The Etiology of Cybersecurity

Marco Rocchetto
V-Research
Verona, Italy
Email: marco@v-research.it

Francesco Beltramini
V-Research
Verona, Italy
Email: francesco@v-research.it

***Abstract*—The objective of this research is the development of a theory that determines (all and only) the possible insecurity and security configurations of any abstract system. Our hypothesis is that errors are at the bottom of the causality chain that leads to cybersecurity attacks. Based on this hypothesis, we formulate a scientific (falsifiable by experiments) theory that predicts all the errors that lead to the presence of cybersecurity weaknesses in the architectural design of an abstract system. Our theory is based on a correlation between the epistemological concepts of knowledge, beliefs, and information with the engineering concepts of requirements, functional architectures, and channels. Ultimately, our theory allows the definition of cybersecurity of a system as a mathematical formula. We implemented a prototype cybersecurity risk assessment tool that, based on our theory, predicts weaknesses in a UML model of a (cyber-physical) system.**

I. Introduction

a) 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 [34]) 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.

If we consider a natural (real-world) phenomenon, such as gravity, we can think of a scientific theory as an *abstract* mathematical formula (or a set of formulas) that defines axiomatically a certain (natural) phenomenon in a specific algebraic structure. While such a theory can be proved to be correct within the mathematical setting it has been conceived in (e.g. with respect to other axioms of the algebraic structure where it has been postulated, or with respect to other background theories), no proof can be carried out to verify that the theory properly abstracts the phenomenon it describes (i.e. that the theory is a sound or complete approximation of the real phenomenon). Similarly, the results of the application of a scientific theory (i.e. its predictions) can be verified in the abstract mathematical structure but how can we be sure that the theory correctly characterizes the phenomenon it describes? Popper's famous "demarcation principle" draws a line between scientific and pseudo-scientific theories by proposing that *any* scientific theory can be considered correct (or, better, acceptable) as long as no empirical experiment falsify it. For example, the theory of gravity, in its mathematical setting, can predict how objects attract each other, and those predictions can be tested by empirical experiment. If the theory correctly abstracts the phenomenon, no empirical experiment should falsify the predictions of that theory.

In [20], 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 (as in the reminder of this paper) only focuses on cybersecurity and his intuition is that there is no scientific theory that can predict the security 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 security). Herley then uses this argument to show that "claims that any measure is necessary for security are empirically unfalsifiable". An example is whether or not password masking makes a system secure. Password masking prevents a complete leakage of the password and a system in which password masking is implemented is believed to be secure because an attacker won't be seeing the password by "shoulder surging". However, this reasoning is circular and it is always true that a system in which password masking is implemented is a system in which a password is masked. No experiment can ever falsify this statement. Given that any theory that is not falsifiable by an empirical experiment is well known[2] to be nonscientific (i.e. unfalsifiability is a fallacy of a theory), Herley concludes that there is no scientific theory of cybersecurity; which means that cybersecurity lays in the realm of pseudo-sciences [19]. Herley, e.g. in [18], discusses the implications of a nonscientific approach to cybersecurity, and highlights the tremendous impact on all the scientific research and engineering of systems. While the criticism is investigated in [20], no solution is provided nor envisioned.

b) 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 theory may hold for any sound implementation such

[1] Albert Einstein wrote 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." [34]

[2] "A theory which is not refutable by any conceivable event is nonscientific. Irrefutability is not a virtue of a theory (as people often think) but a vice." – Karl Popper, Conjectures and Refutations [33].

as networks, mechanical, cyber, or cyber-physical system, or even a single computer or a single device such as a hard-drive). There is also an apparent inconsistency in [20] that we seek to clarify before following (as we agree) the scientific path drawn by Herley: cybersecurity is defined as a set of abstract properties in many formal approaches to the investigation of the security of systems. Let's, for the sake of simplicity, consider the case of a system implementing a security protocol (e.g. an authentication protocol). The proof of security obtained from the formal verification of the design of the protocol (e.g. with respect to a formalization of the authentication property) is indeed falsifiable against the security properties verified. For example, in the protocol verification community, cybersecurity is often defined as a formalization of the high-level properties confidentiality, integrity, and authentication. The problem in such approaches is not the definition of what cybersecurity is but the generality of the results, since they tackle a specific step[3] of the engineering process (of systems) without considering many aspects related to the implementation of a protocol that may generate security vulnerabilities later in the development process. An infamous example is the KRAK attack on the WPA2 4-way handshake protocol where the author wrote [54]: "our attacks do not violate the security properties proven in formal analysis of the 4-way handshake" but "the problem is that the proofs do not model key installation." The theories underlying the verification of security protocols are based on assumptions which non-evidently apply to a general security theory; otherwise the assumptions would clearly state the necessary element that needs to be considered (e.g. modeled) in order to obtain a secure system. As an example, the so called Dolev-Yao attacker model[4] [11] only applies to specific instances (often called scenarios) and abstraction of the protocol. This, in turn, creates a false sense of security since it requires non-justifiable assumptions on the abstraction of the system under verification. More importantly, some of the choices made by the modelers/engineers using those formal approaches may completely change the results of the formal verification of the system (an interesting example on how difficult it is to even just compare the approaches is given in [8]). As an illustrative example, in Figure 1, under the perfect cryptography assumption (i.e. assuming that the cryptographic primitives have no security flaw), and assuming that no violation to any security property occurs after message I), the freedom of choosing the scope determines that the flaws related to the dishonest impersonification of the Server may or may not be considered in the verification process. This choice has a tremendous impact on the focus and findings of the verification of the security of the protocol. While this may seem to turn upon minutiae and foreseeable, this highlights the false sense of security that may derive from a non-falsifiable theory of system security. **tofix:** In other words, if (as Herley mentions in [**herley**]) the problem is in the way we reason on cybersecurity, we should see it in the way we reply to simple cybersecurity (foundamental) questions. Let us consider the foundamental question "what is cybersecurity?", as in [**hintikka**], we can translate the wh-question into an equivalent existentially quantified formula. But if the formula is the following second-order formula: $\exists P.P(s) \rightarrow Secure(s)$, (where we ask if there exists a predicate P such that, given any arbitrary system s, when P holds for s then s can be considered secure), when the answer is (for example) "if the system is confidential, the system is secure" the formula becomes tautological. In fact, P and $Secure$ can be considered as semantically equivalent (security is confidentiality and vice-versa). Instead of starting from reasoning on what makes a system secure or insecure, we reason on what causes insecurities. **tofix:** We focus on insecurities only caused by the exploitation of cybersecurity attacks, and we assume that achieving security 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. For example, a design or implementation error in a sanitization function may allow an attacker to perform SQL-injections. If errors are necessary to enable cybersecurity attacks, a theory that predicts all potential errors (or source of errors) can be effectively used to predict the security of a system. **tofix:** Therefore, a possible answer to the initial question can be that cybersecurity is the absence of attacks and the absence of errors implies cybersecurity. This, however, raises another question: "which are all the errors in (the engineering of) a system that make it possible for a cybersecurity attack to happen?". More rigorously, which scientific theory predicts all possible errors that causes cybersecurity attacks?

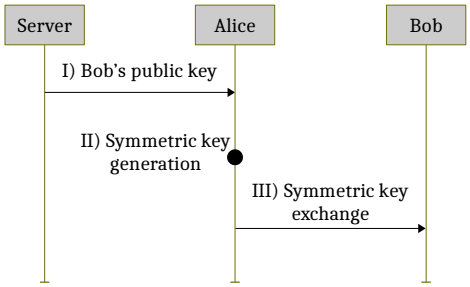


Figure 1. Abstraction of an ad-hoc esemplificative protocol execution

With our approach, a list of weaknesses emerges from the mathematical formulation of the system 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.

[3]In fact, engineering models such as the V-model start with the requirements definition, and physical and functional architecture design, before the logic of protocols is even considered.

