A Language for Describing Attacks on Cyber-Physical Systems

Mark Yampolskiy[a*], Péter Horváth[b], Xenofon D. Koutsoukos[c], Yuan Xue[c], and Janos Sztipanovits[c]

[a] *UniversityofSouthAlabama*

[b] *Budapest University of Technologyand Economics*

[c] *Institute for Software Integrated Systems, Vanderbilt University*

Abstract

The security of cyber-physical systems is of paramount importance because of their pervasiveness in the critical infrastructure. Protecting cyber-physical systems greatly depends on a deep understanding of the possible attacks and their properties. The prerequisite for quantitative and qualitative analyses of attacks is a knowledge base containing attack descriptions. The structure of the attack descriptions is the indispensable foundation of the knowledge base.

This paper introduces the Cyber-Physical Attack Description Language (CP-ADL), which lays a cornerstone for the structured description of attacks on cyber-physical systems. The core of the language is a taxonomy of attacks on cyber-physical systems. The taxonomy specifies the semantically distinct aspects of attacks on cyber-physical systems that should be described. CP-ADL extends the taxonomy with the means to describe relationships between semantically distinct aspects, despite the complex relationships that exist for attacks on cyber-physical systems. The language is capable of expressing relationships between attack descriptions, including the links between attack steps and the folding of attack details.

Keywords:
Cyber-Physical Systems (CPS), Security, Cross-domain attacks, Taxonomy, Attack description language

*Corresponding author. E-mail address: yampolskiy@southalabama.edu

1. Introduction

Cyber-physical systems (CPSs) have become increasingly pervasive in modern society. They are used in all kinds of unmanned vehicles and automated manufacturing plants, butmore importantly, they are used in the critical infrastructure electrical power grids, transportation systems and healthcare systems. At this time, onlya handful of attacks on cyber-physical systems have been detected in thewild. Nevertheless, it is reasonable toassume that attacks oncyber-physical systems will rapidly escalate with increasing connectivity and evolving business models. The means of attacks on cyber-physical systems are essentially similar to those used to target information technology and communication ssystems. However, the goals of cyber-physical attacks and the propagation of their effects are considerably different. The analysis – and ultimately the understanding – of attacks on cyber-physical systems depends on the ability to describe the attacks in a systematic and comprehensive manner.

According to Byres and Lowe [5], attacks on industrial control systems and critical infrastructures assets can be traced as far back as 1995. Currently, the most famous attack is Stuxnet [1, 4]. Discovered in 2010, it supposedly operated undetected for more than three years [20]. The most notable aspect of the Stuxnet attack is the fact that it inflicted physical damage to an industrial infrastructure (i.e., uranium hexafluoride centrifuges) via the cyber domain. The March 2000 attack on Maroochy Water Services in Queensland, Australia is another prominent example of an attack on an industrial infrastructure. The attack disrupted pumping operations and suppressed alarms, resulting in the release of untreated sewage into local waterways [9]. The possibility of similar cross-domain attacks on modern automobiles has been reported. Several researchers (see,e.g., [2, 8]) have shown that elaborate cyber-attacks can lead to physical consequences, including disabling the brakes of an automobile, killing the engine while the automobile is moving at high speed, permanently locking the doors and manipulating the speed indicator. Other researchers [3, 21] haved emonstrated the ability to compromise quad-rotor unmanned aerial vehicles (UAVs) and microsatellites.

Huang et al. [6], emphasize that attacks on industrial infrastructures can have economic consequences. Moreover, the attack consequences can be am-

plified by the interdependencies existing within a single cyber-physical system as well as those existing between multiple cyber-physical systems. Rinaldi et al. [7], specify four types of interdependencies: physical, cyber, geographical and logical. Because of the interdependencies, the effects of anattack can propagate through multiple domains and inflict secondary damage to other cyber-physical systems and infrastructures. Specifically, attacks on cyber-physical system – even attacks executed in cyberspace – can cross domain boundaries, propagating and amplifying the effects in the domains and causing damage in multiple domains.

This paper describes the Cyber-Physical Attack Description Language (CP-ADL), which is based on a taxonomy specified in [12]. The language can express conventional cyber attacks as well as cross-domain attacks on cyber-physical systems. CP-ADL provides a structure for describing a variety of attacks, an important prerequisite for qualitative and quantitative analyses of attacks on cyber-physical systems. These analyses provide valuable knowledge and understanding of the structural properties and probabilities of attacks. Furthermore, the analyses can help identify the degrees to which functionally equivalent architectural elements are vulnerable to various types of attacks. As such, the resulting knowledge and understanding are vital to improving cyber-physical system security and dependability.

2. Related work

In previous work [12], we analyzed the sufficiency of several cyber security taxonomies for describing attacks on cyber-physical systems. Because cyber security focuses only on attacks that execute in and influence the cyber domain, these taxonomies are unable to express cross-domain effect propagation that is characteristic of attacks on cyber-physical systems. To address this deficiency, we created a novel six-dimensional taxonomy for describing cross-domain attacks on cyber-physical systems; Section 3 provides a brief overview of this taxonomy. Since the current knowledge and understanding of cyber-physical attacks are somewhat limited, the taxonomy only defines the dimensions (i.e., the aspects to be described), not the values corresponding to the dimensions. This approach has, in fact, been adopted in the cyber security domain. An example is the taxonomy of Hansman and Hunt [10], which supports structured human-readable descriptions of newly discovered attacks and is used by major entities such as US-CERT.

Although taxonomies specify structures and, in some cases, support elements of the structures, the definition of a description language based on a taxonomy can be a challenging task. The primary purpose of a description language based on the taxonomy defined in [12] is the structured expression of human-readable attack descriptions. Therefore, the description language should support variable-length descriptions in every dimension.

The importance of a description languagegoes beyond the mere specification of a data format for a taxonomy. Especially important is that a description language provide the capability to express metadata such as the relationships between the elements of various dimensions. As will be discussed in Section 3, this is a critical property for describing attacks on cyber-physical systems, especially if multiple elements must be specified for every dimension of anattack step.

