure configurations of any abstract system to be used for the risk assessment of systems. We claim that cybersecurity weaknesses (i.e. errors) are at the beginning of the causality chain that leads to cybersecurity attacks. We formulate an hypothesis that we use to predict the weaknesses in the architectural design of an system. Our hypothesis allows for the definition of a mathematical formula which describes the cybersecurity of a system. We implemented a prototype cybersecurity risk assessment tool that, based on our hypothesis, predicts the weaknesses in a UML model of a (cyber-physical) system.

**Keywords:** Risk Management, Cyber-Physical Systems, Risk Assessment, Security Framework

## 1 Introduction

*Context and motivation.* A *scientific theory* is an explanation of a phenomenon such that the explanation follows the scientific method. The *scientific method* is an *empirical* method that aims at mitigating potential fallacies in theories. Karl Popper famously argued (e.g. in [17]) that a scientific theory can never be verified but only falsified, that a theory should not be conceived by using the principle of induction[1], and that empirical experiments should be considered as the only evidence to support the non-falseness of a scientific theory. In [9], Cormac Herley explores what he calls "an asymmetry in computer security", which he defines as follows: "Things can be declared insecure by observation, but not the reverse. There is no observation that allows us to declare an arbitrary system or technique secure". With security, Herley only focuses on cybersecurity (we also use security and insecurity, in this paper, only to refer to cyber-insecurity and cybersecurity) and his intuition is that there is no scientific theory that can predict the cybersecurity of a system, nor a theory that can predict all possible insecurities of a system (which, by negation, may be used as a theory of cybersecurity). Herley then uses this argument to show that "claims that any measure is necessary for security are empirically unfalsifiable".

*Contributions.* The goal of this paper is to lay the foundations of a scientific cybersecurity theory. We consider the problem raised by Herley not confined to "computer security" but rather we reason on any abstract system (so that our

[1] Einstein to Popper: "[...] and I think (like you, by the way) that theory cannot be fabricated out of the results of observation, but that it can only be invented." [17]

scientific hypothesis[2] may be tested in any sound implementation such as networks, mechanical, cyber, or cyber-physical system, or even a single computer or a single device such as a hard-drive). Instead of starting from reasoning on what makes a system secure or insecure, we reason on what causes insecurities. We focus on insecurities only caused by the exploitation of cybersecurity attacks, and we assume that achieving cybersecurity means preventing all those attacks from being exploitable or exploited. Our hypothesis is that *cybersecurity attacks are only caused by the presence of errors in the design or implementation of a system* (i.e. cybersecurity weaknesses). With our approach, a list of weaknesses emerges from the mathematical formulation of a system in a framework called $\mathcal{ABF}$-framework, and predicts 4 main classes of weaknesses. Those classes of weaknesses are used to calculate all the insecurity configurations of all the components of a system, obtaining a precise estimation of all potential cybersecurity-related risks in any given system. Our hypothesis can be falsified by means of experiments, testing if all the predicted weaknesses are present in the system under test, or testing if other (not predicted by our hypothesis) weaknesses are present. In fact, if any cybersecurity weaknesses were to be found in a system and not predicted by our hypothesis, the hypothesis could be declared incomplete. If a cybersecurity weakness would be predicted by our hypothesis but found to be impossible to realize, our hypothesis could be declared as wrong.

## 2 Literature Review

Cybersecurity attacks seem to be related to the creativity of the attacker and thus unpredictable. Currently, the most complete understanding of insecurity issues is stored into a network of databases of weaknesses (e.g. CWE [4]), vulnerabilities (e.g. CVE [15], NVD [23]), and attacks (e.g. CAPEC [3], ATT&CK [14]). Those insecurity issues can be related to the violation of one or more requirements (explicit or implicit) in the specification, design or implementation of a system. The correlation between insecurity flaws and cybersecurity requirements has been used to define standards such as the IEC 62443-1-3 (the Industrial communication networks - Network and system security – Part 3-3: System security requirements and security levels) which defines requirements as "confidentiality of information in transit/at-rest". More generally, the idea of defining cybersecurity requirements as properties of a system was initially defined in 1970s with the CIA triad (Confidentiality, Integrity, Availability) and refined over the decades introducing related concepts such as authenticity or non-repudiation, or introducing new ones such as "responsibility" in the RITE approach (see [19] for an overview of the evolution of the CIA triad). The link between cybersecurity requirements and vulnerabilities is reported in the NVD databases by the CVSS [13] scoring system. The CVSS evaluates of the severity of a vulnerability by means of different metrics (such as attack complexity

[2] In the reminder of this paper, we will use the word hypothesis to refer to "scientific hypothesis" as a proposed scientific theory that has not gone through an extensive series of tests. We use "logical theory" to refer to a set of formal logical axioms.

and user interaction) and quantitatively evaluates the impact on the CIA triad. While cybersecurity requirements, weaknesses, vulnerabilities, and attacks have been extensively studied and implemented both in academia and industry to provide tools for the testing or verification of systems, no scientific falsifiable theory correlates cybersecurity requirements to necessary and sufficient conditions (e.g. mitigations) to declare a system secure [9]. Nonetheless, the extensive body of literature has scientific foundations, for example, providing formal frameworks for the verification of properties for cybersecurity. As a driver for our argumentation, we start by reviewing the key concepts in the cybersecurity domain.

### 2.1 Terminology

We provide a baseline for a definition of the terms that structure our current understanding of cybersecurity.

*Vulnerability.* As defined in [16] (and adopted in [2]), is a "weakness in an information system, system security procedures, internal controls, or implementation that could be exploited by a threat source".

*Weakness.* The definition given by the MITRE in [5] of weakness is: " a type of mistake that, in proper conditions, could contribute to the introduction of vulnerabilities within that product. This term applies to mistakes regardless of whether they occur in implementation, design, or other phases of a product lifecycle." A vulnerability, such as those enumerated on the Common vulnerabilities and Exposures (CVE) List, is a mistake that can be directly used by an attacker to gain access to a system or network. The definition is circular if we interpret the word "error" and "mistake" with the same semantics: a weakness is an error that leads to a vulnerability and a vulnerability is a mistake which, in turn, is a weakness. The only difference between a weakness and vulnerability seems to be that one can consider weakness as a ground term and state that a vulnerability is caused by a weakness.

*Exploit* "[...] (from the English verb to exploit, meaning to use something to one's own advantage) is a piece of software, a chunk of data, or a sequence of commands that takes advantage of a bug or vulnerability to cause unintended or unanticipated behavior to occur on computer software, hardware, or something electronic (usually computerized)." [10].

*Attack.* As defined by the International Standard ISO/IEC 27000, is an "attempt to destroy, expose, alter, disable, steal or gain unauthorized access to or make unauthorized use of an asset"; where an *Asset* is "anything that has value to the organization". We do not consider ethical hackers as attacking a system. In fact, we consider the term *hack* as non-malicious (as, e.g. in [22]).

*Threat.* As defined in [16], is "Any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, individuals, other organizations, or the Nation through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service".

