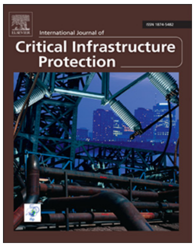
Available online at www.sciencedirect.com

ScienceDirect

www.elsevier.com/locate/ijcip



# Vulnerability modeling and analysis for critical infrastructure protection applications

*Stefano Marrone*[a,*], *Roberto Nardone*[b], *Annarita Tedesco*[b], *Pasquale D'Amore*[b], *Valeria Vittorini*[c], *Roberto Setola*[d], *Francesca De Cillis*[d], *Nicola Mazzocca*[c]

[a]*Seconda Universita di Napoli, Dipartimento di Matematica e Fisica, viale Lincoln, 5, 81100 Caserta, Italy*
[b]*Ansaldo STS, Innovation and Competitiveness Unit, Via Argine 425, Naples, Italy*
[c]*Universita di Napoli Federico II, Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione, Via Claudio 21, 80125 Naples, Italy*
[d]*Faculty of Engineering, Universita campus Bio-Medico di Roma, via Alvaro del Portillo 21, 00128 Rome, Italy*

ARTICLE INFO

*Article history:*
Received 18 January 2013
Available online 14 October 2013

*Keywords:*
Railway infrastructure
Physical vulnerability assessment
UML profiles

ABSTRACT

Effective critical infrastructure protection requires methodologies and tools for the automated evaluation of the vulnerabilities of assets and the efficacy of protection systems. This paper presents a modeling language for vulnerability analysis in critical infrastructure protection applications. The language extends the popular Unified Modeling Language (UML) to provide vulnerability and protection modeling functionality. The extended language provides an abstract representation of concepts and activities in the infrastructure protection domain that enables model-to-model transformations for analysis purposes. The application of the language is demonstrated through a use case that models vulnerabilities and physical protection systems in a railway station.

## 1. Introduction

The impact of the terrorist attacks on September 11, 2011 dramatically underscored the fragility of the critical infrastructure and its importance to modern society. This is especially true of critical infrastructure assets such as railway systems. Indeed, the number of attacks on railway assets during the past decade demonstrates the attractiveness of the infrastructure as a target for criminals and terrorists [6]. The massive crowds, potentially high fatality rates, societal reliance and open and accessible designs are all factors that contribute to the railway infrastructure being considered a soft target for assailants.

Physical protection systems incorporating people, policies and equipment are used to secure critical infrastructure assets from malevolent acts. Despite the increase in threat awareness and published best practices, organizations lack formal approaches for evaluating the effectiveness of decisions regarding the implementation of physical protection systems. Indeed, current assessment practices rely on compliance-based approaches (i.e., presence of appropriate components) and performance-based approaches (i.e., evaluation of the consequences of successful attacks).

This paper describes the results of research conducted under the ongoing EU co-funded project, Methodological Tool for Railway Infrastructure Protection (METRIP) [15], which is focused on developing a decision-making system for physical protection system design. The decision-making system is intended to: (i) suggest the types and dispositions of devices that maximize protection effectiveness; and (ii) help evaluate

*Corresponding author.
*E-mail address:* stefano.marrone@unina2.it (S. Marrone).

1874-5482/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ijcip.2013.10.001

the effectiveness of a given physical protection system against attacks.

A model-driven framework is presented that enables quantitative evaluations of asset vulnerability. The framework is based on a modeling approach that specifies the three main aspects involved in effective physical protection system design [5]: (i) attacks; (ii) assets; and (iii) protection technologies and devices. The approach extends the popular Unified Modeling Language (UML) by applying profiling techniques [13] to express vulnerabilities and protection schemes. Model-to-model transformations [3] are employed to generate formal analysis models from UML artifacts. The framework for critical infrastructure protection vulnerability analysis and modeling (CIP_VAM) satisfies three main requirements that enable its application in industrial settings: (i) the use of domain-specific terminology and concepts; (ii) the use of standardized techniques and tools; and (iii) the ability to strike the right balance between the desired level of protection and the associated costs.

## 2. Motivation

There is a significant shortfall of methods for analyzing and enhancing railway security. With regard to the evaluation of vulnerabilities, a crucial requirement is the classification of attack scenarios. To this end, during the first phase of the METRIP Project, we created a database of criminal incidents and terrorist attacks that occurred worldwide from 1970 to 2011. We analyzed 541 incidents in an attempt to correlate the incidents with the primary features of railways (e.g., number of tracks, daily numbers of trains and passengers, station extensions, and numbers and types of implemented protection systems) [4].

Our analysis revealed that the attacks over the last few years have been more lethal. Starting in 2000, the number of attacks and the number of victims per attack have increased steadily. The findings indicate a change in terrorist tactics with an increased emphasis on killing people as opposed to causing economic harm or destroying iconic monuments. The most commonly used weapon type was a bomb, with suicide bombers accounting for the majority of the victims. Medium to small railway stations were targeted most frequently while the most lethal attacks were perpetrated against larger stations. Cameras were found to be the most commonly used protection system; however, security guards proved to be the most effective at preventing attacks and fatalities.

An interesting finding that emerged from the study was that the selection of security systems was usually more directly related to station attributes than to security requirements. To ensure effective protection, it is important to develop better selection criteria based on limiting vulnerabilities while considering station attributes. The CIP_VAM framework described in this paper considers attacks, threats and protection systems to address this limitation and to apply protection measures more effectively.

## 3. Overview of the approach

Several approaches have been proposed for modeling vulnerabilities and evaluating the effectiveness of security measures. However, the vast majority of traditional models focus specifically on cyber systems or introduce frameworks that can be extended to account for physical protection. For example, LeMay et al. [10] have developed the ADVISE Framework, which employs attack graphs to express and analyze attacker behavior and goals. Similarly, Kotenko and Stepashkin [8] use model checking to conduct cyber security evaluations. However, in the case of physical protection system modeling and evaluation, it is imperative to express all the attributes associated with the underlying framework in a holistic manner.

The proposed CIP_VAM profile facilitates the generation of quantitative models (e.g., Petri nets, Bayesian networks and localization models) for evaluating physical protection system configurations and vulnerabilities. Model-driven engineering and model-to-model transformations are employed very effectively to develop the formal representations.