As in the case of taxonomies, the absence of cross-domain considerations in cyber attack description languages hinders their application to the cyber-physical domain. Nevertheless,these languages can provide valuable guidance in developing CP-ADL. Of special interest are the language used to specify US-CERT alerts [15] and the Common Vulnerabilities and Exposures (CVE) description language [14] used in the National Vulnerability Database [13]. Both the languages uniquely identify attacks and describe them in the form of human-readable free text that is separated into semantically distinct sections. A US-CERT alert contains sections that provide the affected systems, attack overview, description, impact, solution, references and revision history. The CVE format provides a structured means to exchange information about security vulnerabilities [16]; a CVE description includes the standard identifier number with a status indicator, a brief description and references to related vulnerability reports and advisories. The Open Vulnerability and Assessment Language (OVAL) uses the publicly released vulnerabilities identified in the CVE list as the basis for most vulnerability definitions [17, 18]. An important point is that all the languages listed above as well as other related languages use data formats such as text and XML that support variable-length elements (see, e.g., the NVD-CVE feed schema [19]).

3. Taxonomic foundation for language design

The term taxonomy has many meanings in the research literature (see, e.g., [12]). This work considers a taxonomy to be a specification of seman-

tically distinct aspects that are to be described. This type of taxonomy is perfectly suited to establish a basis for an attack description language that, additionally, provides the means to specify metadata information (i.e., relationships between elements of different taxonomy dimensions).

The basis for the Cyber-Physical Attack Description Language (CP-ADL) is the six-dimensional taxonomy introducedin [12]. The original goal of the taxonomy was to specify properties that should be expressed in descriptions of attacks on cyber-physical systems, including cross-domain attacks. However, it was subsequently discovered that the same structure can be used to describe countermeasures against cyber-physical attacks, which also exhibit cross-domain properties. Therefore,the taxonomy dimensions were renamed to enable them to describe attacks as well as defensive measures [21] (see Fig. 1).

At this point, it is important to discuss the rationale for the selected taxonomy dimensions. Attacks and defensive measures can be described in terms of an Action (i.e., execution of some Method), the success of which depends on the fulfilment of one or more Preconditions. The execution of a Method introduces one or more changes in a system; the changes can be immediate changes (described using the Cause dimension group) and follow-up changes (described using the Effect dimension group).The distinction between these two dimension groups reflects the fact that, due to the complex dependencies and interdependencies existing in cyber-physical systems, any change in a cyber-physical system can result in the propagation of Effects. Note that Effects are also changes in a system, which can induce further consequences. Section 5 discusses how CauseEffect causality chains can be described using CP-ADL.

A distinguishing feature of the taxonomy is the clear distinction between Influenced Element and Affected Element. Because these dimensions are independent of each other, an element of one dimension can belong to the cyber or physical domain regardless of the domain affiliation of the other element. This makes it possible to describe cross-domain attacks.

Based on the domains of elements, it is possible to define four attack categories: (i) cyber-to-cyber (C2C); (ii)cyber-to-physical (C2P); (iii) physical-to-physical (P2P); and (iv) physical-to-cyber (P2C). C2C attacks have been studied extensively by the cyber security community. The material science community has traditionally focused on aspects of P2P attacks (e.g., physical component wear due to speed, temperature and vibration). P2C effect propagation has been studied by the computer dependability and embedded

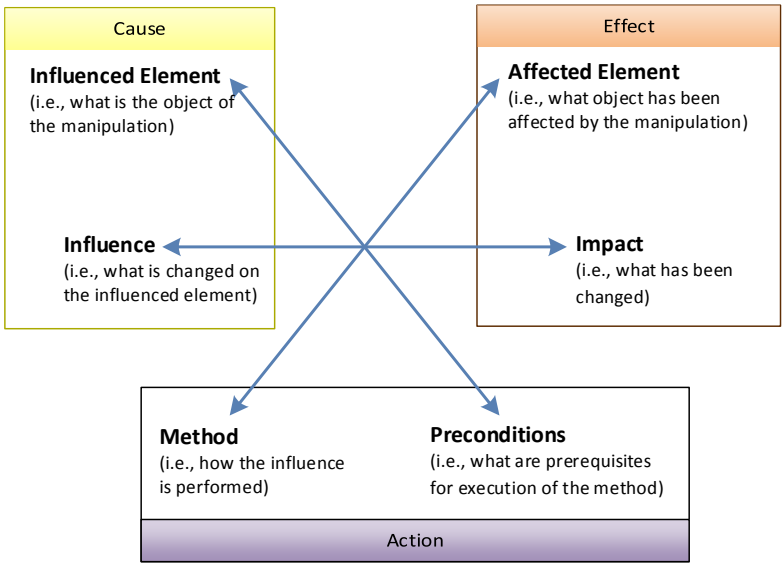

Figure 1: Taxonomy of Cyber-Physical Attacks [12]

systems security communities (e.g., side-channel attacks on embedded systems). However, the C2P category of attacks has barely been investigated as of this time.

Cyber-physical systems are exposed to attacks belonging to all four categories. Moreover, complex attacks can manifest properties of multiple categories at the same time. For instance, complex attacks usually involve multiple stages, each of which could belong to a different category. Furthermore, simultaneous attacks of different categories are also possible.

The taxonomy has been used to describe attacks encountered in several case studies. These includethe Stuxnet attack on Iran's uranium hexafluoride centrifuges [4], attacks on a modern automobile [2], attacks identified during a vulnerability assessment of a quad-rotor UAV [3] and attacks on a microsatellite with a propulsion system [21]. In all these case studies, the proposed structure was able to express all the relevant aspects of conventional cyber attacks and cross-domain cyber-physical attacks. However, the need to express metadata alongside the properties specified by the taxonomy dimensions was also recognized. In particular, it is vital to express the complex relationships that exist between the different dimensions of a single attack (see Fig. 2) as well as the relationships that exist between different attacks. These diverse concepts can only be expressed by a description language that

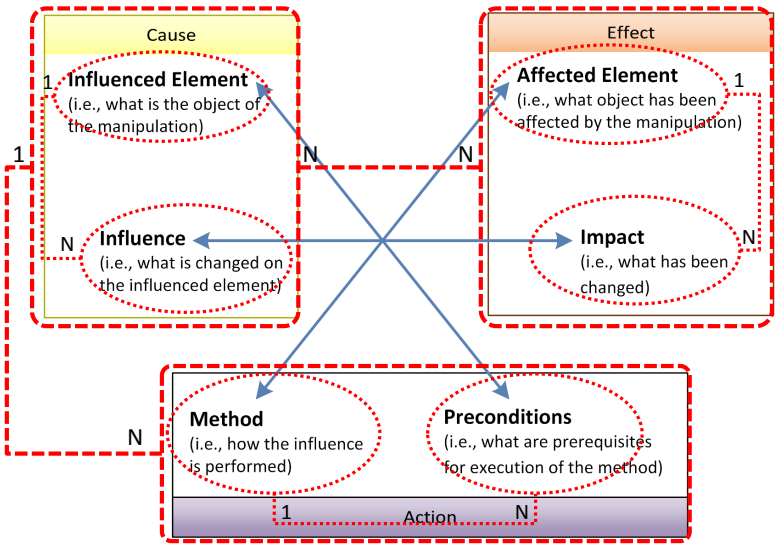

Figure 2: Cardinality relationships between the taxonomy dimensions

is founded on the taxonomy.