As in Fig. 1, in order to define a theory on cybersecurity we may say that the presence of vulnerabilities is a necessary condition to cause an attack in the

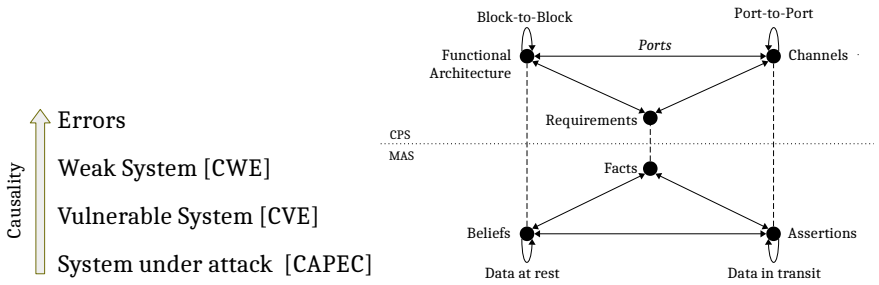


**Fig. 1.** Etiology of cybersecurity (left). Mapping Epistemological Concepts to (Cyber-Physical) Systems Engineering (right).

system. Those vulnerabilities are, in turn, caused by the presence of weaknesses in the system. Weaknesses are errors in the design or implementation of a system and a theory on cybersecurity should first predict the errors in a system design.

## 3 A Cybersecurity Hypothesis in the $\mathcal{ABF}$-framework

To address the problem raised by Herley, we define how to distinguish between a secure and an insecure system in the following steps of the engineering process:

1. *System Specification*: the *functional and physical requirements* are defined.
2. *Architecture Design*: the specification is structured into *functional and physical architectures*.
3. *Cybersecurity Risk Assessment*: potential *weaknesses* (errors) and *cybersecurity requirements* are identified.

We changed the focus from the attacker as the source of insecurity to the potential design errors of a system. We now define a framework for the definition of a system that we use to identify weaknesses as potential design errors.

### 3.1 Mereo-topological Reasoning (2.5)

As in [20], we define agents as a meronomy (an hierarchy of Part-Whole relations) over the constituents that we previously defined (assertions, beliefs, and facts), based on a standard definition of mereology, i.e. based on the definition of parthood relation between *parts*. Due to the necessity of considering different relations between parts (as we'll show afterwards) we extend the mereology to a mereo-topology [18, 21, 25], considering the relations in Tab. 1. For the sake of readability, we use the term *region* both to refer to a mereological part and to a topological region. Our aim is to create a meronomy (hierarchy of part-whole relations) instead of the taxonomies (categorization based on discrete sets) such as the one provided in [15, 23] so that we don't need to rely on a scoring system (such as the CVSS) to assign a quantitative evaluation of the cybersecurity

| RCC3 | RCC5 | Terminology | Notation | Definition |
|---|---|---|---|---|
| | | Connects with | $C(X, Y)$ | reflexive and symmetric |
| | | Disconnected from | $\neg C(X, Y)$ | irreflexive or antisymmetric |
| | | Part of | $P(X, Y)$ | $\forall Z\ C(Z, X) \rightarrow C(Z, Y)$ |
| | | Overlaps | $O(X, Y)$ | $\exists Z\ P(Z, X) \wedge P(Z, Y)$ |
| ● | ● | **Equal to** | $EQ(X, Y)$ | $P(X, Y) \wedge P(Y, X)$ |
| ● | | **Overlaps Not Equal** | $ONE(X, Y)$ | $O(X, Y) \wedge \neg EQ(X, Y)$ |
| ● | ● | **DiscRete from** | $DR(X, Y)$ | $\neg O(X, Y)$ |
| | ● | **Partial-Overlap** | $PO(X, Y)$ | $O(X, Y) \wedge \neg P(X, Y) \wedge \neg P(Y, X)$ |
| | ● | **Proper-Part-of** | $PP(X, Y)$ | $P(X, Y) \wedge \neg P(Y, X)$ |
| | ● | **Proper-Part-of-*inverse*** | $PPi(X, Y)$ | $P(Y, X) \wedge \neg P(X, Y)$ |

**Table 1.** RCC3 and RCC5 relations between regions $X$, $Y$ and $Z$

of each entry. Instead, we want a precise calculation of the number of insecure configurations of a system to emerge from the mathematical formulation of our cybersecurity hypothesis. A mereotopology, as defined in [18], is a mathematical structure where the basic relation between regions is the reflexive and symmetric relation *Connects With*, that we use to order a universe of agents $Ag$ (see in Tab. 1). We use the Region Connection Calculus (RCC), as defined in [7, 11], to provide an axiomatization of the mereo-topological concepts. In its broader definition, the RCC theory is composed by eight axioms, and is known as RCC8 [8]. Using RCC5 instead of RCC8 prevents us from considering tangential connections between spatial regions. However, tangential connections in RCC8 can be considered as special cases of the more general spatial relations considered in RCC5. In Tab. 1, we summarize the axioms of the RCC (see, e.g., [8]). We can now define a system over the mereotopology using the RCC calculus, as follows, where $rcc(X, Y)$ on two generic regions $X, Y$ represents one of the possible RCC relations between $X$ and $Y$. We note that all the RCC relations are symmetric with the exception of those that have an explicit (related) inverse.

**Definition 1. System State** – *A CPS system (or a sub-system) state is defined as the tuple* $s = \langle rcc(\mathcal{F}, \mathcal{B}), rcc(\mathcal{F}, \mathcal{A}), rcc(\mathcal{B}, \mathcal{A})\rangle$*, where* $\mathcal{A}, \mathcal{B}, \mathcal{F}$*, are regions of assertions, beliefs (i.e. the beliefs generated by the behavior), and facts expressed as requirements respectively.*

As in [20], it follows that, defining a system with a fixed number of regions, there exists an upper-bound to the number of possible configuration of a system, defined by the possible relations between the different regions. For completeness, we report in Appendix **??** some details on the calculation done by the authors in [20]. The general formula to calculate the number of different types of agents is $r^{\binom{n}{k}}$, where $r$ is the number of relations with arity $k$, between $n$ different regions. In our case, $\binom{n}{k} = 3$ since we consider 3 regions $(\mathcal{A}, \mathcal{B}, \mathcal{F})$, and all the relations considered in the RCC are binary. Hence, we have up to 125 different type of

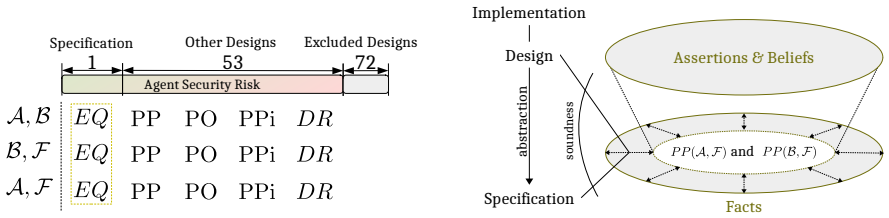


**Fig. 2.** Cybersecurity risk for a single agent (left). Example relation between facts, and assertions and beliefs (right).