Fig. 1 shows the METRIP modeling and analysis schema. In general, physical protection system designers and evaluators

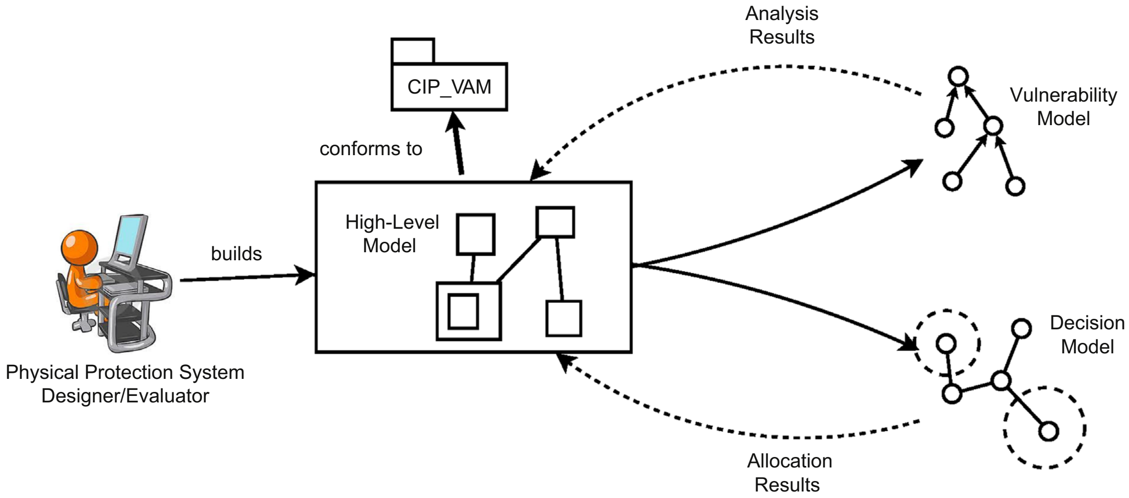


**Fig. 1 – METRIP modeling and analysis.**

first develop a high-level model of the system using the CIP_VAM language. Next, automatic model-to-model transformations generate a formal model for vulnerability evaluation and a decision model for solution optimization. The two models provide feedback to the high-level model for analysis and result allocation.

As an example, consider a high-level model that has no protection components. The initial transformations define possible allocations of protection components (e.g., cameras and sensors). The set of transformations is then used to evaluate the security provided by the protection components. The model can be refined to identify new concerns that may arise after the application of the transformations.

The following three steps are performed during the CIP_VAM language design phase to provide it with a clear and comprehensive structure:

- Identify the main levels into which the domain should be logically decomposed. The three levels are (i) infrastructure, which models the assets that must be protected; (ii) attack, which deals with the malevolent actions that can target the infrastructure; and (iii) protection, which focuses on devices and techniques that protect the assets.
- Characterize the entities and concepts existing in each level. This step is performed by identifying and analyzing relevant information sources (e.g., public databases, interviews with domain experts and the research literature).
- Identify the relationships between the domain levels and the consequent dependencies between the concepts and entities across the three levels.

When defining the associated UML profile, no recognized standard of work was identified specific to modeling critical infrastructure vulnerabilities and protection systems. Other UML profiles, such as UML-CI, define different aspects of infrastructure organization and behavior [1]. The CORAS language assists in modeling and analyzing the risk in changing systems in terms of their quality of service and fault tolerance characteristics [11]. UMLsec provides a method for expressing security information in system specifications [7]. MARTE [12] is a standard profile from the Object Management Group that customizes UML for modeling and analyzing non-functional properties of real-time and embedded systems. The dependability analysis and modeling (DAM) profile [2] is a specialization of MARTE that supports dependability analysis.

Since we chose to design and implement CIP_VAM using UML, we developed a profile based on the guidelines specified in [9,14]. First, we defined a conceptual model and language constructs for representing the essential concepts and relationships. The conceptual model includes constraints that specify how the language constructs can be used to build models. After the conceptual model was defined, the domain models were mapped to the vulnerability analysis modeling profile by identifying the UML notation for each domain concept.

## 4. Vulnerability analysis conceptual model

The CIP_VAM conceptual model is organized using three packages as shown in Fig. 2.

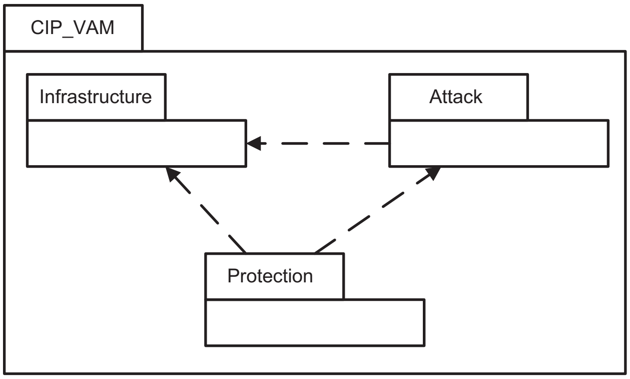


Fig. 2 – Package organization.

- The Infrastructure package provides a complete description of the physical system to be analyzed. In particular, it contains asset and environment related concepts.
- The Attack package describes the threats to the associated infrastructure.
- The Protection package provides a description of techniques and countermeasures that may be applied to defend assets.

Relationships between the packages arise from dependencies existing between the infrastructure, attack and protection levels. The target of an attack is always an asset, as indicated by the relationship existing from the Attack package to the Infrastructure package. The Protection package has a dependency relationship with both the Infrastructure and Attack packages because a protection component is applied to a specific infrastructure asset to combat one or more attacks.

### 4.1. *Infrastructure domain model*

Fig. 3 presents the infrastructure domain model. Four main concepts are identified: `Site`, `Interface`, `Object` and `Service`. The infrastructure consists of a set of sites and objects that may be located within the sites. A site may contain one or more sub-sites (e.g., building with multiple rooms) and is bounded by interfaces (e.g., windows, doors and gratings). An object may comprise sub-objects that provide or request a service, which in turn may be implemented using sub-services.

Note that `Asset` has a special role in the infrastructure model. It encompasses the concepts needed for vulnerability analysis and may be applied to sites, interfaces, objects and services. Notably, an asset is characterized by several attributes: economic value (value), likelihood that an attack is successful (vulnerability), probability of an attack (attackProb), quantitative estimate of the potential of an undesirable outcome (risk), and qualitative estimate of risk (riskLevel). Sites, interfaces, objects and services are all considered assets if they have an economic value. The object constraint language, which describes the rules that apply to the UML model, guarantees that if one instance of an asset exists, then