4. CP-ADL: Cyber-Physical Attack Description Language

In this section we introduce the *Cyber-Physical Attack Description Language* (CP-ADL). CP-ADL is a natural extension of the taxonomy defined in [12]. Therefore, this language supports description of both conventional cyber as well as cross-domain attacks on CPS. Additionally, CP-ADL incorporates the capability to specify medatada information, and thus overcomes the taxonomy limitations outlined in Section 3.

In this section, we first define the CP-ADL structure and then outline how a XML data format can be defined upon this structure. Please note that while our explanations will mainly emphasize the description of attacks, the same is applicable to the description of defensive measures.

4.1. CP-ADL structure

After considering various options, we have decided to pursue the cause-centric language definition. In the case of attack or defense measures, this is the element immediately influenced by the action. Our main reasons for this decision are as follows:

- Every CPS model consists of various components and their interconnections. Both CPS components and their interconnections can be seen as potential Influenced Elements affected, e.g., by various attack Methods. Therefore, such centricity should simplify the vulnerability analysis performed via traversing interconnected CPS components.

- Such centricity is naturally suitable for building a knowledge base of vulnerabilities, attacks, and consequences of attacks on various targets. This knowledge base can then be used in vulnerability analysis approaches traversing cause-effect chains associated with the CPS elements. Furthermore, the knowledge base can be easily extended if a new Method to Influence some CPS component is discovered.

- The knowledge base about target vulnerability can become very useful for consideration of defense measures. This is especially true in the case of alternative functionally equivalent architectural solution. Designers of a new CPS should be capable to select one of such solution based on their non-functional properties, such as costs, weight, but also robustness and resilience against various kinds of attacks.

The presentation of the CP-ADL structure is arranged as follows. First we discuss the relationships between dimensions within groups. After that we explain the relationships between these dimension groups. Formally, the structure is introduced in both BNF form commonly used for the language definitions as well as in form of semantically equivalent UML class diagrams. We have decided to duplicate definitions in order to address readers with different backgrounds.

4.1.1. Dimension group "Cause"

It is reasonable to assume that every single attack targets only a single (Influenced Element, Influence) tuple. Indeed, this may be true in the majority of the cases. Nevertheless, description of some attacks (e.g., correlation of information from multiple sources) will require possibility to instantiate multiple (Influenced Element, Influence) tuples at the same time. Therefore, we define the group Cause as follows (the corresponding UML diagram is depicted in Figure 3):

```
<Cause> ::= (<Influenced Element>, <Influence>)
       [AND (<Influenced Element>, <Influence>)]*

<Influenced Element> ::= (<Category>, <Name>)
<Influence> ::= (<Type>, <Description>)
```

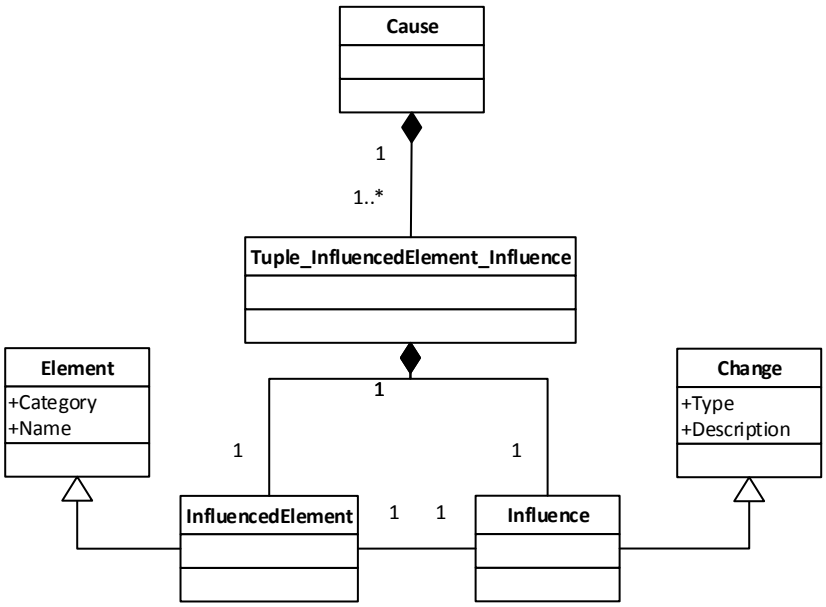

Figure 3: Dimension group "Cause"

The presence of the Tuple_InfluencedElement_Influence class is the only true difference between the BNF form definition and the corresponding UML class diagram. This class is used to represent the 1:1 relationship between Influenced Element and Influence in a single tuple. However, the number of such tuples is unbounded.

Please note that we have imposed no restrictions how often the same Influenced Element may appear in the description of the same attack or defense measure. Therefore, alongside with the description of various influences on different elements, it is possible to describe simultaneously executed multiple influences on the same element. In order to do this, multiple (Influenced Element, Influence) tuples can be specified with an identical Influenced Element.

In the UML diagram, we have derived Influenced Element and Influence classes from Element and Change classes, respectively. As we will see later

in this section, we will reuse these base classes also for the definition of the classes representing dimensions of the Effect group.

Finally, we would like to discuss the refinement of Influenced Element and Influence. Compared to the taxonomy (see Section 3), both contain a slight refinement. Influenced Element can be described in terms of element's Category and Name; Influence can be specified as its Type and Description. Form the theoretical perspective, this refinement is not crucial because it does not change the semantics. However, from the practical perspective it can bring several benefits. Category can simplify the distinction between Elements in cyber and in physical domains. Similarly, Type can simplify the distinction between changing the state, gaining knowledge, or having knowledge of the corresponding Element. We intentionally specify no restrictions on how Category and Type should be used or structured. Therefore, experts using CP-ADL should be able to choose notations and structures most suitable for the purpose of their work and/or reflecting specific constraints.