| | DR($\mathcal{A}$,$\mathcal{B}$) | PO($\mathcal{A}$,$\mathcal{B}$) | PP($\mathcal{A}$,$\mathcal{B}$) | PPi($\mathcal{A}$,$\mathcal{B}$) | EQ($\mathcal{A}$,$\mathcal{B}$) |
|---|---|---|---|---|---|
| DR($\mathcal{B}$,$\mathcal{F}$) | T($\mathcal{A}$,$\mathcal{F}$) | DR($\mathcal{A}$,$\mathcal{F}$)<br>PO($\mathcal{A}$,$\mathcal{F}$) | DR($\mathcal{A}$,$\mathcal{F}$) | DR($\mathcal{A}$,$\mathcal{F}$)<br>PO($\mathcal{A}$,$\mathcal{F}$) | DR($\mathcal{A}$,$\mathcal{F}$) |
| PO($\mathcal{B}$,$\mathcal{F}$) | DR($\mathcal{A}$,$\mathcal{F}$)<br>PO($\mathcal{A}$,$\mathcal{F}$)<br>PP($\mathcal{A}$,$\mathcal{F}$) | T($\mathcal{A}$,$\mathcal{F}$) | DR($\mathcal{A}$,$\mathcal{F}$)<br>PO($\mathcal{A}$,$\mathcal{F}$)<br>PP($\mathcal{A}$,$\mathcal{F}$) | PO($\mathcal{A}$,$\mathcal{F}$)<br>PPi($\mathcal{A}$,$\mathcal{F}$) | PO($\mathcal{A}$,$\mathcal{F}$) |
| PP($\mathcal{B}$,$\mathcal{F}$) | DR($\mathcal{A}$,$\mathcal{F}$)<br>PO($\mathcal{A}$,$\mathcal{F}$)<br>PP($\mathcal{A}$,$\mathcal{F}$) | PO($\mathcal{A}$,$\mathcal{F}$)<br>PP($\mathcal{A}$,$\mathcal{F}$) | PP($\mathcal{A}$,$\mathcal{F}$) | PO($\mathcal{A}$,$\mathcal{F}$)<br>EQ($\mathcal{A}$,$\mathcal{F}$)<br>PP($\mathcal{A}$,$\mathcal{F}$)<br>PPi($\mathcal{A}$,$\mathcal{F}$) | PP($\mathcal{A}$,$\mathcal{F}$) |
| PPi($\mathcal{B}$,$\mathcal{F}$) | DR($\mathcal{A}$,$\mathcal{F}$) | DR($\mathcal{A}$,$\mathcal{F}$)<br>PO($\mathcal{A}$,$\mathcal{F}$)<br>PPi($\mathcal{A}$,$\mathcal{F}$) | T($\mathcal{A}$,$\mathcal{F}$) | PPi($\mathcal{A}$,$\mathcal{F}$) | PPi($\mathcal{A}$,$\mathcal{F}$) |
| EQ($\mathcal{B}$,$\mathcal{F}$) | DR($\mathcal{A}$,$\mathcal{F}$) | PO($\mathcal{A}$,$\mathcal{F}$) | PP($\mathcal{A}$,$\mathcal{F}$) | PPi($\mathcal{A}$,$\mathcal{F}$) | EQ($\mathcal{A}$,$\mathcal{F}$) |

**Table 2.** RCC5 composition table over 3 regions generates the ideal risk matrix (green the low risk, red the high risk state, and a gradient of medium risk states). T($\mathcal{A}$,$\mathcal{F}$) = {DR($\mathcal{A}$,$\mathcal{F}$), PO($\mathcal{A}$,$\mathcal{F}$), PP($\mathcal{A}$,$\mathcal{F}$), PPi($\mathcal{A}$,$\mathcal{F}$), EQ($\mathcal{A}$,$\mathcal{F}$)}

agents but only 54 of the 125 (as showed in [7]) combinations are topologically correct. Generalizing to all the RCCs:

- *RCC3* — theoretical: $3^3 = 27$, correct: 15.
- *RCC5* — theoretical: $5^3 = 125$, correct: 54.

In the quantitative evaluation of a single agent, as in Fig. 2, we argue that only 1 configuration represents the nominal (expected) behavior of the agent while the other configurations are either impossible to implement or diverge from the intended nominal behavior. We note that the numbers reported here do not consider the details of the engineering process and should be considered a limit of an abstract representation of the system.

### 3.2 Qualitative Evaluation of Agent Space in $\mathcal{A},\mathcal{B},\mathcal{F}$ (3)

While a quantitative analysis reveals how many possible configurations of an agent (i.e. a system) exist w.r.t. the $\mathcal{ABF}$-framework (i.e. 54/125 in RCC5), a qualitative analysis of the different configurations describe the configurations allowed by the $\mathcal{ABF}$-framework, and how those configurations can be categorized.

In Tab. 2, we provide the generic composition table of RCC5 over 3 regions instantiated over $\mathcal{A}, \mathcal{B}, \mathcal{F}$, which shows the whole state space for a single agent. The color coding of the table represents the cybersecurity risk related to a generic agent, the risk is highest on the top left corner of the matrix, lowest on the bottom right corner. In Fig. 2, the relation between facts, and assertions and beliefs (as inputs and outputs of the behavior of an agent) is illustrated. Assertions and beliefs generated by the design of a system may not be exactly aligned with what the facts mandate (i.e. what the specification mandates). The relation between facts, and assertions and beliefs can be used as a metric to determine the soundness of the design with respect to the specification. We first analyze the relations between each pair of regions (i.e. $\mathcal{A}, \mathcal{B}, \mathcal{F}$), and then we categorize the different agents as tuple of the three regions. For the sake of simplicity, soundness is opposed to non-soundness in the following, however, with the RCC one should consider different "degrees" of non-soundness. For example, in RCC5, if we consider EQ between two regions as representing soundness, DR over the same regions represents non-soundness; while PP, PO, PPi represents the different degrees of non-soundness. A similar argument can be done for completeness.

*rcc($\mathcal{A}$,$\mathcal{B}$) – Collaboration.* By definition, assertions are defined as transfer of information between two agents. An agent has two main categories of assertions, input and output assertions. Given an agent $a$ and a collection of asserted predicates $\Phi$, the input assertions are those received by $a$ from an agent $s$ acting as a sender, $\mathcal{A}_{s \to a}\Phi$; similarly, output assertions are sent from $a$ to a receiver $r$, $\mathcal{A}_{a \to r}\Phi$. With a slight abuse of notation, in the text we simply write $\mathcal{A}_{s \to a}$ when we abstract away from the content of the assertion (i.e. $\Phi$). We shall consider two pairs of regions:

- rcc($\mathcal{A}_{s \to a}$,$\mathcal{B}$), where the relation between input-assertions and beliefs describes the soundness of the execution of the functional architecture w.r.t. input elicitation. If all the inputs (assertions) are correctly handled in the functional specification (beliefs) the specification is complete.
- rcc($\mathcal{A}_{a \to r}$,$\mathcal{B}$), where the relation between behavior and outputs describes the completeness of the behavior defined in the specification w.r.t. the input elicitation. If all the outputs (assertions) of the functional architecture can be produced, the functional architecture is complete.

*rcc($\mathcal{A}$,$\mathcal{F}$) and rcc($\mathcal{B}$,$\mathcal{F}$) – Honesty and Competence.* The relation of assertions and beliefs with facts determines the quality with respect to the nominal (specified) system. Given that facts define what needs to be true in the system, the relation of assertions and facts determines the degree of quality between the real information circulating in a system (or within an agent) and the one specified. Since the transfer of information through assertions generates beliefs, a dishonest agent may circulate false information, generating false beliefs. The relation between beliefs and facts determines the competence (on the subjects defined by the facts) of an agent (i.e. the more competent an agent is, the more likely a belief of that agent is true).

**$\mathcal{A}, \mathcal{B}, \mathcal{F}$ CyberSecurity Enumeration (CSE)** The following cybersecurity requirement for a CPS specification can be summarized:

CSE-1 Proper interaction between correctly-behaving agents is defined as $EQ(\mathcal{A}_a, \mathcal{B}_a)$ for an agent $a$, and can be detailed as follows when multiple agents are considered.