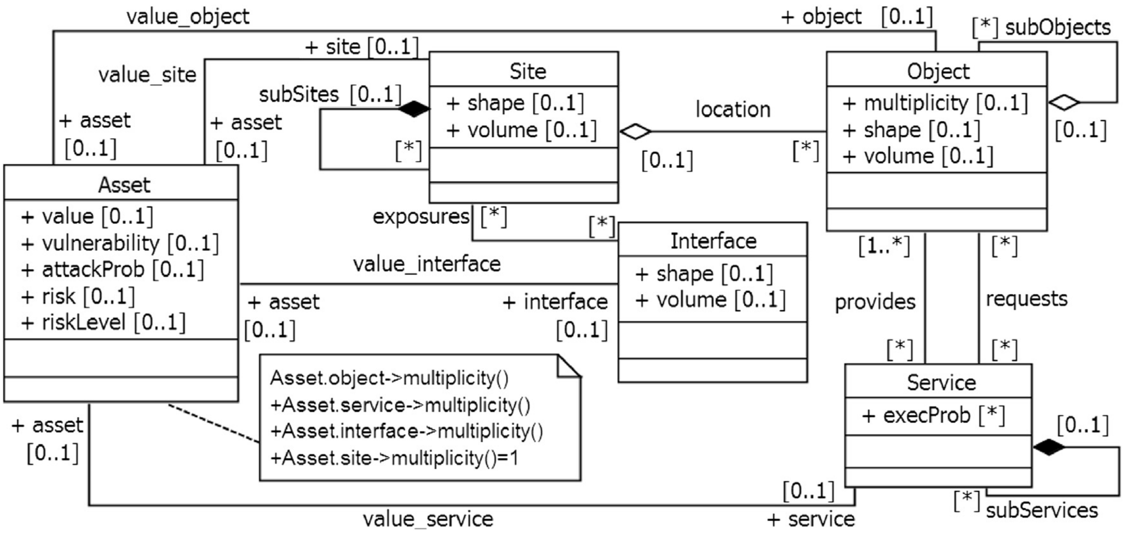


**Fig. 3 – Infrastructure domain model.**

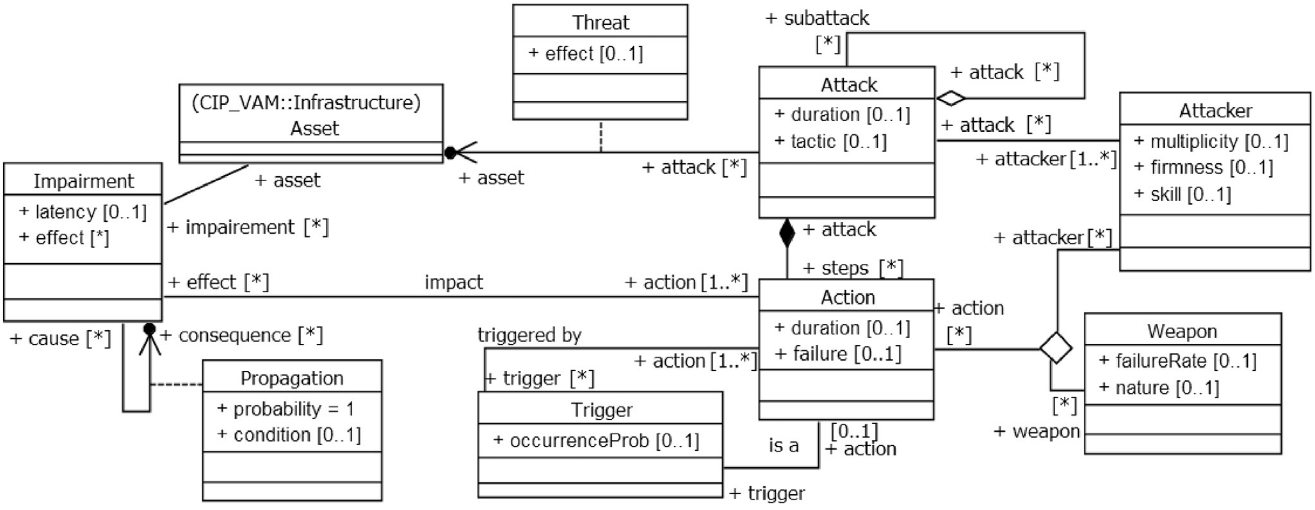


**Fig. 4 – Attack domain model.**

exactly one association is established with an instance of either site, interface, object or service.

## 4.2. *Attack domain model*

Fig. 4 presents the attack domain model. The main class of this package is `Attack`, which depends on `Asset` via the `Threat` association. `Attack` represents the sequence of actions that are required to exploit the system.

Both `Attack` and `Action` are characterized by a temporal duration (i.e., time elapsed between the start and end of the attack or action). Additionally, `Attack` is characterized by tactic (i.e., nature of the attack, such as kidnapping, armed attack or sabotage). The impact to the physical protection system is expressed by the `Impairment` class.

The `Propagation` class indicates how damage can propagate to other impaired components with a given probability and under specific conditions. `Attacker` corresponds to the malicious actors who perform the actions; it is characterized by the number of people performing an action (multiplicity), psychological predisposition to overcome deterrents and defense systems (firmness) and capability of carrying out an attack (skill). `Weapon` represents the tool utilized by `Attacker` to perform the action and is characterized by its failure probability (failureRate) and type (nature). A ternary association models the relationship between `Attacker`, `Action` and `Weapon`. `Trigger` is an event enabling action that is characterized by an occurrence probability (occurrenceProb).

## 4.3. *Protection domain model*

Fig. 5 presents the protection domain model. The main class of this package is the abstract class `Protection`. It is characterized by the probability of successfully defending an asset (succesProb). `InstallationPoint` specifies the position of the application (e.g., height of a closed circuit video camera on a wall). The abstract concept `Protection` is realized by two concrete classes that rely on human