4.1.2. Dimension group "Effect"

The dimension group Effect can be seen as the counterpart of the group Cause. Both reflect changes in the system, former caused by an Action and later triggered by the Cause. Therefore, their structure is essentially the same and can be defined as follows (the corresponding UML diagram is depicted in Figure 4):

```
<Effect> ::= (<Affected Element>, <Impact>)
           [, (<Affected Element>, <Impact>)]*
<Affected Element> ::= (<Category>, <Name>)
<Impact> ::= (<Type>, <Description>)
```

In general, a single change to the system, e.g., caused by an attack, can cause multiple effects. Therefore, we define the Effect group as one or more (Affected Element, Impact) tuples. As in the definition of the Cause dimension group, it is possible to specify multiple Impacts on the same or different Affected Elements.

4.1.3. Dimension group "Action"

Even considering cyber security, the same manipulation described in the Cause dimension group, e.g., escalation of user privileges, can be realized by using different Methods. In the case of CPS, the multiplicity of means for the same manipulation is even bigger. It is because the same manipulation

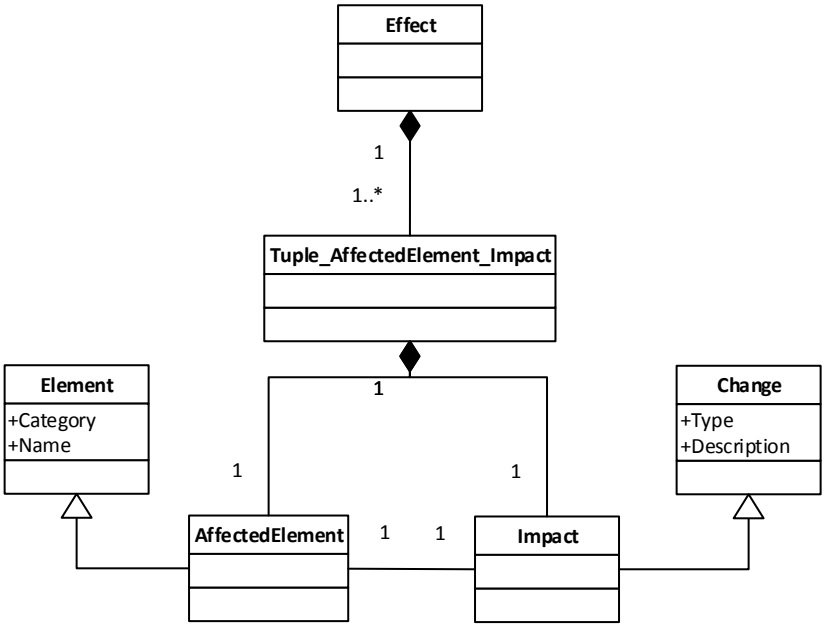

Figure 4: Dimension group "Effect"

often can be done by both cyber and physical means. Therefore, it should be possible to describe various alternative Methods.

At the same time, certain Methods can only be executed and/or be successful if multiple Preconditions hold. For instance, it can include simultaneous presence of several vulnerabilities, e.g., the absence of W⊕X protection in conjunction with the unguarded buffer boundaries.

Consequently, we defined the dimension group Action as follows (see Figure 5 for the corresponding UML diagram):

```
<Action> ::= (<Method>, <Preconditions>)
          [OR (<Method>, <Preconditions>)]*

<Method> ::= (<Category>, <Description>)
<Preconditions> ::= <Precondition>
                [AND <Precondition>]*
<Precondition> ::= (<Category>, <Description>)
```

There are several syntactical differences between the definitions in BNF form and as UML class diagram. We have introduced class Preconditions, Tuple_Method_Preconditions as a means to represent $1:1$ relationship be-

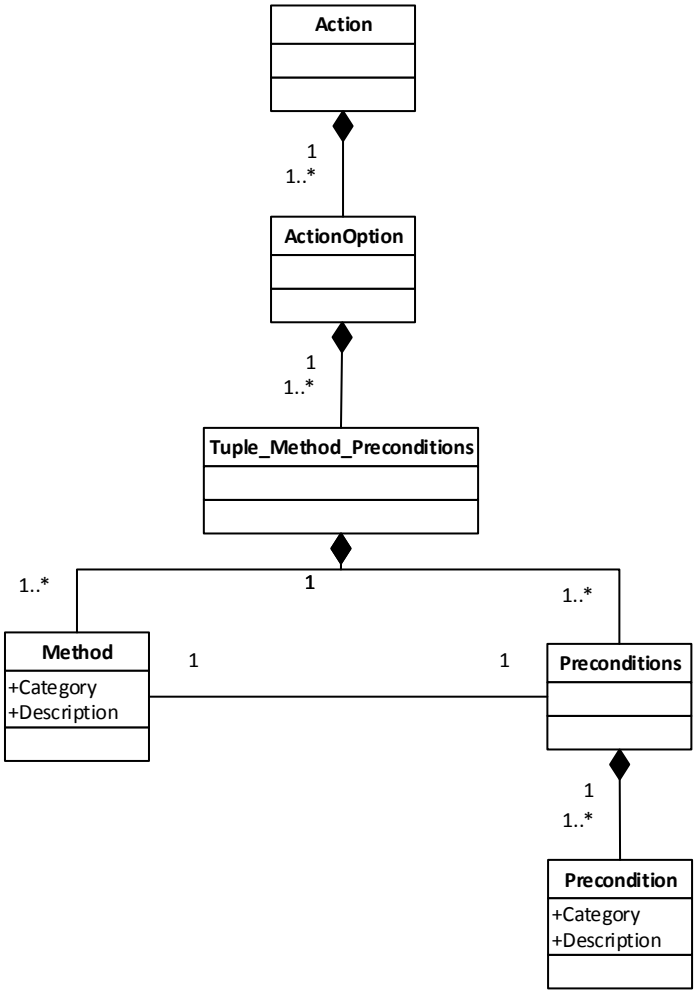

Figure 5: Dimension Group "Action"

tween a particular Method and corresponding Preconditions. In BNF form, the relationship between these tuples is explicitly defined as OR. Further, class Preconditions has been introduced in UML diagram in order to reflect AND-composition of one or more Preconditions associated with a particular Method.

Similar to the refinement of dimensions described previously, we introduce refinement of Method and Precondition. Both can be specified in terms of their Category and Description.