[4]The Dolev-Yao attacker model can be considered as abstract model of an attacker that can be used for the analysis of protocol specifications.

c) Structure: We start, in Section II, by detailing the problem statement, reporting a literature review on the main concepts and definitions related to security. We formulate a security hypothesis in Section III; which we use to propose a theory on system security in Section IV. In Section IV-C, we apply our theory for the tool-assisted security risk assessment of a Cyber-Physical System (CPS), with an ad-hoc example, based on the SWaT testbed [29]. This application shows how our theory can be used to predict all of the possible security weaknesses of a system, allowing the falsification of our theory. In fact, if any security weaknesses were to be found in a system and not predicted by our theory, the theory could be declared incomplete. Similarly, if a security weakness would be predicted by our theory but found to be impossible to realize, our theory could be declared as wrong.

II. Literature Review

In most of the natural languages the concepts of safety and security are not syntactically differentiated and both terms (safety and security) are expressed by the same word, e.g. sicurezza in Italian. A semantic distinction between safety and security is correlated to a belief[5] that safety deals with *accidents* (i.e. an unfortunate incident) posed by the natural environment (e.g. natural events such as wearing of hardware components), whereas security deals with *incidents* posed by mankind (e.g. attackers and bugs). The fundamental difference between nature and mankind (and, in turn, between safety and cybersecurity) is believed to be on the different intents (incidents are intentional, whereas accidents are not) of the causes that generates a threat. Namely, nature is believed not to have malicious intents (but unfortunate causes-effects), whereas threats generated by mankind are malicious[6]. This conclusion seems to be against a general formulation of a cybersecurity theory; how can we define a theory that predicts what a human being will do? 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 [9]), vulnerabilities (e.g. CVE [7], NVD [47]), and attacks (e.g. CAPEC [6], ATT&CK [36]). 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 security 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 security 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 [42] for an overview of the evolution of the CIA triad). The link between security requirements and vulnerabilities is reported in the NVD databases by the CVSS[31]) scoring system. The CVSS evaluates of the severity of a vulnerability by means of different metrics (such as attack complexity and user interaction) and quantitatively evaluates the impact on the CIA triad. While security 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 correlate security requirements to necessary and sufficient conditions (e.g. mitigations) to declare a system secure [20]. Nonetheless, the extensive body of literature have scientific foundations, for example, providing either providing necessary or sufficient conditions for cybersecurity. As a driver for our argumentation, we start by reviewing the key concepts in the cybersecurity domain.

A. Terminology

Most of the safety-preserving principles in the field of engineering of safety-critical cyber-physical systems (such as elevators and aircraft), upon which safety requirements are defined (e.g. in standards such as the IEC 61508 or 61511[1]), are based on empirical tests and measurements. While reasoning by induction based on the empirical observation should be avoided in general, since it may easily lead to false beliefs [34], inducing general principle by means of tests is often justified by the supposed impossibility of defining a theory that correctly predicts failures. A failure of a wire due to environment (e.g. due to humidity, dust, heat &c) is defined from empirical evidences and processes have been standardized to test qualities of hardware components. This process completely breaks down when a malicious environment (i.e. an attacker) is considered instead of the (supposedly honest and predictable) natural environment. Therefore, the same approach that is in use for safety, seems not to be applicable for cybersecurity (e.g. for cybersecurity testing). An overview on the aforementioned aspects of safety and cybersecurity is depicted in Figure 2 and is used as a baseline for a definition of the terms that structure our current understanding of cybersecurity.

- *Mankind* "refers collectively to humans" [25], while the concept of *Nature* is related "to the intrinsic characteristics that plants, animals, and other features of the world develop of their own accord" (e.g. the physical universe) [26].
  - There exist several terms to refer to an *attacker*, e.g. threat agent or threat source, but, from now on, we consider the Causality principle to be the *threat source*, Nature or Mankind to be the *threat agents* and an

[5]A belief has to be intended as a proposition which is supposed to be true by the majority of people in our society, without a scientific underlying theory but based on partial empirical evidences or inductive reasoning based on partial empirical evidences.

[6]Of course, logical flaws or bugs may be introduced by other means (e.g. ignorance) without explicit malicious intents, but the exploitation of those flaws is considered (for now, and detailed afterwards in the article) malicious, and thus we consider any vulnerability to be malicious even if it is due to the lack of skills.

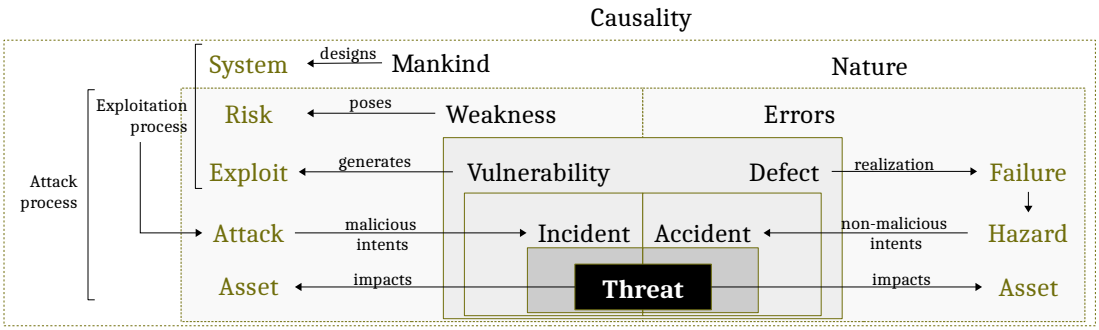


Figure 2. Overview of keywords related to security and safety

attacker as a specific malicious threat agent which materializes a threat.

- *Vulnerability*[7], as defined in [32] (and adopted in [4]), is a "weakness in an information system, system security procedures, internal controls, or implementation that could be exploited by a threat source". On the one hand, the definition is broad to enclose as much causes (that generates a vulnerability) as possible, on the other hand the term vulnerability should have a complete and sound definition, so that no other causes (e.g. other sources) but the ones in the definition are responsible for a vulnerability Furthermore, the term "threat sources" used in the definition in [32] may be identified with both Nature and Mankind, not differentiating between safety and security.

As depicted in Figure 2, a vulnerability does not necessarily become a threat for the system, unless exploited "through a channel that allows the violation of the security policy [...]" [32]. For example, a software or procedure that takes advantage of the vulnerability causing an *attack* to the system may result in several correlated incidents and threats. The process of exploitation of a defect as a vulnerability is reported in Figure 2 such that the difference between exploit and failure, and attack and accident, is to be found just in the maliciousness of the intents that causes this process (i.e. excluding the intent, the terms are just syntactic transformation from a vulnerability to defect, from accident to incident).

- *Weakness*. The definition given by the MITRE in [14] 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 a hacker 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.
- *Causality* refers to the causality principle; defined in [10] as "Causality is a genetic connection of phenomena through which one thing (the cause) under certain conditions gives rise to, causes something else (the effect). The essence of causality is the generation and determination of one phenomenon by another. In this respect, causality differs from various other kinds of connection, for example, the simple temporal sequence of phenomena, of the regularities of accompanying processes".
- An *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)." [24].
- An *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 note that for the purpose of this article, we do not want to focus on a specific organization or business to define asset but, in general, on any abstract organization (e.g. a company or a society). We do not consider ethical hackers as attacking a system. In fact, we consider the term *hack* as non-malicious (as, e.g. in [45]).
- A *Threat*, as defined in [32], 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".
- *Defect*: "anything that renders the product not reasonably safe" [38] (i.e. a characteristic of an object which hinders its proper usability).

[7]The term vulnerability is not present in the Encyclopedia of Cryptography and Security, while it is used in 12 entries (such as in the definition of "penetration testing" [5]) highlighting how commonly this word is used without a proper supporting semantics.

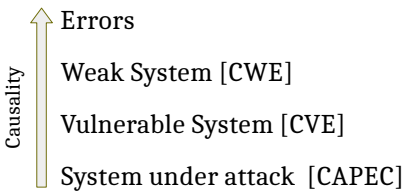


Figure 3. Etiology of cybersecurity

- *Failure*, as defined in[23], is "a state of inability to perform a normal function". The term is structured and detailed in [32, 13], but relying on an abstract notion of failure without a specific definition.
- *Hazard*, "a potential source of harm" [13].