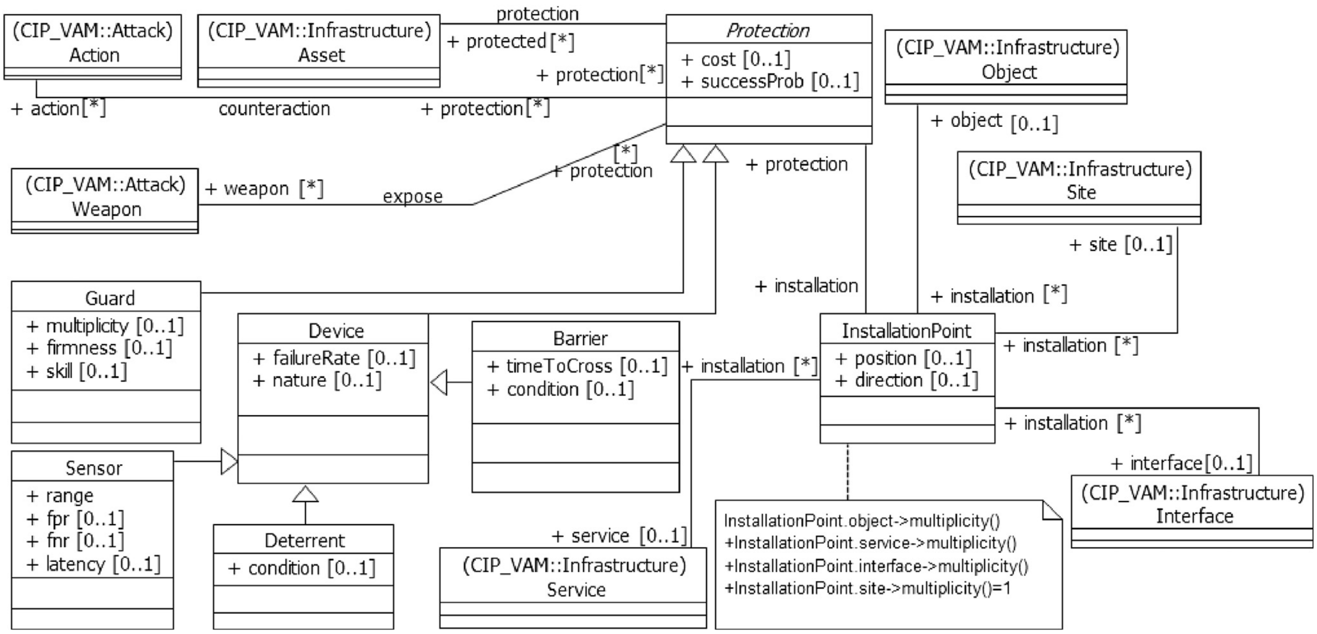


**Fig. 5 – Protection domain model.**

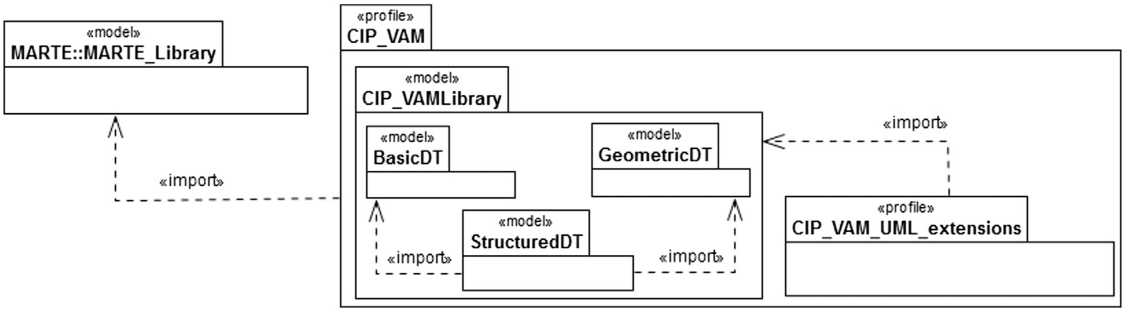


**Fig. 6 – CIP_VAM profile.**

protection and security devices: (i) `Guard` class, which indicates the presence of a security service that relies on human personnel (e.g., armed guards); and (ii) `Device` class, which is characterized by a failure probability (failureRate). `Device` is further specialized by `Barrier`, `Sensor` and `Deterrent`. `Barrier` is intended to lock an interface or increase the time and difficulty to overcome it (timeToCross). `Sensor` is characterized by the maximum area covered by the sensor (range) and by the false positive and false negative rates (fpr and fnr). `Deterrent` expresses the ability to demoralize an attacker from performing a malicious action and is characterized by the condition under which the deterrent has an effect (condition).

## 5. Vulnerability analysis modeling profile

This section describes the mapping of the conceptual model to a concrete profile. For each class, the attributes, associations and constraints are considered to create a suitable UML profile according to the guidelines provided in [9,14]. The resulting CIP_VAM profile provides the UML extensions that represent the vulnerability and protection aspects of the physical protection system.

The profile comprises the CIP_VAM Library and a set of UML extensions (Fig. 6). The library defines the set of basic and complex data types necessary to correctly define the tag values. The UML extension package contains the stereotypes, attributes and constraints that extend the UML meta-model.

### 5.1. *Vulnerability analysis library*

The CIP_VAM Library defines the basic, geometric and structured data types in three packages: BasicDT, GeometricDT and StructuredDT. Note that CIP_VAM imports the MARTE Library [12] in order to define these types. Specifically, CIP_VAM uses MARTE's primitive types and the ability to specify quantitative and qualitative non-functional properties using the Value Specification Language (VSL). In addition, CIP_VAM uses the Measurement Units package to specify

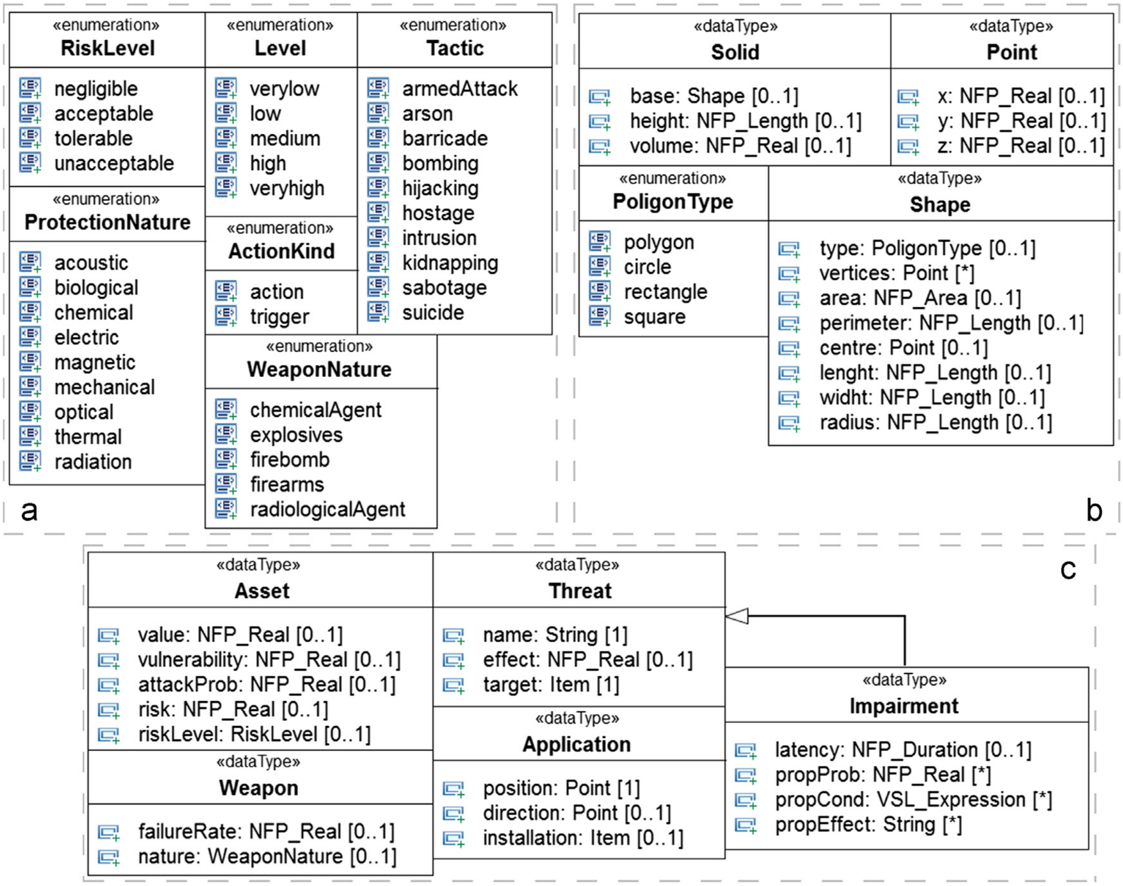


**Fig. 7 – CIP_VAM Library: (a) basic enumeration types; (b) geometric data types; (c) structured data types.**

quantities, measures, units, unit conversions and physical attributes (e.g., area, length, time and energy).

The CIP_VAM Library comprises three packages: (i) BasicDT containing simple enumerations (Fig. 7(a)); (ii) GeometricDT containing geometric data types for modeling physical structures and spaces (Fig. 7(b)); (iii) StructureDT containing complex data types created by aggregating BasicDT and GeometricDT (Fig. 7(c)).

### 5.2. *UML extensions*

The CIP_VAM UML extensions are organized according to the domain model structure. Fig. 8 provides an overview of the UML extensions. To transform a domain model to a UML profile, the domain model concepts are classified according to the taxonomy expressed in [9]: abstract, firm, uncertain or parametric classes. Abstract classes refer to accessory concepts, firm classes are used as language constructs, uncertain classes contain indeterminate concepts, and parametric classes categorize concepts that can change depending on the sub-problem domain. Firm classes are directly mapped to stereotypes while *ad hoc* solutions are adopted for the other classes. After the classification, the UML meta-classes are extended to associate each stereotype with an actual class by considering similar UML profiles. As an example, `Object` and `Service` extend the UML classifier because it is reasonable for an object to be represented by a UML component or a class; in addition, service extends the UML use case.