4.1.4. Structural and temporal relationships between dimension groups: Atomic Attack, Attack Chain, and Attack Chain Folding

The relationship between various dimension groups is slightly more complicated. In the simplest case, a manipulation of a single property of a single element (described as the Cause dimension group) performed by an attack or defense Methods (described within the Action dimension group) leads to the desired change of a single property of the same or a different single element (described as the Effect dimension group). However, it should not be always the case. Especially in the case of attacks on complex systems (e.g., systems with redundancies and fallback capabilities) only simultaneous manipulations performed on multiple elements will lead to the desired Effect(s). Therefore, an attack description language should be able to express simultaneous Influences on the same or different Influenced Elements, i.e., one or more tuples described as part of Cause dimension group, whereby each Cause should be associated with the corresponding Actions. We call such simultaneous manipulations (including the resulting Effects) an *Atomic Attack*.

Further, Effects themselves are changes introduced to the system. Therefore, similar to the Cause, these changes can trigger follow-up Effect(s). We will refer to such auto-induced effect propagations as *Causal Chain*. A Causal Chain can be described as a sequence of Atomic Attacks. The reusing (Affected Element, Impact) tuples of one Atomic Attack as (Influenced Element, Influence) of the follow-up one enables value-based linking of two or more Atomic Attacks in a Causal Chain of effect propagations. In the description of the induced effect propagation chain, specification of Method (member of Action dimension group) can be omitted. Preconditions can be used to express constraints under which the described follow-up Effects will occur.

Considering Causal Chains, it is important to understand that not all intermediate stages of effect propagation are of relevance for the security

analysis. For example, in the buffer overflow attack, depending on the selected level of abstraction, the Influenced Element can be either process buffer or the process themselves. Depending on the injected payload, the attacked process will either crash or execute some malicious code. In the case of CPS, the selection of relevant level of abstraction can be a very challenging task. Therefore, the capability to fold the intermediate steps of effect propagation chains, i.e., to hide irrelevant details without losing them, becomes an important property of an attack description language. We will discuss the folding alongside with the linking of attack descriptions in Section 5.

Concluding the above discussion about Atomic Attacks, Attack Chains, and attack Folding, we define these as well as the relationships between the dimension groups as follows (the corresponding UML diagram is depicted in Figure 6):

```
<CausalChain> ::= { <AtomicAttack>
                    [, <AtomicAttack>]* }

<AtomicAttack> ::= ((<Cause>, <Action>)
                  [AND (<Cause>, <Action>)]*, <Effect>)
                  [OR <Folding>]

<Folding> ::= <CausalChain>
```

Note that in the UML diagram we have introduced the Tuple_Cause_Action class in order to preserve clear relationship between different dimension groups. Despite the slight syntactical differences, the presented UML diagram and the BNF form remain semantically equivalent.

In both, BNF definition and UML diagram, a Causal Chain consists of one or more Atomic Attacks. At the same time, the Atomic Attack can be seen as a Folding of the more detailed attack description presented in Causal Chain. In order to reflect this, in BNF form AtomicAttack has an optional element Folding which is, in turn, CausalChain. This optional element enables recursive definition of details of the folded attack description. In the UML diagram, this recursion is defined as an aggregation relationship between AtomicAttack and CausalChain classes.

4.2. CP-ADL XML data format

The complexity of the presented CP-ADL structure, especially of the relationships between elements belonging to different taxonomy dimensions, reflects the complexity and diversity of attacks on CPS. For the storage of

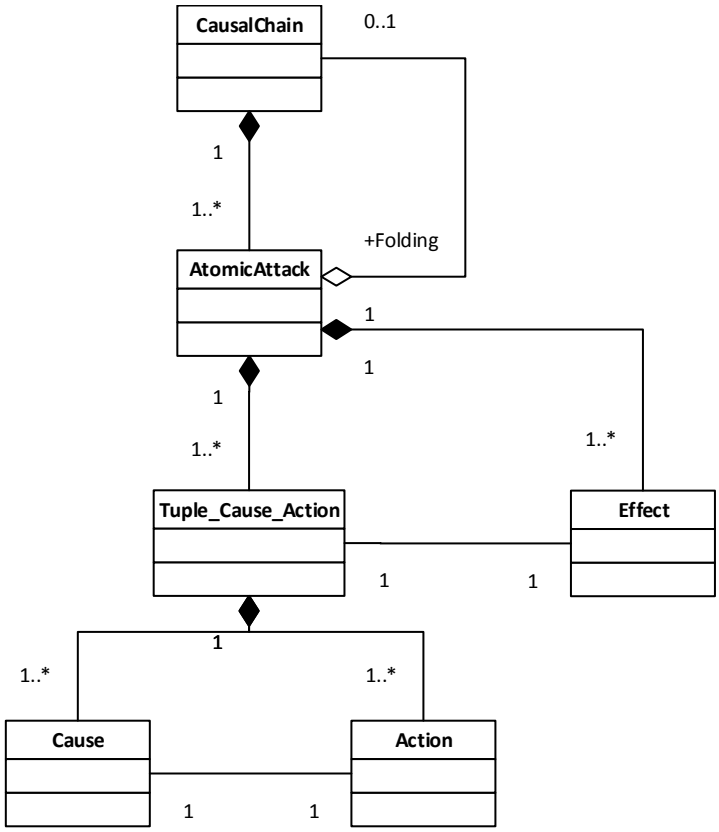

Figure 6: "Atomic Attack" and "Causal Chain"

attack descriptions and/or for the exchange of attack descriptions between various tools, we need to define a data format upon the proposed CP-ADL structure. Because of this complex relationships between dimensions, variable amount of elements to be described, and variable size of any particular element[1], we see XML as the best suitable basis for the definition of the CP-ADL data format.

A XML-based data format for CP-ADL can be derived directly from the presented UML diagrams. As this step is rather trivial, we will omit here the presentation of the format definition. Instead we would like to outline several additional optimizations, which we would introduce by this step. First of all, we would introduce a top-level "wrapper" element, e.g., named CPADL, which would act as a container for the descriptions of one or more attacks. This element should contain a version member variable, in order to ensure the backward compatibility despite the expected evolution of this data format. Furthermore, compared to the structure definition, in the XML schema we would introduce ID fields for the Atomic Attack and Attack Chain. These should enable reusability of the attacks descriptions via their referencing in the description of more complex attacks. In Section 5 we will discuss the usage of IDs in attack linking and folding.

5. Linking and folding attack descriptions

Figure 7 illustrates the encapsulation of various defined CP-ADL elements. This simplified view illustrates that every Causal Chain consists of one or more Atomic Attack descriptions, each of which specifies single change to the system described at the level of abstraction selected by an analyst. In this section we will explain how these CP-ADL elements can be used to express various relationships between different attack descriptions.

5.1. Attack propagation

We first show how various types of relationships between different attacks or attack steps can be represented with CP-ADL.