CSE-1.1 The equality relation $EQ(\mathcal{A}_{s\to a}, \mathcal{B}_a)$ describes the intended secure behavior as: the beliefs generated by the behavior of the functional architecture shall be complete w.r.t. the specified inputs of the agent. Therefore, *the assertions received by an agent or a system shall be compliant with the expected inputs of the functional architecture.*

CSE-1.2 Similarly, the equality relation $EQ(\mathcal{A}_{a\to r}, \mathcal{B}_a)$ defines that the outputs of an agent $a$ shall be the outputs of the functional architecture.

CSE-2 The proper adherence of the data transmitted between agents with respect to requirements, is defined as $EQ(\mathcal{A}, \mathcal{F})$.

CSE-3 The proper adherence of the behavior (in terms of input and output beliefs) with respect to requirements is defined as $EQ(\mathcal{B}, \mathcal{F})$.

We note that our CSE define the properties of a secure system, and correlated weaknesses can be found in the CWE dataset. For example, CSE-1 can be seen as correlated to the weakness class of "Improper Interaction Between Multiple Correctly-Behaving Entities" defined by the CWE–435, CSE-2 with the "Insufficient Control Flow Management" defined by MITRE in the CWE–691, and CSE-3 with the "Improper Calculation" defined by MITRE in the CWE–682. All those CWE are in the top "view" of the "research concepts" in [4], while the other classes of weaknesses do not have a direct counterpart in our hypothesis; we believe they can be seen as sub-classes, but a full comparison with the CWE is out of the scope of this article. We can now define what a secure system is (with respect to the $\mathcal{ABF}$-framework) and, based on that definition, what the cybersecurity risk is and how to quantify it in a risk matrix. The following definition holds for abstract systems defined in the $\mathcal{ABF}$-framework but will be refined for CPS afterwards in the paper.

**Definition 2. Cybersecurity of a System or an Agent** – *A secure system is a system where CSE-1, CSE-2, and CSE-3 holds for each agent of the system.*

The ISO 31000 consider risk as the "effect of uncertainty on objectives" and refers both to positive and negative consequences of uncertainty. Accordingly, we consider risk as follows.

**Definition 3. Risk** – *The risk is the uncertainty related to the whole space of potential designs resulting from a specification in the $\mathcal{ABF}$-framework.*

The definition of Risk leads to the risk matrix in Tab. 2, defined as follows.

**Definition 4. Risk Matrix** – *The risk matrix, as summarized in Tab. 2, is a function of the three relations $s = \langle rcc(\mathcal{F}, \mathcal{B}), rcc(\mathcal{F}, \mathcal{A}), rcc(\mathcal{B}, \mathcal{A})\rangle$, where the maximum risk is defined by the DR relation between the three groups of regions, and the minimum risk by the EQ relation over the same regions. In between the two extremes, the granularity of possible intermediate configuration is defined by the calculus used (RCC5 in our case).*

While a risk matrix is often defined as a function of the likelihood and impact of attacks (based on quantitative ad-hoc estimation of how likely it is that an attacker will exploit one or more vulnerabilities and what is the magnitude of the incidents produced by this exploitation), we suggest that cybersecurity weaknesses are all equally likely to be exploited if there's a connection between the weakness and the asset that an attacker wants to impact. When the whole system is considered an asset, all weaknesses are equally likely to be exploited. Therefore, a risk matrix should capture the number of insecure configurations of a system, rather than predicating over likelihoods of weaknesses exploitation.

## 4 Prediction of Cybersecurity Weaknesses

Several standards mandate a secure-by-design approach in which cybersecurity shall be considered at the very early stages of the design process. Standards do not describe in detail how to perform a cybersecurity risk assessment and only vaguely define the overall objective, which can be summarized as to provide an understanding of the potential cybersecurity risks. All the methodologies and tools we reviewed (e.g. Threatmodeler [24], CORAS [12], SECRAM [6]) rely on the expertise of the person who performs the risk assessment for the identification of threats and for the quantitative estimation of risks. In contrast, in this section, we define how to specify a CPS in the $\mathcal{ABF}$-framework and we identify a cybersecurity metric.

### 4.1 From Multi-Agent to Cyber-Physical Systems

As in Fig. 1, we relate MAS and CPS as follows.

- We consider a System as a hierarchy of agents. So, we map agents to *systems, sub-systems, or devices*, depending on the granularity of the design. For example, a modeler can model a specific device as a system not decomposed into sub-systems or devices (and the device is considered as an agent).
- Agents reason over beliefs (i.e. transform beliefs into other beliefs) and each *component of a CPS* (system, sub-system, or device) is composed by a *functional architecture* that transforms input-beliefs into output-beliefs.
- Components of a CPS have *ports* to exchange information with the outer environment (which may be a sub-system), similarly to agents. The transfer of information in a CPS is defined by *channels*.
- The concept of facts is related to the *requirements* that describe how the CPS shall behave and its *physical architecture*.

*Input and Output Ports.* Since the $\mathcal{ABF}$-framework is a theory of agents, we could consider ports as agents that allow the exchange of information between a channel and another agent. However, we considered a port as a special type of agent to avoid an *infinite regress*, as described in the following. While a channel transfers information between agents, and a functional architecture processes

information, a port is simply a connector between a channel and a functional architecture. One, however, may argue that a similar connection is needed between a channel and the port itself. While this is not excluded by the $\mathcal{ABF}$-framework, it would obviously lead to an infinite nested structure of ports. To avoid this infinite structure, we assume that a port doesn't require any other means to transfer information from/to a channel or from/to a functional architecture.

**Definition 5. Input or Output Port** – *A port forwards information from the outside of an agent's boundary to the inside (input-port) or vice versa (output-port). There exist two types of ports with the following behavior: an input-port transforms assertions from a sender s ($\mathcal{A}_{s\to a}$) to beliefs of an agent a ($\mathcal{B}_a$), while an output-port transforms beliefs into assertions.*

The *quality of a port* is determined by the *rcc* relation between the assertions received or sent and the belief, i.e. $rcc(\mathcal{A}_{s\to a}, \mathcal{B}_a)$ for input-port or $rcc(\mathcal{A}_{r\to *}, \mathcal{B}_a)$, for output-port (where $*$ is a wildcard for an agent). A port is, in fact, syntactic sugar to express the relation between assertions and beliefs. We note that the definition of an input/output port can be considered "secure", meaning that we implicitly formulated the requirement that a port always forwards the information without modifying it in any way. This is assumed since we defined a port as if the RCC equality relation holds between the input/output assertions and the input/output beliefs. In contrast, assuming any *other* RCC relation between inputs and outputs of a port can be considered as generating a weaknesses.