The UML extension supports the optimization of model attributes by grouping common features. An example is the `Item` stereotype of the Infrastructure package that was introduced to prevent replication of the asset tag. Additionally, optimization patterns [9] are used to translate attributes to data types using an *is-a* relationship. An example pattern in the Attack package translates the `Action` and `Trigger` domain classes into the `Action` stereotype and tag type, which can be set to action or trigger. Note that some relationships included in the domain model are not translated to profile elements because we rely on the existing UML notation when appropriate (as prescribed in [14]). An example is the subService association that is implemented using the ≪ include ≫ relationship between use cases and the owner attribute of classifier.

## 6. Using the vulnerability profile

This section applies the CIP_VAM profile to develop and analyze the vulnerability and protection models of a notional railway station. The physical layout of the station, potential attack scenarios and protection systems are modeled to provide a high-level representation for the subsequent analysis. For reasons of brevity, an excerpt of the model is presented.

Fig. 9 shows the layout of the railway station. The station has three access points from the street, one main entrance

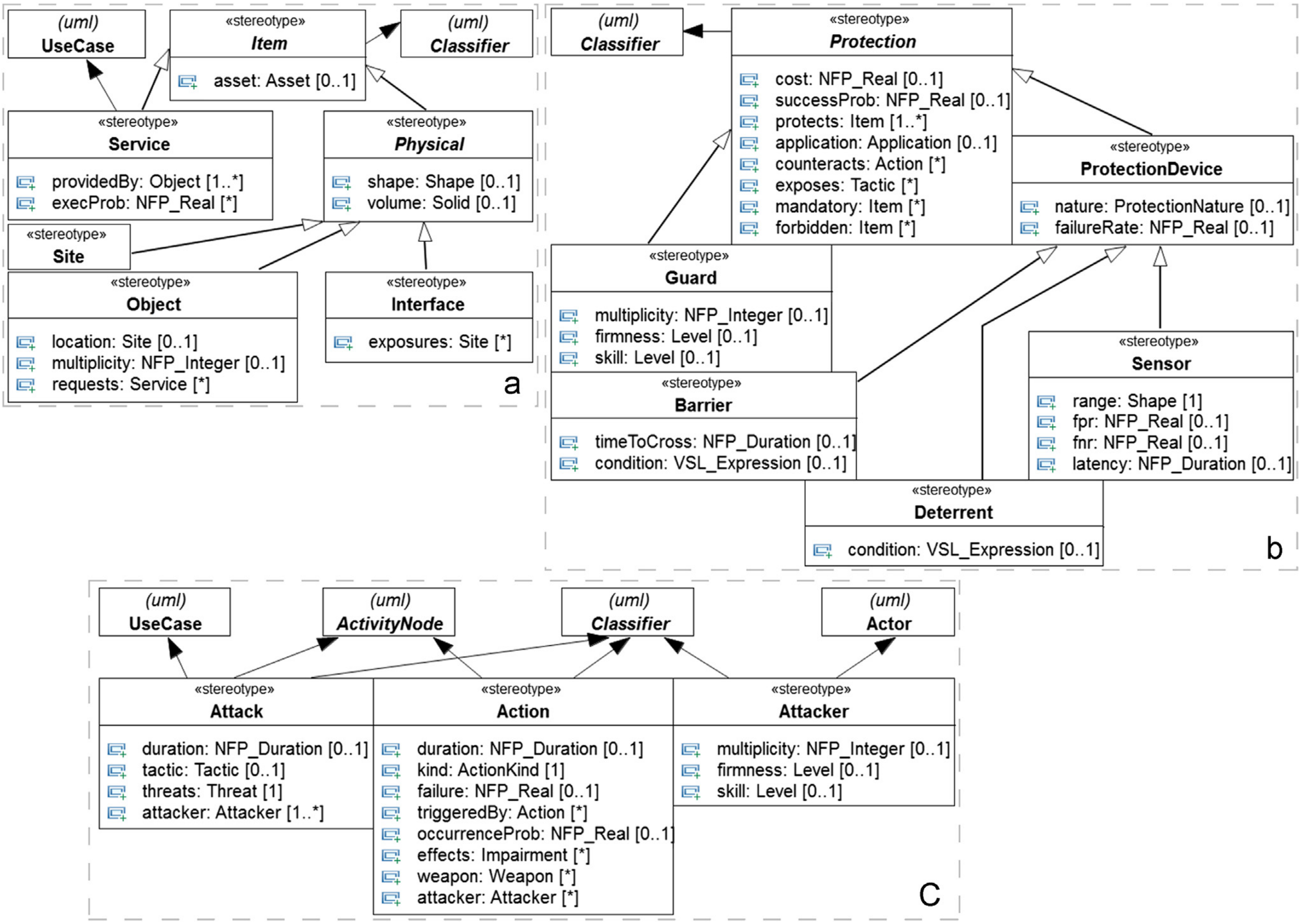


**Fig. 8 – Overview of UML extensions: (a) infrastructure package; (b) attack package; (c) protection package.**

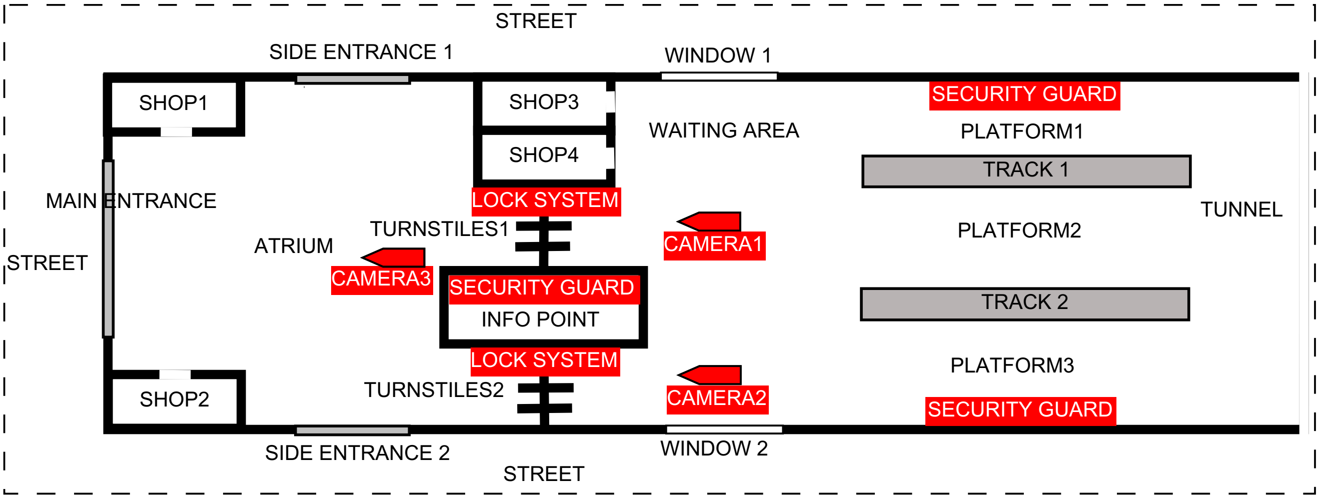


**Fig. 9 – Railway station layout.**

and two side entrances. The station is internally divided into two functional areas, an atrium on the left that is freely accessible and a waiting area on the right that can be accessed only by passing through turnstiles. An information point that hosts security personnel is located near the two lines of turnstiles. Four shops are located in the station, two in the atrium and two in the waiting area. Two sets of tracks are available for passenger trains, three platforms for boarding and disembarking, and a tunnel through which trains must pass when entering and departing the station. The protection systems include three video cameras that monitor the main entrance and turnstiles, lock systems for the