5.1.1. Linking between Effect and Cause within single Causal Chain

We should distinguish between cascading (avalanche-like) and triggered attack effect propagations. In Section 4.1.4 we have defined such cascading

[1]This is especially due to the intended support of human-readable descriptions.

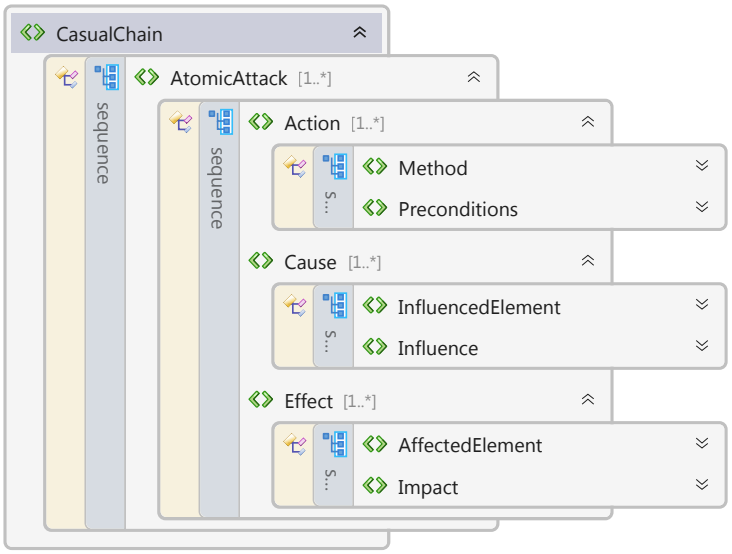

Figure 7: Encapsulation of CP-ADL elements, simplified view

effect propagations as Causal Chain. A straightforward example of a causal chain is the Stuxnet's cross-domain attack. Simplified[2], when triggered by a combination of conditions, the Stuxnet payload installed at PLC sends to the frequency converter drives commands to change the maximum frequency at which centrifuges rotate. Describing this attack in CP-ADL, this whole action can be described as a single Atomic Attack. Preconditions are that the payload is installed and triggered, Method is the description of the specially crafted command sequences sent via Profibus, Influenced Element is the frequency converter drive, Influence is the change of the designated rotation frequency, Affected Element are the centrifuges under the control of the frequency converter drive, and the Impact is the rotation of centrifuges with the speeds outside of the operational range they have been designed for as well as the frequent change of the speed.

The rotation of the centrifuges with the speed outside of operational range is, in turn, the Cause of the follow-up Effect, increased wear, reduced life time, and finally physical damage of the affected centrifuges. This can be described as a second Atomic Attack with the (Influenced Element, Influence) tuples reusing one or more (Affected Element, Impact) tuples described in the first

[2]A detailed description of the Stuxnet attack can be found in [4].

Atomic Attack. As mentioned before, there is no need to specify Method in the second Atomic Attack description. Preconditions expressed in the second Atomic Attack might reflect the fact that the specified Effect only occurs if the conditions described in Cause are applied over sufficient amount of time.

In order to reflect the tight cause-effect relationships between these two Atomic Attacks, they can be described within a single Causal Chain section. Please note that, even described as a part of the same Causal Chain, the first Atomic Attack is of C2P (i.e., Cyber-to-Physical) and the second one of P2P (i.e., Physical-to-Physical) type. This illustrate the fact that attacks on CPS can cross the domain boundary and that the effect can propagate in a different domain than the one in which the attack was initiated.

5.1.2. Linking between Effect and Cause in different Causal Chains

Indeed, it seems to be very natural to describe the effect propagation initiated by an attack within a single Causal Chain. However, we think that it is not always the most optimal solution. Especially if different attacks can lead to the same Influence on Influenced Element it might be more reasonable to describe the Causal Chain containing the effect propagation as a separate attack. We see two major advantages of this. First, this will reduce the size and thus improving the human readability of the XML file containing multiple attack descriptions. Second, and most important, it will allow explicitly depict relationships between different attacks and consequences thereof, thus improving the ability to analyze the system and/or attack properties.

We would like to motivate such separation on one slightly more complex example. Let us assume that an adversary has two distinct means, one of which disables the heat dissipation system and another one increases the heat production in a particular part of a CPS. Both these scenarios can be described as two distinct Causal Chains. The Effect of both these attacks will contain identical (Affected Element, Impact) tuples, e.g., describing the raising temperature in this part of CPS. This tendency, in turn, can lead to the situation when the temperature raises beyond the operational range of some Electronic Control Unit (ECU) located in this part of the system. This, in turn, can lead to the reduced performance or even crash of the affected ECU. Depending on the role of the ECU in the CPS, this can lead to further consequences to the whole system and, potentially, to its environment. The outlined transition from the temperature raising tendency, through breaking thresholds, and up to the resulting consequences can be described within another separate Causal Chain. Therefore, linking across Causal Chains

enables reusability of the same effect propagations (described in a dedicated Causal Chain) instead of replicating the same description multiple times.

Please note that, similarly to the linking between Effect and Cause within single Causal Chain, the relationship is established based on the matching of corresponding values. We also would like to discuss the cardinality relationships between linked Causal Chains. As described above, the same Causal Chain can be triggered by different attacks, each of which can be described in numerous other Causal Chains. However, similar argumentation applies to the opposite direction as well. As every attack can produce numerous (Affected Element, Impact) tuples describing Effect of an attack, numerous combinations of these tuples can be used as "triggers" for the effect propagations described in different Causal Chains. Therefore, the cardinality relationship between linked Causal Chains is $N : N$.

5.1.3. Linking between Effect and Preconditions

Another kind of logical relationships is representative for complex attacks executed as a sequence of multiple stages. In the car case study [2], an infected ECU spreads infection in two stages. In the first stage, it sends to a target ECU a request to enter the reprogramming mode. As no Authentication and Authorization (AA) protection mechanisms are implemented, the target ECU enters this mode. This enables the second stage, in which the firmware of the ECU is reprogrammed with a malicious code. From the attack propagation perspective, the Effect of the first stage enables certain Preconditions required for the second stage of this attack.

We would like to emphasize the cardinality relationship between attack descriptions linked this way. In general, multiple different attacks can produce to the same Effects. The same Effect(s) can enable Preconditions for numerous follow-up attacks. Further, some attacks can require that a combination of multiple Preconditions is satisfied at the same time. Each of these Preconditions can be enables by number of independent (and/or independently described) attacks. Therefore, the relationship between the Causal Chains producing Effects and the Causal Chains reusing these Effects as Preconditions is $N : N$.