Our literature review shows that most of the definitions relate insecurity to dis-honesty (also called maliciousness or adversarial) of an agent (often called adversary or attacker). In fact, weaknesses become vulnerability if an attacker exploits them in an attack. This, however, just moves the problem of defining what cybersecurity is as the problem of defining what a dishonest agent can or cannot do in a system, where an agent is any virtual or physical entity of the system or using the system (e.g. a device, a software, or a human being) and dishonesty is not necessarily related to malicious motivation but also to incompetence or lack of skills. As in the Dolev-Yao theory, we may correlate "being dishonest" to "not following the intended behavior/rules". In the case of a generic system, a dishonest agent is, therefore, any agent that doesn't follow the intended behavior (or functionality or logic) of the system. For example, a software can be considered an agent of the system and whenever it has a bug, it can be exploited causing an Incident. However, this may lead to the conclusion that the dishonest behaviors of agents cannot be defined in general (e.g. due to the heterogeneity of agents and systems) and that huge repository of dishonest behaviors should be kept as definition, such as CAPEC [6]; or that dishonesty of an agent with respect to a security protocol should be defined as a number of predefined actions as in [52, 3, 2, 39] (to name a few)[8]. As depicted in Figure 3, in order to define a theory on cybersecurity we may say that the presence of vulnerabilities is a necessary condition in order to cause an Attack in the system. Those vulnerabilities are, in turn, caused by the presence of weaknsesses in the system. Weaknesses are errors in the design or implementation of a system. Therefore, a theory on cybersecurity should first predict the errors in a system design.

III. Security Hypothesis – ABF Theory of Agents

In order to address the problem raised by Herley, we shall define how to distinguish between a secure and an insecure system. While most of the literature have correlated the problem of in-security to the maliciousness of agents interacting with the system, we show that security doesn't seem to stem from a malicious nature but, rather, insecurity raises from the lack of well-defined security requirements for the development process of a system. We argue that the high number of security vulnerability reported today are simply the realization of potential system configurations, which deviate from the nominal behavior because the *intended (or nominal) behavior* of the system is not precisely defined in the specification at the very early stages of the engineering process.

As a reference, we consider the following steps of the engineering process and our objective is to be able to predict the security requirements instead of axiomatically assume them.

1) *System Specification*, where the *functional and physical requirements* are defined
2) *Architecture Design*, where the specification is structured into *functional and physical architectures*
3) *Security Risk Assessment*, where the potential *weaknesses* (errors) are identified and, consequently, *security requirements* are captured.

Given that, in our investigation, we changed the focus from the attacker to the potential designs of a system, we start by defining a general framework for the definition of a system. On top of this framework we will identify system weaknesses as potential design errors.

A. Epistemology and Multi-Agent Systems

The ISO/IEC/IEEE 15288:2015 (System Life Cycle Processes) provides a definition of system as "A combination of interacting elements organized to achieve one or more stated purposes."[46]. If a system is structured into multiple sub-systems, a sub-system can be defined in the same way as a system is defined (i.e. as a composition of interacting elements). A hierarchy of sub-systems can be limited by a "bottom-subsystem" often called device (or element in the aforementioned definition) which is not itself composed by other elements but is considered atomic. For the sake of simplicity, we consider a device as a system with only one element: itself. We then first define a system as composed by a single element and then extend the definition to a "combination of interacting" elements. Given that we use concepts from the multi-agent system research field, instead of using the word element we will use the word agent.

There is no agreement between the research communities (e.g. Multi-Agent-System, Epistemic Logic) on which are the constituent of an agent. However, the same ideas revolved around for thousands of years. Some relevant examples for our objective are the following[9].

- In [21], Hintikka describes the difference between knowledge and belief (as epistemological concepts), and the

[8]In formal protocol verifiers, in the DY model, the attacker has a set of hardcoded inference rules that defines how the attacker deduces new messages based on the observed message exchange.

[9]It is interesting to notice that in [12], the author states: "The logical criterion also may be used in three senses – of the agent, or the instrument, or the "according to what"; the agent, for instance, may be a man, the instrument either sense perception or intelligence, and the "according to what" the application of the impression "according to" which the man proceeds to judge by means of one of the aforesaid instruments.". The author than proceeds showing that even in this framework, knowledge still is non-apprehensible.

whole Doxastic logic defines in details how beliefs can be formalized.
- In [22], Hintikka describes the concept of information and the difference with knowledge and belief.
- In [43], the authors defines an agent as a tuple of assertions, beliefs, and facts.

For our argument, as in [43], an agent is composed by its knowledge, beliefs, and the information or assertions it provides; where knowledge is defined, as in [48], as a set of proposition known by an agent, such that: (i) knowledge requires belief, (ii) knowledge require truth, (iii) knowledge must be *properly justified*, and the only objective of information is to exchange beliefs between agents. As defined in [22], "information is specified by specifying which alternatives concerning the reality it admits and which alternatives excludes". This means that if we consider a propositional variable (which admits the two alternatives True/False) its information is defined as Believed to be True/False and not Believed to be the opposite. Due to the definition of information and then with its relation to the probabilistic correlation to truth/reality, we consider information in relation with an agent's beliefs. Similarly, we consider beliefs to define the actual behavior of an agent or a system; i.e. we consider behaviors as belief revision systems. The difference between knowledge and information as used by Hintikka, and facts and assertions as used in [43], are not precisely discussed in [43]. Furthermore, those epistemological concepts are difficult to define [51]. For example, Hintikka in [22] states that "A purely logical definition of information is impossible". However, a detailed formalization of those concepts is out of the scope of this paper.

We now define those concepts abstractly, inspired by the $\mathcal{ABF}$-framework defined in [43] and, in Section IV we detail them as engineering concepts.

1) We consider information only when the intention of exchanging that information is from a sender to recipient is defined; and we call it *assertion*
2) Similarly, the portion of *beliefs* we consider for system engineering is the one that builds (input) or describes (output) the behavior of an agent (strategy rules [10]), and
3) We consider a set of axiomatic *facts* (definitory rules[11]) instead of considering the more general epistemic definition of knowledge. Specifically, facts describes:
   - The functional architecture of each agent
   - The physical/structural (HW/SW) architecture of each subsystem of agents and agent within a subsystem

[10]"The logical structure of information is one of the most basic and one of the most basic and one of the simplest thing in the wide and wonderful world of logical analysis. This point can be put in a deeper perspective. A distinction [...] ought to be made [–] between two kinds of rules (or principles) in any strategic activity like knowledge seeking. On the one hand you have the rules that define the game, e.g. how chessmen are moved on a board. The can be called *definitory* rules. They must be distinguished from rules [...] that deal with what is better and what is worse in the game in question. Definitory rules do not say anything about this subject. Rules which do can be called *strategic rules*" – Hintikka in [22]

[11]Facts are definitory rule as they don't define how a real system is finally implemented but how it should be, trough a series of requirements. Those requirements may not hold, through insecurity, in the implemented system.

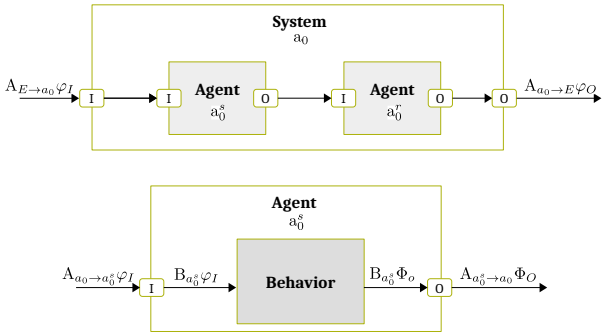


Figure 4. Example of system and agent structure

We give a graphical representation of (sub-)system and agent in Figure 4. The notation will be introduced in the next section but, intuitively, Figure 4 shows a system composed by two agents. The system has two ports denoted by "I" (with incoming assertions) and "O" (with outgoing assertions). The bottom part of Figure 4 details the structure of a single agent which is not decomposed into other (sub-)agents. In this case, ports translates incoming assertions into beliefs that feeds the behavior block, and outgoing beliefs from the behavior block are then translated back into (outgoing) assertions.