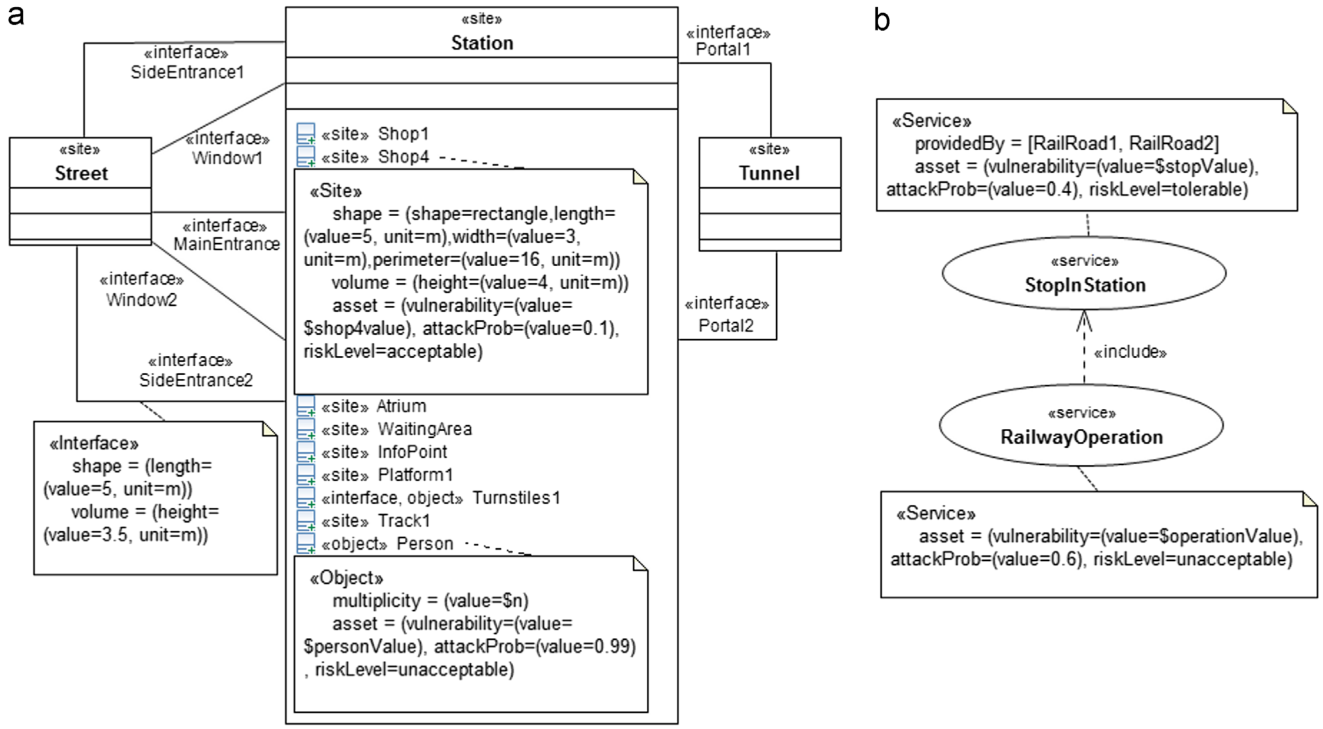

a
«site»
Station
«interface»
SideEntrance1
«site»
Street
«interface»
Window1
«interface»
MainEntrance
«interface»
Window2
«interface»
SideEntrance2
«Interface»
shape = (length=
(value=5, unit=m))
volume = (height=
(value=3.5, unit=m))
«site» Shop1
«site» Shop4
«Site»
shape = (shape=rectangle,length=
(value=5, unit=m),width=(value=3,
unit=m),perimeter=(value=16, unit=m))
volume = (height=(value=4, unit=m))
asset = (vulnerability=(value=
$shop4value), attackProb=(value=0.1),
riskLevel=acceptable)
«site» Atrium
«site» WaitingArea
«site» InfoPoint
«site» Platform1
«interface, object» Turnstiles1
«site» Track1
«object» Person
«Object»
multiplicity = (value=$n)
asset = (vulnerability=(value=
$personValue), attackProb=(value=0.99)
, riskLevel=unacceptable)
«interface»
Portal1
«site»
Tunnel
«interface»
Portal2
b
«Service»
providedBy = [RailRoad1, RailRoad2]
asset = (vulnerability=(value=$stopValue),
attackProb=(value=0.4), riskLevel=tolerable)
«service»
StopInStation
«include»
«service»
RailwayOperation
«Service»
asset = (vulnerability=(value=$operationValue),
attackProb=(value=0.6), riskLevel=unacceptable)

**Fig. 10 – UML diagrams of the station infrastructure: (a) structural view; (b) service provided.**

turnstiles that are controlled from the information point, and four security guards who patrol the platforms, waiting area and access points to the waiting area.

### 6.1. *Infrastructure domain model*

Fig. 10(a) presents the UML model of the station using the Infrastructure package constructs. Site contains several subsites (e.g., shops, atrium and waiting area) with associations represented by the interfaces of the station to the street and the tunnel (e.g., three entrances, two windows and two portals). Passengers inside the station are represented by the object construct through a member person with a multiplicity tag value of $n. The tag values pertaining to Shop4 show that CIP_VAM can represent geometrical features such as shape to express the rectangular shape of the shop (i.e., length of 5 m and width of 3 m).

The tag asset identifies the assets that are to be protected. In this use case, they are Shop4 and Person. The probability of an attack on the shop is 0.1, the economic value of the shop is $shop4value and the risk level is acceptable. However, the probability of an attack against a person is 0.99, the value of each person is $personValue and the risk level is unacceptable.

Fig. 10(b) shows the StopInStation service provided by the objects RailRoad1 and RailRoad2. The relationship is expressed by the tag providedBy for the StopInStation use case. Note that the service is included in RailwayOperation because it requires the correct execution of StopInStation for proper railway functionality. Both services can result in an economic loss due to potential unexpected stops, as expressed by $stopValue and $operationValue.

### 6.2. *Attack domain model*

Consider the attack scenario shown in Fig. 11(a). In this scenario, a terrorist intends to detonate a bomb near the turnstiles inside Shop4 or on Platform3.

Two use cases stereotyped as ≪attack≫ are created. Both instances include the *intrusion_in_waiting_area* attack. The tag values associated with the terrorist actor express the fact that the attacks can be conducted by just one terrorist with medium skill and medium firmness. The tag value for tactic is set to intrusion for the use case representing the intrusion into the waiting area. The threat value for the *place_a_bomb_in_shop4* use case models the destruction chain of a successful attack execution – the explosion inside the station propagates to all the members included in the station. As such, the result is the destruction of Shop4 with an economic loss of $shop4value, the loss of lives with an economic loss of $personValue (multiplied by $n), and the destruction of the two railroads with the loss of service StopInStation, and the resulting loss of RailwayOperation.