*Port Weaknesses.* We now apply the $\mathcal{ABF}$-framework to list all possible cybersecurity weaknesses of a port. From our definition of ports, it follows that

**Theorem 1.** *Port Weaknesses. There exist only the following* six *types of weaknesses, generating six types of insecure port in RCC5:*

*W1)* Replace port*, where assertions reach the port but are replaced with different and un-related information before passing the boundary.*
*W2)* Drop port*, where assertions reach the port but do not pass the boundary of the agent (i.e. do not become belief of the agent).*
*W3)* Insertion port*, where new information is transferred along with the information incoming from a channel, and then sent to the recipient (agent).*
*W4)* Injection port*, where part of the incoming information is substituted with new information and transferred to the intended recipient.*
*W5)* Selective port*, where some information passes the port and part is either: (W5.1) Dropped or (WP5.2) Replaced.*

*Proof.* An input port is, in the $\mathcal{ABF}$-framework, defined secure as long as the relation between the two regions of input assertions $\mathcal{A}$ and output beliefs $\mathcal{B}$ are equal, i.e. $EQ(\mathcal{A}, \mathcal{B})$. Therefore, any other relation should result in a weakness (related to an insecurity flaw) of that input port. Using RCC5, there exist exactly other 4 different types of relations, one of which is the discrete-from (DR) relation, i.e. $DR(\mathcal{A}, \mathcal{B})$. When two regions are related by the DR relations, they have

no subregion in common. Let us define a function weight $|X|$ such that, for any region $X$, it represents the smallest possible cardinality of a (mereo)topological base for $X$; where a base is a collection of regions in a (mereo)topology such that every region can be written as union of elements of that base. We distinguish between regions that are related to information and regions that don't by writing the latter as $\emptyset$ (e.g. a region corresponding to an assertion is written $A \neq \emptyset$, while a region corresponding to an empty assertion as $A = \emptyset$).

1. If $EQ(\mathcal{A}, \mathcal{B})$ then either $\mathcal{A} = \mathcal{B} = \emptyset$ (no communication) or $\mathcal{A} = \mathcal{B} \neq \emptyset$ (forward communication).
2. If $DR(\mathcal{A}, \mathcal{B})$ then $\mathcal{A} = \emptyset \oplus \mathcal{B} = \emptyset$ (we call *insert* the former, *full drop* the latter case), or $\mathcal{A} \neq \emptyset \wedge \mathcal{B} \neq \emptyset \wedge \mathcal{A} \neq \mathcal{B}$ called *replace* (i.e. drop and insert).
3. If $PP(\mathcal{A}, \mathcal{B})$ then $\mathcal{B}$ contains and extend $\mathcal{A}$ which we call *insertion*.
4. If $PPi(\mathcal{A}, \mathcal{B})$ then $\mathcal{A}$ contains and extend $\mathcal{B}$ which we call *drop* (or selective drop to stress the difference with the full drop).
5. If $PO(\mathcal{A}, \mathcal{B})$ then a part of the $\mathcal{A}$ is contained in the $\mathcal{B}$ which is a combination of *selective drop and generation* which we call *injection*.

*Communication Channels.* In this work, we only consider mono-directional channels and communication but the extension to bi-directional channel can be considered as the union of two unidirectional channels. A mono-directional channel is defined by the assertions sent or received (over the channel). We consider first the difference between a (communication) mono-directional channel (channel from now on) and an agent, as we did for the ports, since the $\mathcal{ABF}$-framework is a logical theory of agents. In fact, if a channel were considered an agent (channel-agent) then the question would be how an agent would transfer its assertions to the channel-agent. If the channel between the agent and the channel-agent is again an agent, we would generate an *infinite regress*. Therefore, we do allow channel-agents but we assume a finite depth (of detail) for a channel, where there exists a bottom-channel which is not an agent. For now, we do not constrain a channel-agent in any way so there is no difference between a channel agent and agent. Therefore, we consider channels to be bottom-channels, defined as agents with the pre-defined behavior (i.e. defined in an axiomatic way) of forwarding any input-assertion as output-assertion, without modifying it[3].

**Definition 6. Mono-directional Channel (bottom-channel)** – *A mono-directional channel between two agents ($s \rightarrow r$) is defined as an agent whose behavior is (dogmatically) defined as: to forward any assertion received from $s$ over an input-port, to the output-port where $r$ is listening to.*

The *quality of a mono-directional* channel is defined as the *rcc* relation between the assertions of the sender and the ones received by the receiver, i.e. $rcc(\mathcal{A}_s, \mathcal{A}_r)$.

[3] Nothing prevents us from introducing additional constraints to the channel as storing assertions that are transferred over the channel, or filter out some input-assertions.

*Channel Weaknesses.* Given that a mono-directional bottom-channel is assumed to be perfectly forwarding any assertion (as we assumed for ports) from its input-port to its output-port, there is no insecure behavior but only the combination of the weaknesses of the input and output port; therefore there exist $(7^2) - 1 = 48$ theoretical configurations ($7^2$ because there are 6 insecure types of port, plus 1 secure type, on both input and output side; and we exclude the configuration with 2 secure types as input and output, $-1$); where only 44 are possible. For the sake of readability, we report 6 examples in the following but the proof by exhaustion over all the possible cases is straightforward.

W6) *Secure output port* and *input drop port.*
W7) *Secure output port* and *input insertion port.*
W8) *Output drop port* and *input drop port.*
W9) *Output drop port* and *input insertion port.*
W10) *Output drop port* and *input secure port.*
W11) *Output injection port* and *input secure port.*

*Cybersecurity Weaknesses – The RIDI-Hypothesis.* All the results of the application of the $\mathcal{ABF}$-framework to channels (the analysis of the RCC relations between output and input assertions of an agent) lead to the same results of the analysis of a pair of an input and output port. So far we have considered information generated by a port $P_I$ and then sent through a channel $C$ to another (recipient) port $P_O$. In this scenario, where ports and channels are atomic (otherwise raising infinite regress), we can only consider the relations between ports and channel; considering both input-port to channel and channel to output-port. In fact, the weaknesses of a channel are defined in terms of the weaknesses of ports. The same result can be obtained by analyzing the relation between the outputs of a functional block and the inputs of another functional block, where functional blocks are constituents of the functional architecture as described afterwards. To define a functional block without encountering an infinitely recursive definition, we must reach the same conclusions as for the channel. So, describing the information as flowing over a channel or in a functional block is purely syntactic sugar. We can summarize these results by saying that the relations between assertions and beliefs, output assertions of an agent and input assertions of another agent, or output beliefs of a functional block and input beliefs of another block can *only* be affected by the following weaknesses: replace, drop, injection, insertion, selective drop, and selective drop + insertion. We call this the RIDI-Hypothesis, being the four main categories of weaknesses: Replace, Insertion, Drop, Injection. We can, then, deduce the following cybersecurity properties to mitigate cybersecurity weaknesses of a port or a channel (between ports, functional blocks, or both).

- *Order-preserving* – it shall be known if information is *replaced.*
- *Availability* – it shall be known if information is *dropped* or *selectively dropped.*
- *Integrity* – it shall be known if information is *injected.*
- *Authentication* – it shall be known if information is *inserted.*

*Functional Architecture.* A functional architecture takes information as input-beliefs and transforms the information into output-beliefs. Those transformations occur within the functional architecture, where functional blocks transform beliefs into other beliefs. Similarly to channels, we could consider a functional block as a functional architecture occurring in an infinite regress. Therefore, we consider functional blocks as executing an abstract undefined behavior, of which we only observe the inputs and the resulting outputs (beliefs).

**Definition 7. Functional Block and Architecture** – *A functional block of an agent takes beliefs as inputs (input-beliefs) and returns output-beliefs. A functional architecture is an interconnected system of functional blocks.*

The quality of a functional block cannot be determined by the difference between its inputs and outputs (as we did for ports and channels), because the behavior of a functional block cannot be determined in general; since any functional block will have its own purpose based on functional requirements. Therefore, while the semantics of a port is determined by the relation between assertions and beliefs, the semantics of a functional block is determined by the relation between facts/requirements and I/O beliefs. In other words, a functional block is a generic agent with no pre-defined general behavior (while ports and channels have a pre-defined behavior). In the following, for the sake of simplicity, we use the generic region $\mathcal{B}$ to refer to the behavior (i.e. the beliefs generated by the behavior).