B. Mereo-topological Reasoning

Similarly to [43], we define agents as a meronomy (an hierarchy of Part-Whole relations) over the constituent 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 [44, 55, 35], considering the relations in Table I. 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 [47, 37] so that we don't need to rely on a scoring system (such as the CVSS) to assign a quantitative evaluation of the security of each entry. Instead, we want a precise calculation of the number of insecurity configurations to emerge from the mathematical formulation of our cybersecurity theory.

A mereotopology, as defined e.g. in [35], is an ordered mathematical structure where the basic relation between regions is the reflexive and symmetric relation *Connects With* $\subseteq$, that we use to order a universe of agents Ag (see in Table I). We use the Region Connection Calculus (RCC), as defined in [27, 16], 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. In the text, for brevity, we will often focus only on RCC5 without loss of generality. Using RCC5 instead of RCC8

RCC3	RCC5	RCC8	Terminology	Notation	Definition
			Connects with	$C(X,Y)$	$X \subseteq Y$
			Disconnected from	$\neg C(X,Y)$	$X \not\subseteq Y$
			Part of	$P(X,Y)$	$\forall Z\ C(Z,X) \rightarrow C(Z,Y)$
			Overlaps	$O(X,Y)$	$\exists Z\ P(Z,X) \wedge P(Z,Y)$
●			**Overlaps Not Equal**	$ONE(X,Y)$	$O(X,Y) \wedge \neg EQ(X,Y)$
●	●	●	**Equal to**	$EQ(X,Y)$	$P(X,Y) \wedge P(Y,X)$
●	●	●	**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-*i*nverse**	$PPi(X,Y)$	$P(Y,X) \wedge \neg P(X,Y)$
		●	**Externally Connected**	$EC(X,Y)$	$C(X,Y) \wedge \neg O(X,Y)$
		●	**Tangential PP**	$TPP(X,Y)$	$PP(X,Y) \wedge \exists Z\ [EC(Z,X), EC(Z,Y)]$
		●	**Tangential PPi**	$TPPi(X,Y)$	$TPP(Y,X)$
		●	**Non-Tangential PP**	$NTPP(X,Y)$	$PP(X,Y) \wedge \neg\exists Z\ [EC(Z,X), EC(Z,Y)]$
		●	**Non-Tangential PPi**	$NTPPi(X,Y)$	$NTPP(Y,X)$

Table I
RCC3, RCC5, AND RCC8 RELATIONS BETWEEN REGIONS X, Y AND Z

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 Table I, we summarize the axioms of the RCC (see, e.g., [17]). 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 has 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 already presented in [43] it follows that, defining a system with a fixed number of regions, there exist 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 the next paragraph the calculation done in [43].

a) 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 [16]) 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
- *RCC8* — theoretical: $8^3 = 512$, correct: 193

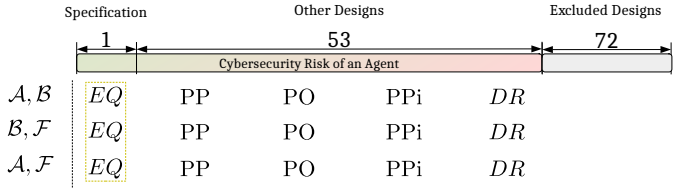


Figure 5. Cybersecurity risk for a single agent

Hence, even if considering a different number of regions than the three $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{F}$ exponentially affects the number of theoretical agents, the application of RCC downscales that number of a factor that ranges from 1.8 to 2.5. In addition, using RCC5 we consider 3.6 times more (different) types of agents than RCC3, but using RCC8 would allow us to consider 3.5 times more different agents. In the quantitative evaluation of a single agent, depicted in Figure 5, 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.

C. Qualitative Evaluation of Agent Space in $\mathcal{A},\mathcal{B},\mathcal{F}$

While a quantitative analysis reveals how many possible configurations of an agent (i.e. a system) exist w.r.t. the $\mathcal{A},\mathcal{B},\mathcal{F}$ theory (i.e. $54/125$ in RCC5), a qualitative analysis of the different configurations describe the configurations allowed by the $\mathcal{A},\mathcal{B},\mathcal{F}$ theory, and how those configurations can be categorized. In Table II, 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 security risk related to a generic agent, the risk is highest on the top left corner of the matrix, lowest on the bottom right corner.

	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($\mathcal{A}$,$\mathcal{F}$) PO($\mathcal{A}$,$\mathcal{F}$)	DR($\mathcal{A}$,$\mathcal{F}$)	DR($\mathcal{A}$,$\mathcal{F}$) PO($\mathcal{A}$,$\mathcal{F}$)	DR($\mathcal{A}$,$\mathcal{F}$)
PO($\mathcal{B}$,$\mathcal{F}$)	DR($\mathcal{A}$,$\mathcal{F}$) PO($\mathcal{A}$,$\mathcal{F}$) PP($\mathcal{A}$,$\mathcal{F}$)	T($\mathcal{A}$,$\mathcal{F}$)	DR($\mathcal{A}$,$\mathcal{F}$) PO($\mathcal{A}$,$\mathcal{F}$) PP($\mathcal{A}$,$\mathcal{F}$)	PO($\mathcal{A}$,$\mathcal{F}$) PPi($\mathcal{A}$,$\mathcal{F}$)	PO($\mathcal{A}$,$\mathcal{F}$)
PP($\mathcal{B}$,$\mathcal{F}$)	DR($\mathcal{A}$,$\mathcal{F}$) PO($\mathcal{A}$,$\mathcal{F}$) PP($\mathcal{A}$,$\mathcal{F}$)	PO($\mathcal{A}$,$\mathcal{F}$) PP($\mathcal{A}$,$\mathcal{F}$)	PP($\mathcal{A}$,$\mathcal{F}$)	PO($\mathcal{A}$,$\mathcal{F}$) EQ($\mathcal{A}$,$\mathcal{F}$) PP($\mathcal{A}$,$\mathcal{F}$) PPi($\mathcal{A}$,$\mathcal{F}$)	PP($\mathcal{A}$,$\mathcal{F}$)
PPi($\mathcal{B}$,$\mathcal{F}$)	DR($\mathcal{A}$,$\mathcal{F}$)	DR($\mathcal{A}$,$\mathcal{F}$) PO($\mathcal{A}$,$\mathcal{F}$) 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 II
RCC5 COMPOSITION TABLE OVER 3 REGIONS. THE RESULTS SHOW THAT THERE EXIST 54 POSSIBLE RELATIONS AND THE COLORING ANTICIPATES THE IDEAL RISK MATRIX (GREEN THE SECURE STATE WITH 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($\mathcal{A}$,$\mathcal{F}$)}

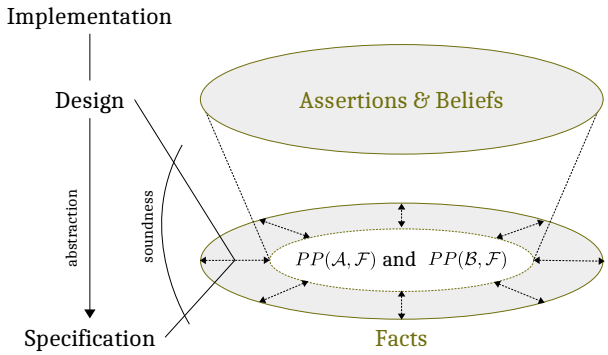


Figure 6. Example relation between facts, and assertions and beliefs

In Figure 6, 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 categorize agents by first analyzing the relations between each pair of regions defining an agent (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.

a) rcc($\mathcal{A}$,$\mathcal{B}$) – Collaboration: In order to reason on the relation between assertions and beliefs we first need to consider that, by definition, assertions are defined as transfer of information between two agents, e.g., a and b. Therefore, as depicted in Figure 4, an agent has two main categories of assertions, input and output assertions. Given an agent a and a collection of asserted predicates Φ, 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, $\mathcal{A}_{a\to r}\Phi$. We shall consider two pairs of regions[12]:

- 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. With more details, the ideal specification of the functional architecture, along with the expected inputs, defines the functional behavior of an ideal system. 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. With more details, if all the outputs (assertions) of the functional architecture can be produced, the functional architecture is complete.

b) 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 defines what needs to be true in the system, the relation of assertions an 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. We note that it is often implied that the intention behind circulating false information discerns a dishonest and an incompetent agent. However, we consider honesty related to sharing truths[13]. 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).