The *intrusion_in_waiting_area* attack scenario is expressed using the activity diagram in Fig. 11(b). The action sequence models the possibility of a terrorist reaching the waiting area by crossing one of the two entrances, or by forcing a window open after 10 p.m. when the station is closed. This latter condition is expressed by the triggeredBy tag value with the impact of the action modeled through the effects tag value.

### 6.3. *Protection domain model*

Fig. 12 shows the UML model of the protection systems in the railway station. In particular, the model highlights how the

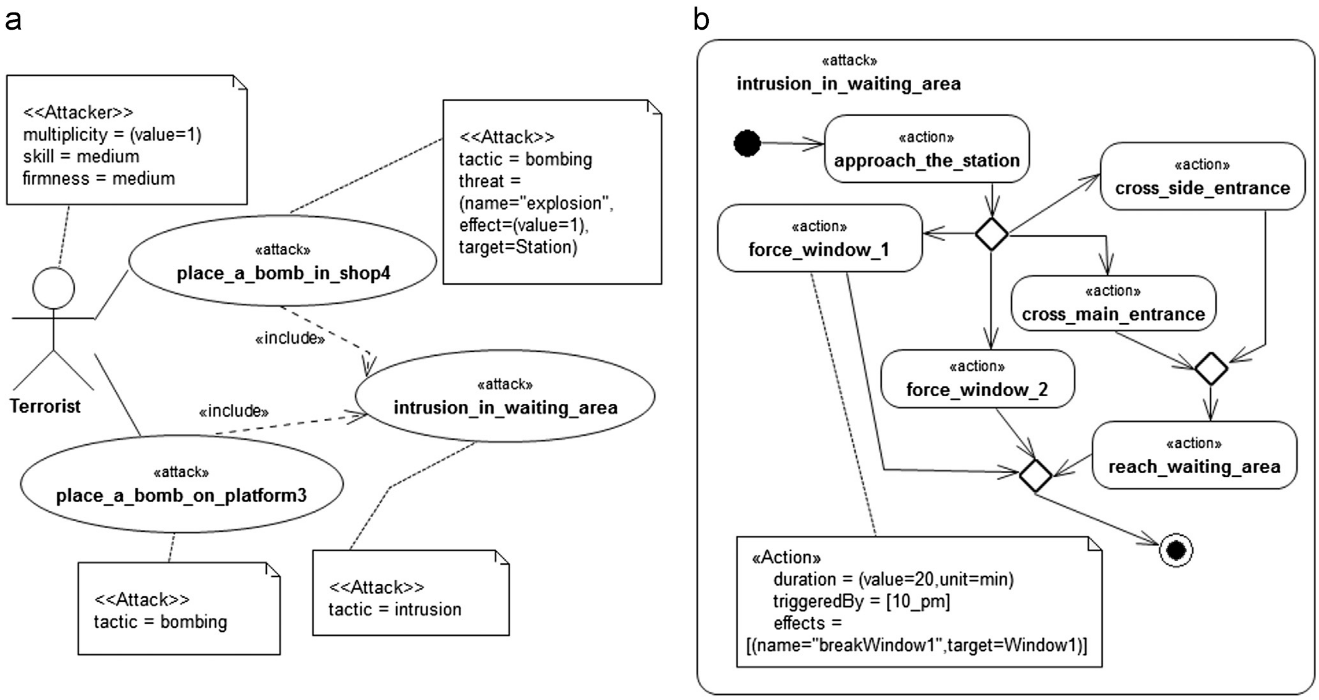


**Fig. 11 – Station attack scenario: (a) attack use case diagram; (b) intrusion activity diagram.**

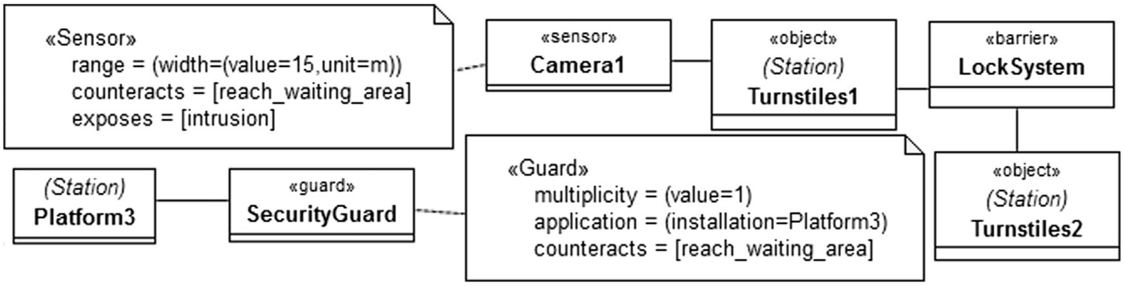


**Fig. 12 – UML view of station protection systems.**

tag values can express the technical specifications of the protection systems. The counteracts tag value lists the actions against which the protection systems provide effective safeguards. For example, the action *reach_waiting_area* may be detected by the camera or prevented by the lock system.

### 6.4. *Formal model generation*

The scenario helps clarify how a model-to-model transformation is used to translate a UML model comprising infrastructure, attack and protection sub-models. A Bayesian network is derived from the UML model in order to perform vulnerability analysis. We use a simple example to demonstrate how the Bayesian network transformation can be used to evaluate the vulnerability of the railway station.

Fig. 13 shows a portion of the Bayesian network derived from the railway station UML model. The three dashed boxes contain the aspects of the Bayesian network obtained from the infrastructure, attack and protection UML sub-models. Starting with the activity diagram, a Bayesian network node is generated for each activity. When two activities are adjacent, a dependency (arc) exists between the nodes (e.g., *approach_the_station* and *cross_side_entrance*). The Infrastructure package is used to reason about the locations of attack events and protection systems. For example, the node *attack_going_on_1* models the continuity of the attack when the *cross_side_entrance* branch of the activity diagram is considered. Note that the *attack_going_on_1* depends on two nodes that represent an attack inside the area (*cross_side_entrance*) and the presence of a guard (*guard*).

The protection sub-model can be used to evaluate the effect of the reliability and trustworthiness of a sensor for detecting an attack event. An additional Bayesian network node is generated for each protection device. For example, the node *cam_1* has the states ok (camera is working properly) and not ok (camera is not working properly). Table 1 shows the conditional probability table for the *attack_going_on_5* node with respect to the states of the *force_window_1* and *cam_1* nodes.