5.2. Attack encapsulation

We will now present how an attack descriptions can be reused and how multiple attack descriptions can be organized in hierarchies.

5.2.1. An attack as Method of another one

In [2] has been shown that the engine will stop if the ECU controlling it enters the reprogramming mode. This will ultimately cause that the car will stop. Most critical to safety, this attack can be successfully executed even during the car is driving at a high speed. It is self-evident that, if this attack is executed on a highway, it can lead to a severe car accident.

Please note that, in this case, entering the reprogramming mode is the Method by which attack on the engine's ECU is performed. Of course, it is possible to describe the entering reprogramming mode attack in all details and then link the Effect and Cause values of the effect propagation chain in the description of a single Causal Chain. However, we see two major disadvantages of this approach. First, it will lead to the replication of the same description multiple times. Such multiplicity of descriptions, especially if performed manually, will inevitably lead to multiple different variants of the description, thus preventing automation of the syntactical correlation between attacks' descriptions needed for the analysis. Second, this will lead to overflowing of an attack description with many details, which logically belong to different abstraction layers. As a consequence, the analysis of information can become a more demanding task.

Instead, we propose to support in the CP-ADL language the possibility to describe Method as a reference to another, already described attack. For this purpose, the Category element of Method can be set to a predefined value signaling that the Description element contains an ID of either an Atomic Attack or a Causal Chain. The Category value can signal whether used ID is the Atomic Attack ID or the Causal Chain ID.

This approach would mean that we introduce an unidirectional association between classes CausalChain (see Figure 6) and Method (see Figure 5). The cardinality of this association is $1 : N$ because the same (more fundamental) attack can be reused as Method in multiple different (more complex) attacks. Please note that this definition is recursive.

Advantages of this language extension are following. Connections between different attacks will be explicitly expressed and only information at the relevant abstraction layers will be described, thus simplifying the analysis of properties of attacks on CPS significantly. Furthermore, this enables description of various Effects, i.e., consequences of an attack, under different Preconditions. For instance, an attack intended to kill the motor of a car (as described previously, in [2] it is done by the Method of commanding the

motor ECU to enter the reprogrammable mode) is only safety-critical under certain Preconditions, e.g., driving at a high speed.

5.2.2. Hiding attack details

During both attack description and attack analysis, an expert has to deal with the selection of an appropriate level of abstraction. By this decision, the tradeoff is between the ability to abstract away the unnecessary details, thus focusing on important aspects only, and the ability to describe all details, which might be relevant for further investigations. In such situations, it is a common best practice to provide a possibility to hide (or to fold, blind out) irrelevant details without losing the information.

In CP-ADL structure, we have incorporated the possibility to express a relationship between descriptions of the same attack at different levels of abstractions. This possibility is defined as aggregation relationship between classes AtomicAttack and CausalChain (see Figure 6). Implementing this aggregation in the CP-ADL XML schema, the straightforward decision would be to incorporate CausalChain block describing attack details as a part of the AtomicAttack block. However, this would have disadvantages similar to those described in Section 5.2.1 for the incorporation of an attack description as Method in the description of another attack. Therefore, we prefer following alternative. The folding of attack details can be specified as an association between two separately defined Causal Chains. In the definition of CP-ADL XML schema, this association can be realized as an optional attribute of the AtomicAttack element, which points to the Causal Chain containing attack details.

We would like to illustrate the usefulness of the attack folding on the example of the attack on the engine's ECU as it was described in [2]. The Causal Chain containing the detailed description of the attack would start with the Atomic Attack targeting the engine's ECU in the form of requesting it to enter reprogrammable mode. As a consequence, the process running on this ECU stops. As a consequence thereof (described in the follow-up Atomic Attack within the same Causal Chain) the motor stops. The subsequent Atomic Attacks would describe further steps of effect propagation, such as stopping of the wheels caused by the motor stop, and finally stopping of the car. It is clear that such detailed description can be extremely lengthy. More importantly, such detailed description can contain information irrelevant for the security analysis. For instance, for the consideration of environmental implications it might be interesting to describe an Atomic Attack containing

entering ECU into reprogramming mode described as Cause and the resulting – after numerous of effect propagation steps – stopping of the car described as Effect. Such Atomic Attack description would be fully sufficient for analysis and description of a Causal Chain, in which an analyst focuses on the consequences of the car's stopping, e.g., possible collisions with the objects in the car's environment, which, in turn, would lead to the physical damage, injuries, legal and/or social consequences, etc.

Please note that, different from the previously discussed examples of linking between attack descriptions, we have defined folding as a $1:1$ relationship. However, this operation is recursive and multiple layers of folding can be described. Folding is also an optional component, thus enabling termination of more detailed description at any necessary level of abstraction. Furthermore, Effect(s) described in the folded Atomic Attack should not necessarily be identical to Effect(s) of the last Atomic Attack in the Causal Chain describing the same attack in more details. Instead, it can be a combination of any (Affected Element, Impact) tuples of one or more Atomic Attacks within the Causal Chain describing attack details. This enables the analysis- and analyst-dependent selection of relevant tuples.

6. Application areas

We see several application areas of the proposed CP-ADL. In this section, we will outline the three, in our opinion, most important ones: structured documentation of known attacks on CPS, qualitative and quantitative analysis of known attacks on CPS, and support of the CPS vulnerability analysis.

Structured attack description is a prerequisite for many other activities, including attack analysis. It will ensure that all information aspects needed for the analysis are documented. At the same time, it will prevent description of the irrelevant information, thus simplifying and speeding up various types of attack analysis.

Qualitative and quantitative attack analysis should reveal structural attack properties as well as their frequency. In its simplest form, a comparison of two attack description should be able to determine whether these are equivalent or not. If they aren't, which properties distinguish them from each other. Applied to the descriptions of already known attacks, it should be possible to identify whether the described attack

is a principally new one or just an already known attack applied to yet another CPS or CPS component. As a result, it should be possible to identify and to document a variety of distinct attack aspects. We expect that the qualitative analysis will result in the consolidation of the (currently – free-form textual) description into a more formal tree-like taxonomies. Such taxonomies should greatly simplify the analysis of attacks, foster collaborations across multiple research teams, and ensure the interoperability across various tools. Furthermore, such taxonomies will enable the automated attack analysis. The quantitative analysis will provide insights regarding frequency of various attack methods or selection of particular element as targets of manipulation. Both qualitative and quantitative analysis will provide information necessary for the vulnerability analysis.