1) $\mathcal{A}, \mathcal{B}, \mathcal{F}$ Security Enumeration (SE): The following security requirement for a CPS specification can be summarized:

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

SE-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*. For example, the inputs of the user of a SW must be sanitized to exclude

[12]"I am not asserting, as Lotze did, that a relation between X and Y consists of a quality in X and a quality in Y – a view which I regard as quite indefensible. – I assert that a relation Z between X and Y involves the existence in X of the quality "having the relation Z to Y" so that a difference of relations always involves a difference in quality, and a change of relations always involves a change of quality." – Ellis J. McTaggart, The Unreality of Time[30]

[13]"[...] truth exists only in the good man, but the true in the bad man as well; for it is possible for the bad man to utter something true." Sextus Empiricus, Outlines of Pyrrhonism, II-83[12]

deviations w.r.t. the expected inputs of the functions implemented in the SW. Another example is the type checking between allowed inputs and expected inputs.

SE-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.

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

SE-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 SE define the properties of a secure system, and correlated weaknesses can be found in the CWE dataset. For example, SE-1 can be sees as correlated to the weakness class of "Improper Interaction Between Multiple Correctly-Behaving Entities" defined by the CWE–435, SE-2 with the "Insufficient Control Flow Management" defined by MITRE in the CWE–691, and SE-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[9], while the other classes of weaknesses do not have a direct counterpart in our theory; we believe they can be sees as sub-classes, but a full comparison with the CWE is out of the scope of this article.

We are now in the position to define what a secure system is (with respect to the $\mathcal{ABF}$-framework) and, based on that definition, what the security risk is and how to quantify it in a risk matrix.

Definition 2. Security of a System or an Agent – *A secure system is a system where SE-1, SE-2, and SE-3 holds for each agent composing 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 – *A risk is the whole space of potential designs of a specification with respect to the $\mathcal{ABF}$-framework.*

The definition of Risk leads to the risk matrix in Figure 5, defined as follows.

Definition 4. Risk Matrix – *The risk matrix, as summarized in Table II, 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 security 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. Therefore, a risk matrix should capture the number of insecure configurations of a system, rather than predicating over likelihoods of weaknesses exploitation.

IV. Prediction of Security Weaknesses

Several standards mandate a secure-by-design approach in which cybersecurity shall be considered at the very early stages of the design process. For example,

1) DO-326A – "Airworthiness Security Process Specification" requires a cybersecurity risk assessment of the design and "are the only Acceptable Means of Compliance (AMC) by FAA & EASA for aviation cyber-security airworthiness certification, as of 2019" as pointed out by SAE in [41].
2) NIST 800-82 [49] – "guide to Industrial Control System (ICS) Security"
3) J3061:2016-1 [40] – "Cybersecurity Guidebook for Cyber-Physical Vehicle Systems" defines "set of high-level guiding principles for Cybersecurity as it relates to cyber-physical vehicle systems" and states that "incorporate Cybersecurity into cyber-physical vehicle systems from concept phase through production, operation, service, and decommissioning"

However, standards do not describe in detail how to perform a security risk assessment and only vaguely define the overall objective, which can be summarized as to provide an understanding of the potential security risks. Roughly speaking, a security risk assessment (e.g. CORAS [28]) methodology starts from the (i) identification of assets, (ii) determines the threat scenarios (correlating vulnerabilities, threats and incidents) e.g. threat diagrams in the CORAS approach, (iii) calculates the risk as a function of impact and likelihood of threat scenarios, and (iv) finally provides treatments (mandating a partial re-design) and residual risks. All the methodologies and tools we reviewed (e.g. Threatmodeler [50], CORAS [28], SECRAM [15]) rely on the expertise of the person who performs the risk assessment for the identification of threats and for the quantitative estimation of risks. As Herley puts it [19], "we don't have a test for security and we don't have a metric for security" while standards require a security risk assessment which, in turn, mandates a metric to evaluate the security risk. In contrast, in this section, we define how to specify a CPS in the $\mathcal{ABF}$-framework and we identify a security metric.

So far, we have reasoned on systems in the abstract, considering generic Multi-Agent System (MAS); mapping Epistemological concepts to MAS. We now map MAS to CPS and show how this mapping allows us to predict security weaknesses in a system.

A. From Multi-Agent to Cyber-Physical Systems

As depicted in Figure 7, we relate MAS and CPS as follows.

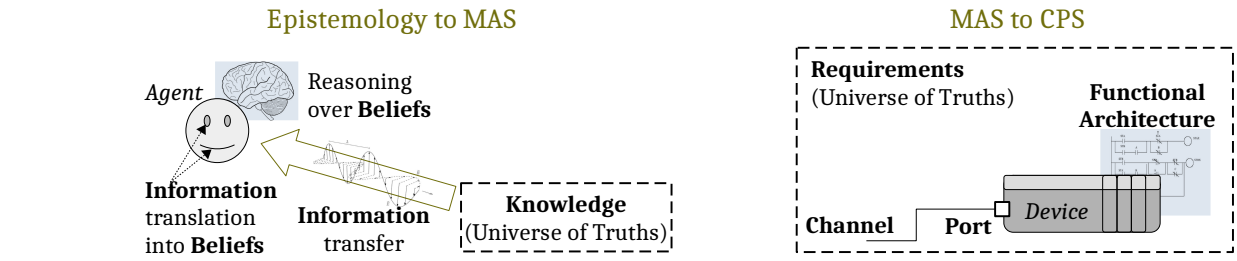


Figure 7. Mapping Epistemological Concepts to (Cyber-Physical) Systems Engineering

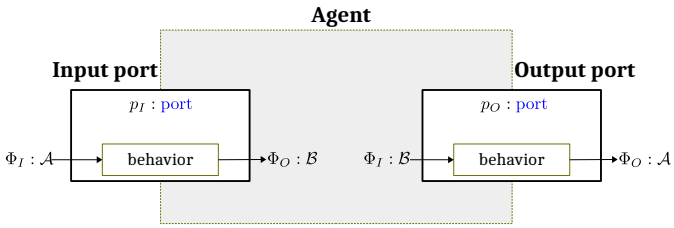


Figure 8. Input and Output Port of an agent

- 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 in Figure 7. In this case, the system is 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 knowledge is related to the *requirements* that, in a factorial way, describe how the CPS shall behave and its *physical architecture*.

a) Input and Output Ports: Since the $\mathcal{ABF}$-framework is a theory of agents, we could consider ports as agents that allows 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*. While a channel transfers information between agents and a functional architecture elaborate information, a port is simply a connector between a channel and a functional architecture. One, however, may argue that a similar connect is needed between a channel and the port itself. While this is not excluded by the $\mathcal{ABF}$-framework, would obviously lead to an infinite nested structure of ports. In order to avoid this, 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 belief into assertion.*

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 the input-port or $rcc(\mathcal{A}_{r\to *}, \mathcal{B}_a)$ for the output-port. A port is, in fact, syntactic sugar to express the relation between assertions and beliefs. We only define secure input port but the definition of a secure output port is symmetrical. Moreover, 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 because we define a port as if the RCC equality relation holds between the input/output assertions and the input/output beliefs. In contrast, assuming any RCC relation, between inputs and outputs of a port, which is not an equality can be considered as generating a weaknesses, as we describe in more details in the next paragraph.

b) Port Weaknesses:

Theorem 1. *Port Weaknesses. From our definition of ports, it follows that there exist only the following* six *types of weaknesses, generating six types of insecure port in RCC5 (the notation is reported for input-ports, i.e. on the left-hand side of the arrow):*

W1) replace port, *where assertions reach the port but is 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 some new information is believed by a as incoming from the port but s didn't send it*

W4) injection port, *where information coming from s is substituted with new information which becomes believed by a*

W5) selective port, *where some information passes the port and part is either:*

W5.1) drop,

W5.2) drop+insert.