W50) $PO(\mathcal{B}, \mathcal{F})$ the component has a Byzantine behavior. Not all the inputs are handled properly, nor all the expected outputs are always generated when correct inputs are given.
W51) $PP(\mathcal{B}, \mathcal{F})$ some expected outputs are not generated with the correct inputs
W52) $PPi(\mathcal{B}, \mathcal{F})$ the components correctly perform the expected behavior when the correct inputs are provided but is subject to input injections
W53) $DR(\mathcal{B}, \mathcal{F})$ the component never performs the expected behavior (e.g. physical damage)

*Requirements as Facts.* During the specification phase, for any agent, channel, port, functional block and architecture, there may exist a requirement (fact) predicating over them. In other words, any requirement is defined as a fact since they must be true in any design or implementation. As in Section 4 and depicted in Fig. 2, facts are definitory rules that define how the system shall behave (by specification), while reality may be shown to be insecure (i.e. diverging from the expected behavior). As an example, considering a functional block that performs the summation of two inputs defined by the requirement $r := b_3 = b_1 + b_2$ for any $b_1$ and $b_2$. The possible relations between the beliefs generated by the behavior of the functional block and the requirements (i.e. $rcc(\mathcal{B}, \mathcal{F})$ is determined by the relations between the I/O beliefs $sum(b_1, b_2) = b_3$ and the requirement $sum(b_1, b_2) = b_1 + b_2$, as follows.

- $EQ(\mathcal{B}, \mathcal{F}) = EQ(\mathcal{B}_3, \mathcal{B}_{1+2})$, where $\mathcal{B}_3$ represents the region of the outputs of *sum* while $\mathcal{B}_{1+2}$ the expected outputs of an ideal implementation of the requirement $r$. The functional block correctly implements the requirements.

| RCC5 | Quantity: Data Flow | Quality: Requirements Adherence |
|---|---|---|
| EQ | Expected/Nominal | Expected/Nominal |
| DR | Drops all inputs and inserts new data | The component never performs/carries the expected behavior/information |
| PP | Selectively drops inputs | Part of the expected outputs are not generated in response to the correct inputs |
| PPi | Forwards all the inputs but crafts and inserts new malicious data | The components correctly performs/carries the expected behavior/information when the correct inputs are provided but is subject to input injections |
| PO | Selectively drops inputs and inserts new data | Byzantine behavior. Occasionally outputs the expected output given the correct inputs. Not all inputs are handled properly, nor all expected outputs always generated on correct inputs |

**Table 3.** Weaknesses Categorization

- $DR(\mathcal{B}, \mathcal{F}) = DR(\mathcal{B}_3, \mathcal{B}_{1+2})$, the functional block do not implement the requirements.
- $PP(\mathcal{B}, \mathcal{F}) = PP(\mathcal{B}_3, \mathcal{B}_{1+2})$, some inputs produce incorrect output.
- $PPi(\mathcal{B}, \mathcal{F}) = PPi(\mathcal{B}_3, \mathcal{B}_{1+2})$, some outputs are not the result of the summation of the inputs but with the correct inputs the function outputs correctly.
- $PO(\mathcal{B}, \mathcal{F}) = PO(\mathcal{B}_3, \mathcal{B}_{1+2})$, Byzantine behavior where occasionally outputs are produced with the correct inputs. Not all the inputs are handled properly, nor all the expected outputs are generated when correct inputs are given.

*Assertions and Facts.* The whole reasoning on the relation between beliefs and facts can be duplicated for the relation between assertions and facts; we cannot appreciate the difference at this level of abstraction. If the functional architecture would be extended to capture the semantics (i.e. the logic) of the communication and cybersecurity protocols, with the relation between assertions and facts we would compare protocol logics with the requirements. We won't consider the verification of the functional architecture and protocol logic in this paper since we focus on the architecture specification step of the engineering process without going into the design of the behavior of agents. We summarize our results by categorizing the weaknesses predicted by our hypothesis into: data-flow-related and functionality-related weaknesses; as in Tab. 3. Functional weaknesses can be seen as a general formulation of our hypothesis, while data-flow weaknesses as an application of our hypothesis to components with defined behavior/requirements.

### 4.2 Security and Insecurity of a System

**Hypothesis 1 System Security Design** – *A system security design (in the $\mathcal{ABF}$-framework) is given by a precise system specification over the physical and functional architectures that uniquely defines the design built on top of those requirements.*

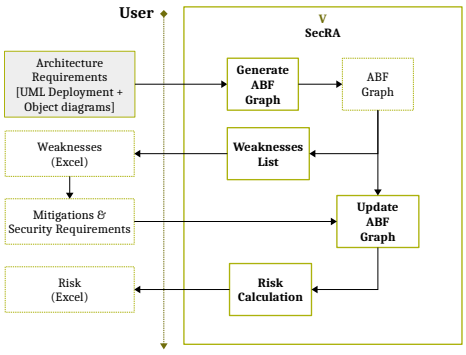


**Fig. 3.** Cybersecurity Risk Assessment Tool

**Hypothesis 2 System Insecurity Design** – *If, given a system specification as a collection of requirements, there exist a non-unique design with respect to those requirements, the number of equivalent designs (that fulfills the requirements) quantitatively defines the magnitude of insecurity of a system design.*

Based on these hypothesis, we can formulate the concepts of security and insecurity (in the $\mathcal{ABF}$-framework) as mathematical equations. Let us consider a CPS $S$, represented as a graph $G = \langle V, E \rangle$ where $V$ represents the set of functional blocks and ports of $S$, and $E \subseteq V \times V$ is the set of pairs representing the channels and connections (data flows) between functional blocks. We define $R \subseteq V \times F$, where $F$ is the set of all the requirements of $S$, and extend $G$ as $G' = \langle V', E' \rangle$ with $E' = E \cup R$ and $V' = V \cup F$. Let $\pi : E' \rightarrow RCC$ (where $RCC$ is the set of relations in the RCC) be the total function associating an RCC relation to each edge in $G'$, and $\Pi$ be the set of all different permutations of RCC relations over $E'$ (i.e. $\Pi = \{\langle \pi(e_0) = EQ, \ldots, \pi(e_n) = EQ \rangle, \ldots, \langle \pi(e_0) = DR, \ldots, \pi(e_n) = DR \rangle\}$ where $e_i \in E'$ for $0 \leq i \leq n$ and $|E'| = n$). If $\sigma : \Pi \rightarrow \{0, 1\}$ is an evaluation function such that $\sigma(p) = 1$ (where $p \in \Pi$) iff the input configuration is satisfiable with respect to the logical theory defining the algebraic structure (mereotopology) and constraints of the calculus RCC (otherwise $\sigma$ returns 0), we can define

$$I = \sum_{p \in (\Pi \setminus \pi_{eq})} \sigma(p)$$

where $\pi_{eq} \in \Pi$ is the output of the function $\pi$ that associates only EQ relations to any $e \in E'$, and $I \in \mathbb{N}$ represents all the insecurity configurations of the CPS $S$ where, at least, one of the RCC relations isn't EQ. In other words, we consider $\pi_{eq}$ as the only secure configuration.