Vulnerability analysis is a common approach used to improve the system's security. The insights gained in the qualitative attack analysis can be used to discover vulnerabilities existing in the system. The frequency of particular kinds of attacks, provided by the quantitative analysis, multiplied by the estimated costs of the successful attack provides a value commonly used to rank between different vulnerabilities. Such ranking a necessary prerequisite for a cost-effective decision making regarding which of the identified vulnerabilities should be mitigated. It has been shown in computer and network security that such vulnerability analysis can be successfully automated. This, in turn, can result in better scalability and objectivity of vulnerability analysis, what ultimately will improve the security properties of CPS.

7. Conclusions

The Cyber-Physical Attack Description Language (CP-ADL) described in this paper builds on a taxonomy designed to capture cyber-physical attacks [12]. The language supports structured descriptions of conventional cyber attacks as well as novel cross-domain attacks on cyber-physical systems. Furthermore, the language permits the expression of six semantically different aspects (dimensions)of attacks and the complex relationships that exist between them. The language also supports the specification of relationships existing between attack descriptions, including causal chains of effect propagation, relationships between attack stages and descriptions of attacks at multiple levels of abstraction.

One of the key advantages of CP-ADL is its structure for describing attacks discovered by different research groups. This structure is an important prerequisite for qualitative and quantitative analyses of attacks on cyber-physical systems. Such analyses provide knowledge and understanding of the structural properties and probabilities of cyber-physical attacks. Furthermore, they can help identify which one of several functionally equivalent architectural solutions is more or less vulnerable to specific attacks. The resulting knowledge and understanding are crucial to enhancing cyber-physical system security and dependability.

Future research will focus on the development and population of a knowledge base containing known attacks on cyber-physical systems. This knowledge base, intended to provide a foundation for attack analyses, will introduce new challenges and requirements that will contribute to the advancement of CP-ADL. One area of enhancement is the specification of relationships between descriptions of different attacks. The enhancement would reflect aspects such as generalization and/or specialization of attack descriptions at multiple abstraction levels and also indicate the semantic equivalence of syntactically different descriptions.

8. Acknowledgement

This research was conducted while Dr. Yampolskiy and Dr. Horváth were at Vanderbilt University, Nashville, Tennessee.

References

[1] Albright, D., Brannan, P., Walrond, C. 2010. Did Stuxnet Take Out 1,000 Centrifuges at the Natanz Enrichment Plant? Institute for Science and International Security, Washington, DC (url: http://isis-online.org/uploads/isis-reports/documents/stuxnet_FEP_22Dec2010.pdf), 2010.

[2] Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., Savage, S., Experimental security analysis of a modern automobile, in: Proceedings of the IEEE Symposium on Security and Privacy (SP), 2010, pp. 447-462.

[3] Yampolskiy, M., Horvath, P., Koutsoukos, X. D., Xue, Y., Sztipanovits, J., Systematic analysis of cyber-attacks on CPS-evaluating applicability of DFD-based approach, in: Proceedings of the fifth International Symposium on Resilient Control Systems, 2012, pp. 55-62.

[4] Falliere, N., Murchu, L. O., Chien, E., W32. Stuxnet Dossier. Version 1.4, Symantec, MountainView, California, 2011.

[5] Byres, E., Lowe, J., The myths and facts behind cyber security risks for industrial control systems, in: Proceedings of the VDE Kongress, 2004.

[6] Huang, Y.-L., Cardenas, A.A., Amin, S., Lin, Z.-S., Tsai, H.-Y., Sastry, S., Understanding the physical and economic consequences of attacks on control systems, International Journal of Critical Infrastructure Protection (IJCIP), 2009, vol. 2, nr. 3, pp. 73-83.

[7] Rinaldi, S. M., Peerenboom, J. P., Kelly, T. K., Identifying, understanding, and analyzing critical infrastructure interdependencies. IEEE Control Systems 21(6), 2001, pp. 11-25.

[8] Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., Savage, S., Koscher, K., Czeskis, A., Roesner, F., Kohno, T., Comprehensive experimental analyses of automotive attack surfaces, in: Proceedings of the Twentieth USENIX Conference on Security, 2011.

[9] Slay, J., Miller, M., Lessons learned from the maroochy water breach. Critical Infrastructure Protection, 2007, pp. 73-82.

[10] Hansman, S., Hunt, R., A taxonomy of network and computer attacks. Computers & Security 24(1), 2005, pp. 31-43.

[11] Lippmann, R. P., Ingols, K. W., Scott, C., Piwowarski, K., Kratkiewicz, K. J., Artz, M., Cunningham, R. K., Evaluating and strengthening enterprise network security using attack graphs, Defense Technical Information Center, 2005.

[12] Yampolskiy, M., Horvath, P., Koutsoukos, X. D., Xue, Y., Sztipanovits, J., Taxonomy for Description of Cross-Domain Attacks on CPS, in: Proceedings of the Second ACM International Conference on High Confidence Networked Systems, 2013, pp. 135-142.

[13] National Institute of Standardsand Technology, National Vulnerability Database, Gaithersburg, Maryland (url: http://nvd.nist.gov).

[14] MITRE, Common Vulnerabilities and Exposures, Bedford, Massachusetts (url: http://cve.mitre.org).

[15] United States Computer Emergency Readiness Team, 2014 Alerts, Washington, DC(url: http://www.us-cert.gov/ncas/alerts), 2014.

[16] International Telecommunications Union, ITU-T Recommendation X.1520, Geneva, Switzerland (url: http://www.itu.int/rec/T-REC-X.1520-201104-I/en), 2014.

[17] MITRE, OVAL Language, Bedford, Massachusetts (url: http://oval.mitre.org/language).

[18] Baker, J., Hansbury, M., Haynes, D., The OVAL Language Specification, MITRE, Bedford, Massachusetts (url: http://ebookbrowsee.net/oval-language-specification-08-08-2011-pdf-d222972411), 2011.

[19] National Institute of Standards and Technology, CVE XML Schema, Gaithersburg, Maryland (url: http://nvd.nist.gov/schema/nvd-cve-feed_2.0.xsd).

[20] McDonald, G., Murchu, L. O., Doherty, S., Chien, E. (2013). Stuxnet 0.5: The Missing Link, Symantec, Mountain View, California, 2013.

[21] Forbes, L., Vu, H., Udrea, B., Hagar, H., Koutsoukos, X. D., Yampolskiy, M., SecureCPS: Defending a Nanosatellite Cyber-Physical System, in: Proceedings of The SPIE Defense and Security Symposium, 9085, 2014.