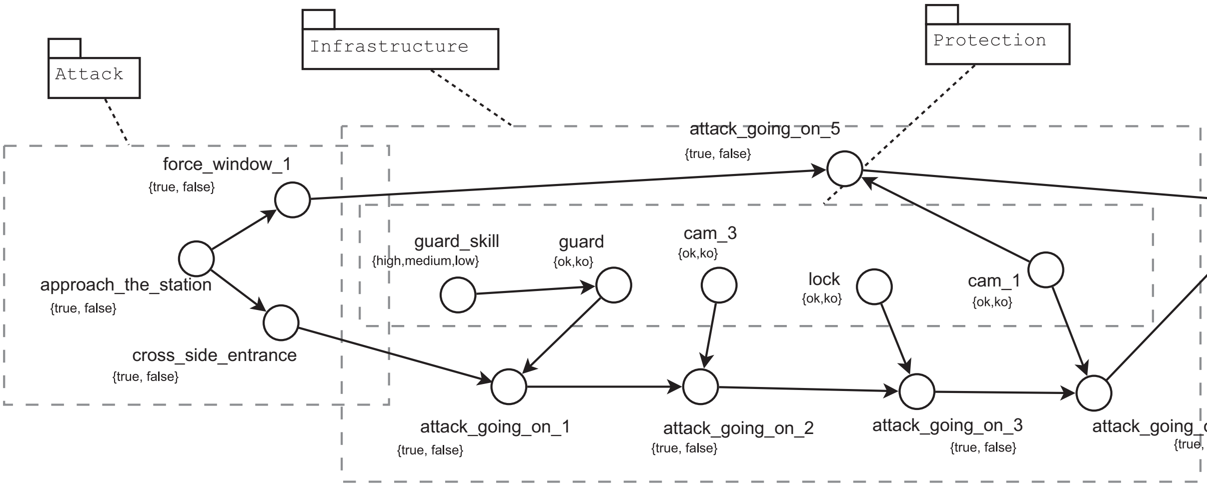


**Fig. 13 – Portion of the Bayesian network generated from the case study.**

**Table 1 – Conditional probability table for *attack_going_on_5*.**

| *force_window_1* | *cam_1* | Output=true | Output=false |
|---|---|---|---|
| True | Ok | 1 − fnr | fnr |
| True | Not ok | 0 | 1 |
| False | Ok | fpr | 1 − fpr |
| False | Not ok | 0 | 1 |

**Table 2 – Sensitivity analysis.**

| Scenario | Description | Vulnerability |
|---|---|---|
| Nominal | fnr of *cam_1* is 0.8 | 0.147 |
| Degraded | *cam_1* is absent | 0.702 |
| Improved | fnr of *cam_1* is 0.95 | 0.041 |

For demonstration purposes, the model is populated with input data values taken from the CIP_VAM model. The fnr and fpr values shown in the table are derived from the *≪sensor≫ camera_1* tag values. Other data values used in the analysis are the camera *cam_1* fnr and reliability values of 0.8 and 0.99, respectively.

The vulnerability is evaluated by computing the probability distribution of the *intrusion* node given that the state of *approach_the_station* is true. A simple sensitivity analysis of the variations in the physical protection system is conducted in order to demonstrate the utility of the CIP_VAM modeling approach. Table 2 summarizes the results of the analysis for three cases. The first case corresponds to a nominal scenario with the physical protection system configured according to the scenario described above. The second case corresponds to a modification of the physical protection system where *camera_1* is removed from the system. The third case uses a more trustworthy camera with a lower fnr. As expected, the results confirm that the presence of the trustworthy camera enhances the protection of the system. Although this example is simple, it demonstrates the utility of the model, especially for modeling and analyzing complex physical protection schemes.

## 7. Conclusions

The formal method for evaluating vulnerabilities existing in critical infrastructure assets engages the model-driven paradigm. Important characteristics of the infrastructure, attacks and protection systems are expressed using the CIP_VAM domain-specific modeling language. The model-driven methodology provides analysts and evaluators with powerful modeling constructs associated with field applications. Additionally, formal models and decision-support models may be derived through transformations of CIP_VAM artifacts.

The CIP_VAM language is designed using the UML profiling technique. The case study focusing on protection systems for a railway station demonstrates the effectiveness of the modeling approach and the utility of the CIP_VAM profile. Also, the example model-to-model transformation from a high-level UML model to a Bayesian network clarifies the approach for evaluating the vulnerabilities of critical infrastructure assets.

Our future research will investigate the completeness of the CIP_VAM language, with a focus on the railway domain. Model-to-model transformations will be developed to automatically convert decision-making results to high-level models. Additionally, the CIP_VAM profiling approach will be augmented using a dynamical method to enhance vulnerability analyses. Finally, the CIP_VAM framework will be used to analyze how various factors could influence the attractiveness, vulnerability and fragility of railway assets to targeted attacks.

## Acknowledgement

This research was partially supported by the Prevention, Preparedness and Consequence Management of Terrorism and Other Security-Related Risks Program of the Directorate-General Home Affairs, European Commission under Grant HOME/2010/CIPS/AG/035 METRIP – Methodological Tools for Railway Infrastructure Protection.

## REFERENCES

[1] E. Bagheri, A. Ghorbani, UML-CI: a reference model for profiling critical infrastructure systems, *Inf. Syst. Front.* 12 (2) (2010) 115–139.

[2] S. Bernardi, J. Merseguer, D. Petriu, A dependability profile within MARTE, *Software Syst. Model.* 10 (3) (2011) 313–336.

[3] K. Czarnecki, S. Helsen, Feature-based survey of model transformation approaches, *IBM Syst. J.* 45 (3) (2006) 621–645.

[4] F. De Cillis, M. De Maggio, C. Pragliola, R. Setola, Analysis of criminal and terrorist related episodes in railway infrastructure scenarios, *J. Homeland Secur. Emergency Manage.* 10 (2) (2013) 1–30.

[5] M. Garcia, *Design and Evaluation of Physical Protection Systems*, Butterworth-Heinemann, Burlington, Massachusetts, 2008.

[6] B. Jenkins, B. Butterworth, Explosives and Incendiaries Used in Terrorist Attacks on Public Surface Transportation: A Preliminary Empirical Analysis, MTI Report WP 09-02, Mineta Transportation Institute, San Jose, California, 2010.

[7] J. Jurjens, *Secure Systems Development with UML*, Springer-Verlag, Berlin Heidelberg, Germany, 2005.

[8] I. Kotenko, M. Stepashkin, Attack graph based evaluation of network security, in: *Proceedings of the Tenth IFIP TC-6/TC-11 International Conference on Communications and Multimedia Security*, 2006, pp. 216–227.

[9] F. Lagarde, E. Espinoza, F. Terrier, S. Gerard, Improving UML profile design practices by leveraging conceptual domain models, in: *Proceedings of the Twenty-Second IEEE/ACM International Conference on Automated Software Engineering*, 2007, pp. 445–448.

[10] E. LeMay, M. Ford, K. Keefe, W. Sanders, C. Muehrcke, Model-based security metrics using adversary view security evaluation, in: *Proceedings of the Eighth International Conference on Quantitative Evaluation of Systems*, 2011, pp. 191–200.

[11] M. Lund, B. Solhaug, K. Stolen, Risk analysis of changing and evolving systems using CORAS, in: A. Aldini, R. Gorrieri (Eds.), *Foundations of Security Analysis and Design VI*, Springer-Verlag, Berlin Heidelberg, 2011, pp. 231–274.

[12] Object Management Group, The UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems, MARTE Version 1.1, Needham, Massachusetts, 2011, URL: ⟨http://www.omgmarte.org⟩.

[13] Object Management Group, Unified Modeling Language: Infrastructure and Superstructure, V2.4.1, Needham, Massachusetts, 2011, URL: ⟨http://www.omg.org/spec/UML/2.4.1⟩.

[14] B. Selic, A systematic approach to domain-specific language design using UML, in: *Proceedings of the Tenth IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing*, 2007, pp. 2–9.

[15] The METRIP Project, Methodological Tools for Railway Infrastructure Protection, University Campus Bio-Medico of Rome, Rome, Italy, URL: ⟨metrip.unicampus.it⟩.