Figure 9. Communication over a Mono-directional Channel

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 open region can be written as union of elements of that base. We distinguish between regions that contain information and regions that don't by writing the latter as $\emptyset$.

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 and *full drop* the latter case), or $\mathcal{A} \neq \emptyset \wedge \mathcal{B} \neq \emptyset \wedge \mathcal{A} \neq \mathcal{B}$ which we call *replace* (i.e. drop and insert)
3) if $PP(\mathcal{A}, \mathcal{B})$) then $\mathcal{B}$ contains and extend $\mathcal{A}$ which we call *injection*
4) if $PPi(\mathcal{A}, \mathcal{B})$) then $\mathcal{A}$ contains and extend $\mathcal{B}$ which we call *selective 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*

□

c) 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 start by considering 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 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[14].

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 made by the sender and the ones received by the receiver, i.e. $rcc(\mathcal{A}_s, \mathcal{A}_r)$.

d) 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 $(7^2) - 2 = 47$ 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, -2); where only 43 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*

e) Security Weaknesses – The RIDI-Hypothesis: It is important to note that 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. 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**FIX**[15]. As depicted in Figure 10, 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. Io order to define a functional block without encountering an infinitely recursive definition, we must reach the same conclusions as for the channel. Therefore, describing the information as flowing over a channel or in a functional block is purely syntactic sugar.

[14]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.

[15]**mv:** Questo è difficile da seguire. Non si può spostare a dopo aver definito i functional blocks e il modo in cui li interpretate nel vostro MAS?

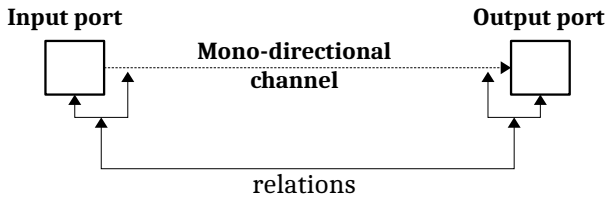


Figure 10. Relations between Ports and Channel

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 security properties to mitigate security 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*

f) Functional Architecture: A functional architecture takes information as input-beliefs and transforms the information into output-beliefs. Those transformations occurs within the functional architecture, where functional blocks transforms 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 a region **FIX**[16] of input-beliefs and outputs a region of output-beliefs. A functional architecture is an interconnected system of functional blocks.*

It is evident that the quality of a functional block cannot be determined by the difference between its inputs and outputs (as we did for ports and channels). 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 a port semantics 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).

[16]**mv:** Perché qui specificate region mentre nelle definizioni di port e channel non lo facevate?

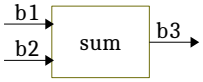


Figure 11. Example of a Functional Block

W50) $PO(\mathcal{B},\mathcal{F})$ 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.

W51) $PP(\mathcal{B},\mathcal{F})$ part of the expected outputs are not generated in response to the correct inputs

W52) $PPi(\mathcal{B},\mathcal{F})$ the components correctly performs 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)

g) 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 we stated in Section IV and depicted in Figure 6, facts are strategic 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, consider a functional block that performs the summation of two inputs, as in Figure 11, 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 the functional block sum while the region $\mathcal{B}_{1+2}$ represents the expected outputs of an ideal implementation of the requirement r. Therefore, the functional block correctly implements the requirements.
- $DR(\mathcal{B},\mathcal{F}) = DR(\mathcal{B}_3,\mathcal{B}_{1+2})$, the function block never implements the requirements.
- $PP(\mathcal{B},\mathcal{F}) = PP(\mathcal{B}_3,\mathcal{B}_{1+2})$, there exist some inputs for which the output is incorrect.
- $PPi(\mathcal{B},\mathcal{F}) = PPi(\mathcal{B}_3,\mathcal{B}_{1+2})$, there exist some outputs that are not the result of the summation of the inputs but given the correct inputs the function always outputs correctly.
- $PO(\mathcal{B},\mathcal{F}) = PO(\mathcal{B}_3,\mathcal{B}_{1+2})$, 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.

h) 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 security 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 can give an example but research is needed to apply our theory to the behavior of agents.

If we consider a functional block that generates nonces (i.e. number used only once), the semantics of this block must define a procedure such that a freshness requirement holds. By the RIDI-Hypothesis we have that the following weaknesses may occur:

- Replace: the output-nonces of the block are replaced with non-fresh numbers
- Insert: the output-nonces are concatenated with other numbers, resulting in non-fresh nonces. For example, the nonces 110 and 111 are modified by inserting a 1 and a 0 at the beginning and end respectively, resulting in the two identical (and the non-fresh) nonces 1110 and 1110
- Delete: part of output-nonce is dropped. For example, the last bit of 1101 and 1100 is dropped, resulting in the two equal and non-fresh nonces: 110 and 110.
- Inject: substitution at bit level results in non-fresh nonces. For example, 110 becomes 111 becomes equal by flipping either the last 0 of the former nonce or the last 1 of the latter.

We summarize our results by categorizing the weaknesses predicted by our theory into: data-flow-related and functionality-related weaknesses; as in Table III.

B. Security and Insecurity of a System

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

Hypothesis 2. System Insecurity Design – *If, given a system specification as a collection of requirements on the system, 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 $E \subseteq V \times E$ 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 $G' = \langle E' \times V' \rangle$ 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 Π 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 σ 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 π 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 π_{eq} as the only secure configuration.

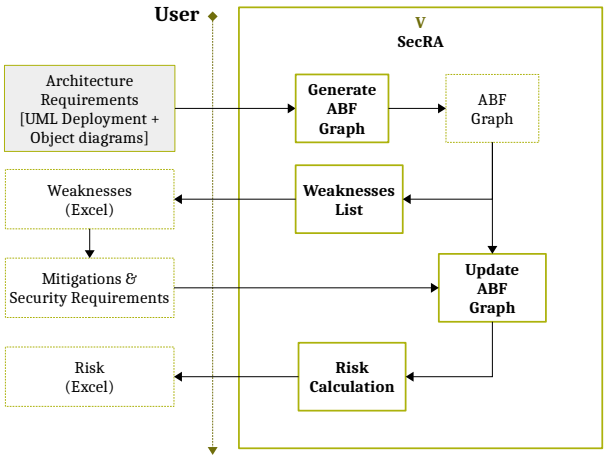


Figure 12. Security Risk Assessment Tool

C. Cybersecurity Risk Assessment

In order to test our theory we implemented a tool-chain (open-source under the AGPLv3 license and available at [53]) for the identification of weaknesses and the precise calculation of potential insecure configurations. The engineering of the $\mathcal{ABF}$ is summarized in the UML Class diagram in Figure 16 (in appendix).

As depicted in Figure 12, the security risk assessment process starts with the definition of the use cases and architectural requirements; the specification of a system. 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 a Node with (physical) ports, and agent's ports are connected via 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*

RCC5	Quantity: Data Flow	Quality: Requirements Adherence
EQ	Expected/Nominal	Expected/Nominal
DR	Drops all the inputs and inserts new malicious 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 the inputs are handled properly, nor all the expected outputs are always generated when correct inputs are given

Table III
WEAKNESSES CATEGORIZED

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 internal structure which represents the specification (called ABF graph). The 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. 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 internal structures of the tool can be viewed as PDF and the results are reported into an Excel file. The Excel file also reports the total number of configurations as indicating the security risk associated to the specification. A user can change the status of each weakness in the Excel file, from the initial (default) status (open) to "mitigated". As a consequence, 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 Excel file.

In our approach, security requirements are not imposed by the specification but are automatically extracted by our tool as potential weaknesses. Those weaknesses are related to the different insecure configurations of the specified system.