### 4.3 Cybersecurity Risk Assessment

To test our hypothesis we implemented a tool-chain (open-source with AGPLv3 license, available at [1]) for the identification of weaknesses and the calculation

of potential insecure configurations. The engineering of the $\mathcal{ABF}$-framework for CPS is summarized in the UML Class diagram in Fig. 5 (in appendix). As in Fig. 3, the cybersecurity risk assessment process starts with the definition of the use cases and architectural requirements. In our process, the specification is manually translated into a UML design where:

- a *deployment diagram* describes the *physical architecture*. Each agent is defined as an UML node with (physical) ports, and agent' ports are connected via UML information flow connectors, representing the physical channel.
- a *functional architecture* is linked to each agent in the deployment diagram and is defined by an *object diagram*. The object diagram is composed by instances of functional blocks, connected via information flow connectors.
- the connection between the two diagrams is implemented by "sockets", functional blocks connected to a physical port.

The tool generates a graph-like structure which represents the specification ($\mathcal{ABF}$-graph). The $\mathcal{ABF}$-graph defines the system as a number of regions of assertions, beliefs, and facts. Those regions are connected by a generic relation which is evaluated as follows (according to the formula in Section 4.2). The graph is translated into a logical formula that represents the specification in the $\mathcal{ABF}$-framework and, along with the axiomatization of the RCC5 calculus, is given as input to the Z3 SMT solver. The solver identifies all possible configurations of the system and, in turn, identifies all potential weaknesses. The $\mathcal{ABF}$-graph can be viewed as PDF and the results are reported into an spreadsheet file. The spreadsheet file also reports the total number of configurations as indicating the cybersecurity risk associated to the specification. A user can change the status of each weakness in the spreadsheet file from the default status (open) to "mitigated" and the risk is re-calculated on-the-fly, i.e. without the need of running the tool again, based on annotations and formulas in the spreadsheet file. In our approach, cybersecurity requirements are not imposed by the specification but are automatically extracted by our tool as mitigations to potential weaknesses, which are related to the insecure configurations of the specified system.

*Case Study.* We report here the results of the evaluation of a water level reader (sensor ad-hoc example). As in Fig. 4 (left), we defined 2 agents: sensorInTank and sensorBoard, as the physical reader that needs to be placed in a tank, and the board that interprets the readings and outputs them as signals. The two components are connected by a wire. In Fig. 4 (right), we report the functional architecture that receives the incoming communications from the sensor in the tank and communicates them encrypted. The tool (App. C) reports 16777216 scenarios in which at least one component diverge from the specification.

## 5 Conclusion and Future Works

We proposed an hypothesis for a foundational theory on security, arguing that cybersecurity-related issues are not linked to the maliciousness of an agent but

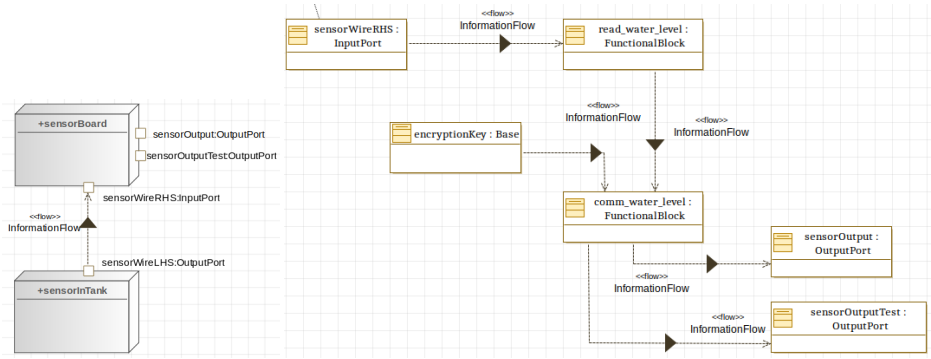


**Fig. 4.** Sensor board object (left) and water level reader deployment (right) diagrams.

to the vagueness in the design processes. We provided a prototype tool for the quantitative estimation of the cybersecurity risk based on a UML model of a system. The verification and test-case generation will be our next steps.

## References


[1] 42. *Repository*. URL: `https://github.com/....`

[2] Rebecca M. Blank and Patrick D. Gallagher. "NIST Special Publication 800-53 Revision 4 - Security and Privacy Controls for Federal Information Systems and Organizations". In: *National Institute of Standards and Technology Special Publication* (Apr. 2013).

[3] *Common Attack Pattern Enumeration and Classification*. URL: `https://capec.mitre.org/`.

[4] *CWE VIEW: Research Concepts*. URL: `https://cwe.mitre.org/data/definitions/1000.html`.

[5] *FAQ – What is the difference between a software vulnerability and software weakness?* URL: `https://cwe.mitre.org/about/faq.html#A.2`.

[6] Martina de Gramatica et al. "The role of catalogues of threats and security controls in security risk assessment: an empirical study with ATM professionals". In: *International Working Conference on Requirements Engineering: Foundation for Software Quality*. Springer. 2015, pp. 98–114.

[7] Rolf Grütter, Thomas Scharrenbach, and Bettina Bauer-Messmer. "Improving an RCC-Derived Geospatial Approximation by OWL Axioms". In: *ISWC*. 2008.

[8] Rolf Grütter, Thomas Scharrenbach, and Bettina Bauer-Messmer. "Improving an RCC-Derived Geospatial Approximation by OWL Axioms". In: *The Semantic Web (ISWC)*. Ed. by Amit Sheth et al. Springer Berlin Heidelberg, 2008, pp. 293–306.

[9] Cormac Herley. "Unfalsifiability of security claims". In: *Proceedings of the National Academy of Sciences (PNAS)* 113.23 (2016), pp. 6415–6420.

[10] Wikipedia Foundation Inc. *Exploit (computer security)*. Mar. 18, 2022. URL: https://en.wikipedia.org/wiki/Exploit_(computer_security).
[11] Tsau Young Lin, Qing Liu, and Y. Y. Yao. "Logics Systems for Approximate Reasoning: Approximation via Rough Sets and Topological Spaces". In: *ISMIS*. 1994.
[12] Mass Soldal Lund, Bjørnar Solhaug, and Ketil Stølen. *Model-driven risk analysis: the CORAS approach*. Springer Science & Business Media, 2010.
[13] Peter Mell, Karen Scarfone, and Sasha Romanosky. "A complete guide to the common vulnerability scoring system version 2.0". In: *Published by FIRST-forum of incident response and security teams*. Vol. 1. 2007, p. 23.
[14] MITRE. *ATT&CK*. URL: https://attack.mitre.org/.
[15] MITRE. *Common Vulnerabilities and Exposures (CVE)*. URL: https://cve.mitre.org/ (visited on 03/18/2022).
[16] Committee on National Security Systems (CNSS). "Glossary No 4009". In: *National Information Assurance (IA) Glossary* (Apr. 6, 2015). URL: https://rmf.org/wp-content/uploads/2017/10/CNSSI-4009.pdf.
[17] R. Karl Popper. *The Logic of Scientific Discovery*. New York London, 1959.
[18] Saikeerthi Rachavelpula. "The Category of Mereotopology and Its Ontological Consequences". In: *University of Chicago Mathematics Research Program*. Ed. by Michael Neaton and Peter Peter. 2017.
[19] Spyridon Samonas and David Coss. "the CIA Strikes Back: Redefining Confidentiality, Integrity and Availability in Security". In: *Journal of Information System Security* 10.3 (2014).
[20] Katia Santacà et al. "A Topological Categorization of Agents for the Definition of Attack States in Multi-agent Systems". In: *Proceedings of the European Conference on Multi-Agent Systems and Agreement Technologies (EUMAS)*. 2016, pp. 261–276.
[21] Barry Smith. "Mereotopology: A theory of parts and boundaries". In: *Data & Knowledge Engineering* 20.3 (1996). Modeling Parts and Wholes, pp. 287–303.
[22] Richard Stallman. *The Hacker Community and Ethics: An Interview with Richard M. Stallman*. 2002. URL: https://www.gnu.org/philosophy/rms-hack.html.
[23] National Institute of Standards and Technologies (NIST). *National Vulnerability Database*. URL: https://nvd.nist.gov/.
[24] Threatmodeler. *ThreatModeler*. URL: https://threatmodeler.com/.
[25] Achille C. Varzi. "On the Boundary Between Mereology and Topology". In: *Proceedings of the International Wittgenstein Symposium*. 1994, pp. 261–276.