The risk matrix is often defined as a function of likelihood and impact of attacks. We suggest that security 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.

a) Case Studies: We report the results of the evaluation of a water level reader (sensor ad-hoc example). As in Figure 13, we defined 2 agents: sensorInTank and sensorBoard, which represent the physical reader that needs to be placed in a tank of water, and the board that interprets the readings and outputs them as digital signals. The two components are connected by a wire. In Figure 14 we report the functional architecture that receives the incoming communications from the sensor in the tank, and communicates them encrypted. The results, summarized in Figure 15 reports more than 16 millions possible scenarios in which at least one component diverge from the specification.

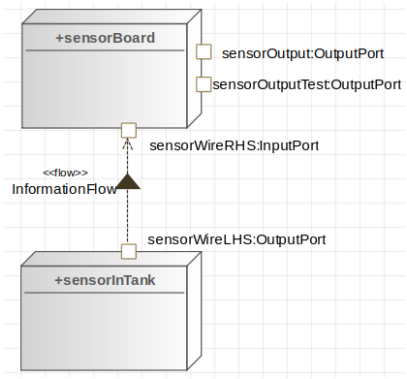


Figure 13. Water Level Reader Deployment Diagram

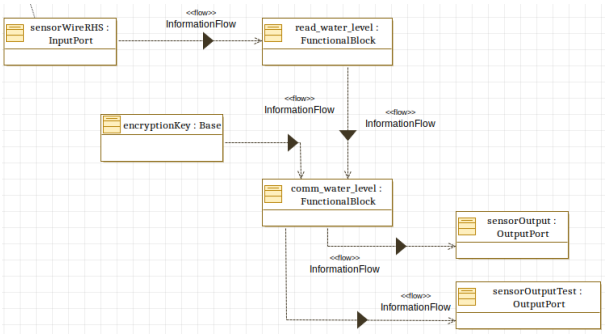
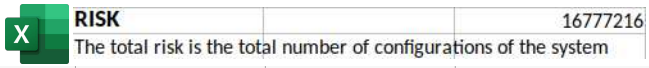


Figure 14. Sensor Board Object Diagram

V. CONCLUSION AND FUTURE WORKS

FIX[17] We proposed a foundational theory on security, arguing that security-related issues are not related to the maliciousness of an agent but to the vagueness of the security controls on the engineering processes. The current state-of-the-art processes allow, given a specification, an engineer to design the system in such a way that security issues arise due

[17]**mv:** Se ho capito bene, detto a parole mie, quello che il vostro approccio permette di fare, dati requisiti e architettura del sistema, è di elencare in maniera sistematica tutti i "punti" in cui potrebbero sorgere dei problemi. Spiegherei un po' meglio, qui nelle conclusioni o nella sezione precedente, come questo approccio dovrebbe essere usato nella pratica... Voglio dire che non mi sembra necessariamente in opposizione ad altri approcci ma magari può integrarsi a questi. Inoltre quando parlate dei passi successivi, ad esempio il considerare la generazione di test cases, se avete già qualche idea su come procedere, potreste discuterne brevemente.



RISK		16777216
The total risk is the total number of 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

Figure 15. Partial View of the results in the Excel file

to the lack of proper security risk assessment processes. Those design, again, lack of security verification processes based on a solid security foundational theory and then permit the generation of insecure implementation. The security verification and test-case generation will be the focus of our next steps.

We conclude that the problem of security is a problem related to the many possible design and, in turn, implementation given a specification. The problem is related to the epistemological search for truth, where the challenge is to relate information and belief to human knowledge. Scientifically, generalizing the human to an agent, the problem is to relate assertions and beliefs, to facts. From an engineering standpoint, by generalizing the concept of agents as architectural subsystems, the problem is to link channels (and ports) and functional architectures to requirements.

Acknowledgment

We thank Katia Santacà for the help in the development of the initial ideas, and for the helpful discussions that allowed us to make this step forward.

Appendix

The Class Diagram for the Engineering of the $\mathcal{ABF}$-framework is reported in Figure 16. A *specification* of a CPS is viewed as an aggregation of *architectures* which can describe the functional or physical requirements. The physical components 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 subsystems or components) are composed by ports and functional blocks, as an aggregation of architectures.

References

[1] International Electrotechnical Commission (IEC). *IEC 61511-1:2016+AMD1:2017 CSV Consolidated version*. Aug. 16, 2017. URL: https://webstore.iec.ch/publication/61289 (visited on 01/21/2020).

[2] Alessandro Armando, Roberto Carbone, and Luca Compagna. "SATMC: a SAT-based model checker for security protocols, business processes, and security APIs". In: *International Journal on Software Tools for Technology Transfer* 18.2 (2016), pp. 187–204.

[3] David Basin, Sebastian Mödersheim, and Luca Vigano. "OFMC: A symbolic model checker for security protocols". In: *International Journal of Information Security* 4.3 (2005), pp. 181–208.

[4] 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). URL: http://dx.doi.org/10.6028/NIST.SP.800-53r4.

[5] Tom Caddy. "Penetration Testing". In: *Encyclopedia of Cryptography and Security*. Ed. by Henk C. A. van Tilborg. Boston, MA: Springer US, 2005, pp. 456–457. ISBN: 978-0-387-23483-0. DOI: 10.1007/0-387-23483-7_297. URL: https://doi.org/10.1007/0-387-23483-7_297.

[6] *Common Attack Pattern Enumeration and Classification*. URL: https://capec.mitre.org/ (visited on 02/19/2020).

[7] The MITRE Corporation. *CVE – Common Vulnerabilities and Exposures*. Jan. 16, 2020. URL: https://cve.mitre.org/ (visited on 01/22/2020).

[8] Cas JF Cremers, Pascal Lafourcade, and Philippe Nadeau. "Comparing state spaces in automatic security protocol analysis". In: *Formal to Practical Security*. Springer, 2009, pp. 70–94.

[9] *CWE VIEW: Research Concepts*. 2020. URL: https://cwe.mitre.org/data/definitions/1000.html (visited on 02/03/2020).

[10] *Dialectical Materialism*. Progress Publishers, 1983. URL: https://www.marxists.org/reference/archive/spirkin/works/dialectical-materialism/index.html.

[11] Danny Dolev and Andrew Yao. "On the security of public key protocols". In: *IEEE Transactions on information theory* 29.2 (1983), pp. 198–208.

[12] Sextus Empiricus. *Outline of Pyrrhonism*. Prometheus Book, 1990. ISBN: 978-0-87975-597-3.