## A Class Diagram for $\mathcal{ABF}$-framework

The Class Diagram for the Engineering of the $\mathcal{ABF}$-framework is reported in Fig. 5. A *specification* of a CPS is viewed as an aggregation of *architectures* which can describe the functional or physical requirements. The physical com-

ponents of the architecture are input/output *ports* and *channels* (aggregations of pairs of ports) while *functional blocks* are the only constituents of the functional architecture. All of the classes are abstract except input/output ports and functional blocks. Therefore, agents (which represents sub-systems or components) are composed by ports and functional blocks, as an aggregation of architectures.

## B Number of different configurations of a system.

The general formula to calculate the number of different types of agents is $r^{\binom{n}{k}}$, where $r$ is the number of relations with arity $k$, between $n$ different regions. Therefore, $r^{\binom{n}{k}}$ expresses the number of permutation of $r$ relations over $\binom{n}{k}$ elements with repetitions, with $\binom{n}{k}$ being the number of $k$-ary combinations of $n$ regions. In our case, $\binom{n}{k} = 3$ since we consider 3 regions $(\mathcal{A}, \mathcal{B}, \mathcal{F})$, and all the relations considered in the RCC are binary. Hence, using RCC5 (with five different spatial relations) over three regions, we can theoretically define up to 125 different type of agents. However, only 54 of the 125 (as showed in [7]) combinations are topologically correct with respect to the definition of the relations of RCC5. Generalizing to all the RCCs:

- *RCC3* — theoretical: $3^3 = 27$, correct: 15.
- *RCC5* — theoretical: $5^3 = 125$, correct: 54.

## C Overview of the Results of the Tool

In Fig. 6 we show a screenshot of the results reported by our tool.

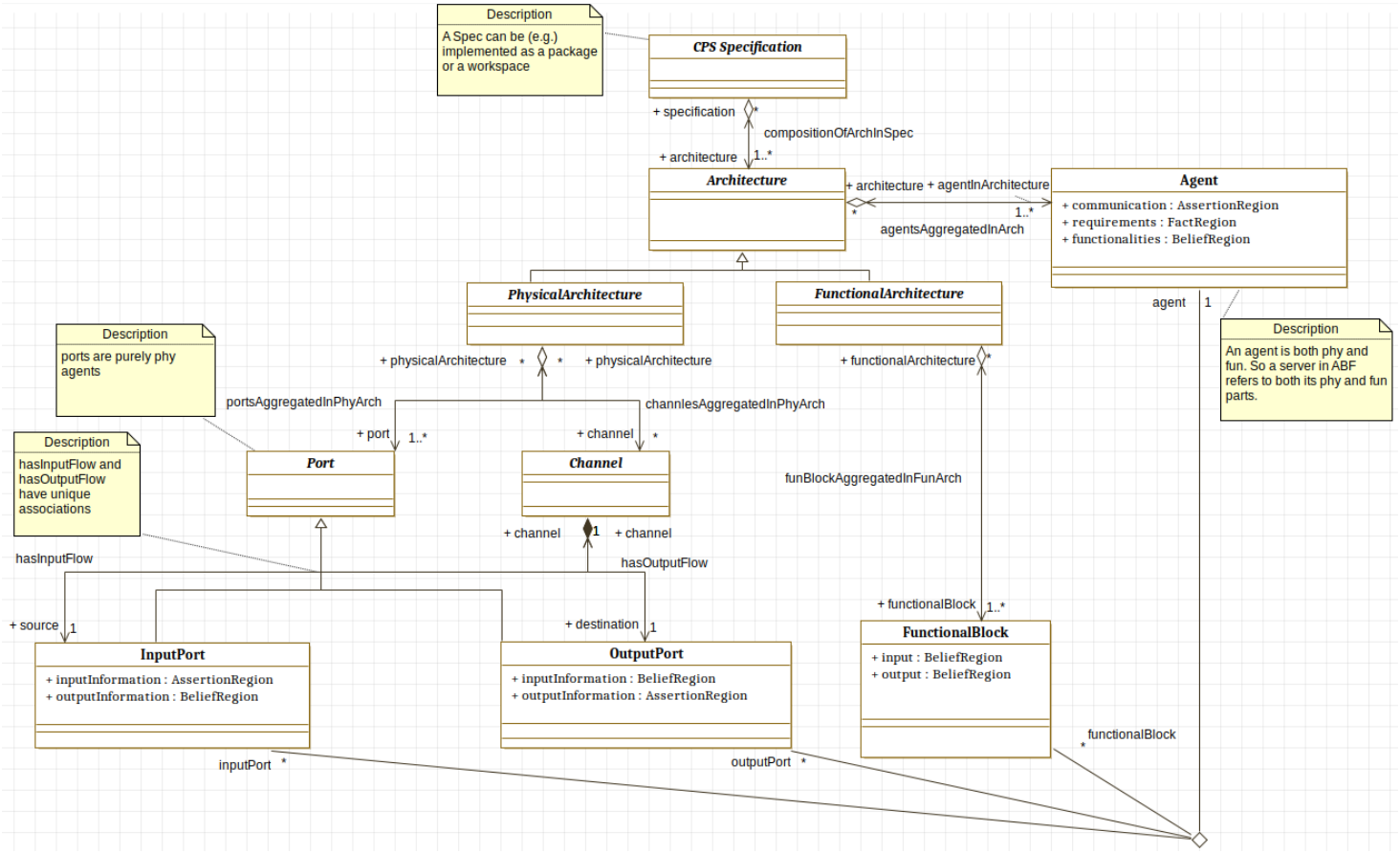


**Fig. 5.** $\mathcal{ABF}$-framework for CPS Design – Class Diagram

| RISK | | 16777216 |
|---|---|---|
| The total risk is the total number of insecure configurations of the system | | |

| B | C | D | E | F |
|---|---|---|---|---|
| **Agent** | **Component** | **Comp. Type** | **Weakness** | **Status** |
| sensorBoard | sensorWireRHS | inputport | selectively drops inputs and inserts new malicious data | open |
| root | sensorWireLHS2sensorWireRHS | channel | selectively drops inputs and inserts new malicious data | open |
| sensorInTank | sensorWireLHS | outputport | selectively drops inputs and inserts new malicious data | open |
| sensorInTank | sensorWireLHS | outputsocket | the component has a Byzantine behavior where occasionally outputs the expected output given the correct inputs. Not all the inputs are handled properly, nor all the expected outputs are always generated when correct inputs are given. | open |

**Fig. 6.** Partial View of the results in the spreadsheet file