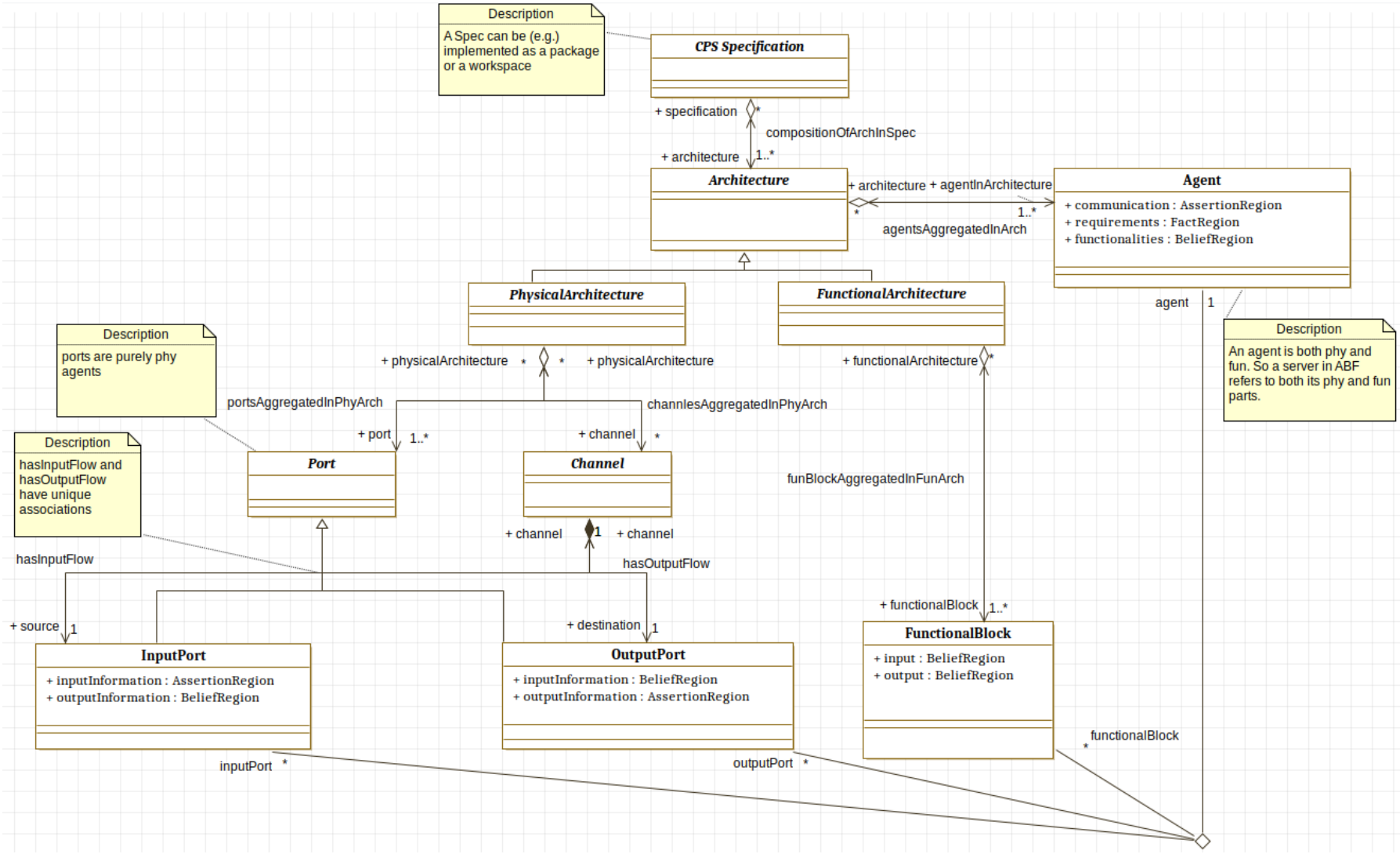


Figure 16. Security Risk Assessment – Class Diagram

[13] Institution of Engineering and Technology (IET). *Glossary of safety terminology*. Jan. 2017. URL: https://www.theiet.org/media/1435/hsb00.pdf.

[14] *FAQ – What is the difference between a software vulnerability and software weakness?* Sept. 17, 2019. URL: https://cwe.mitre.org/about/faq.html#A.2 (visited on 02/03/2020).

[15] 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.

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

[17] 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.

[18] Cormac Herley. "Justifying Security Measures—a Position Paper". In: *European Symposium on Research in Computer Security*. Springer. 2017, pp. 11–17.

[19] Cormac Herley. *The Unfalsifiability of Security Claims - Invited Talk USENIX Security*. Aug. 2016. URL: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/herley (visited on 01/15/2020).

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

[21] Jaakko Hintikka. "Knowledge and Belief: An Introduction to the Logic of the Two Notions". In: *Studia Logica* 16 (1962), pp. 119–122.

[22] Jakko Hintikka. "On proper (popper?) and improper uses of information in epistemology". In: *Theoria*. Wiley Online Library, 1993, pp. 158–165. URL: https://doi.org/10.1111/j.1755-2567.1993.tb00867.x.

[23] Merriam-Webster Inc. *failure*. 2020. URL: https://www.merriam-webster.com/dictionary/failure (visited on 01/18/2020).

[24] Wikipedia Foundation Inc. *Exploit (computer security)*. Nov. 23, 2019. URL: https://en.wikipedia.org/wiki/Exploit_(computer_security) (visited on 01/21/2020).

[25] Wikipedia Foundation Inc. *Mankind*. Dec. 19, 2019. URL: https://en.wikipedia.org/wiki/Mankind (visited on 01/15/2020).

[26] Wikipedia Foundation Inc. *Nature*. Jan. 14, 2020. URL: https://en.wikipedia.org/wiki/Nature (visited on 01/15/2020).

[27] Tsau Young Lin, Qing Liu, and Y. Y. Yao. "Logics Systems for Approximate Reasoning: Approximation via Rough Sets and Topological Spaces". In: *ISMIS*. 1994.

[28] Mass Soldal Lund, Bjørnar Solhaug, and Ketil Stølen. *Model-driven risk analysis: the CORAS approach*. Springer Science & Business Media, 2010.

[29] Aditya P Mathur and Nils Ole Tippenhauer. "SWaT: A water treatment testbed for research and training on ICS security". In: *International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*. IEEE. 2016, pp. 31–36.

[30] J Ellis McTaggart. "The unreality of time". In: *Mind* (1908), pp. 457–474.

[31] 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.

[32] 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.

[33] R. Karl Popper. *Conjectures and refutations: The growth of scientific knowledge*. New York London, 1962.

[34] R. Karl Popper. *The Logic of Scientific Discovery*. New York London, 1959.

[35] 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.

[36] MIT Research and Engineering (MITRE). *ATT&CK*. URL: https://attack.mitre.org/ (visited on 06/09/2020).

[37] MIT Research and Engineering (MITRE). *Common Vulnerabilities and Exposures (CVE)*. URL: https://cve.mitre.org/ (visited on 01/27/2020).

[38] Patricia A Robinson. *Writing and designing manuals and warnings*. CRC Press, 2019.

[39] Marco Rocchetto, Luca Viganò, and Marco Volpe. "An interpolation-based method for the verification of security protocols". In: *Journal of Computer Security* 25.6 (2017), pp. 463–510.

[40] SAE. *Cybersecurity Guidebook for Cyber-Physical Vehicle Systems*. 2016. URL: https://www.sae.org/standards/content/j3061_201601.

[41] SAE. *DO -326A and ED-202A : An Introduction to the New and Mandatory Aviation Cyber-Security Essentials C1949*. 2019. URL: https://www.sae.org/learn/content/c1949/.

[42] 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).

[43] 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.

[44] Barry Smith. “Mereotopology: A theory of parts and boundaries”. In: *Data & Knowledge Engineering* 20.3 (1996). Modeling Parts and Wholes, pp. 287–303.

[45] Richard Stallman. *The Hacker Community and Ethics: An Interview with Richard M. Stallman*. 2002. URL: https://www.gnu.org/philosophy/rms-hack.html (visited on 01/22/2020).

[46] International Organization for Standardization (ISO). *ISO/IEC/IEEE 15288:2015, Systems and Software Engineering – System Life Cycle Processes*. May 2015. URL: https://www.iso.org/standard/63711.html (visited on 02/20/2020).

[47] National Institute of Standards and Technologies (NIST). *National Vulnerability Database*. URL: https://nvd.nist.gov/ (visited on 01/22/2020).

[48] Matthias Steup and Ram Neta. “Epistemology”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Spring 2020. Metaphysics Research Lab, Stanford University, 2020.

[49] Keith Stouffer. *Guide to industrial control systems (ICS) security*. 82, pp. 16–16.

[50] Threatmodeler. *ThreatModeler*. URL: https://threatmodeler.com/ (visited on 06/09/2020).

[51] John Turri. “Is knowledge justified true belief?” In: *Synthese* 184.3 (2012), pp. 247–259.

[52] Mathieu Turuani. “The CL-Atse protocol analyser”. In: *International Conference on Rewriting Techniques and Applications*. Springer. 2006, pp. 277–286.

[53] V-Research. *V-Research Cybersecurity Repository*. URL: https://github.com/v-research/cybersecurity (visited on 06/09/2020).

[54] Mathy Vanhoef. *The 4-way handshake was mathematically proven as secure. How is your attack possible?* URL: https://www.krackattacks.com/ (visited on 06/25/2020).

[55] Achille C. Varzi. “On the Boundary Between Mereology and Topology”. In: *Proceedings of the International Wittgenstein Symposium*. 1994, pp. 261–276.