Preliminary Risk and Mitigation Assessment in Cyber-Physical Systems

András Földvári*, Francesco Brancati**, András Pataricza*

*Department of Measurement and Information Systems
*Budapest University of Technology and Economics
**ResilTech S.r.l.
{foldvari,pataric}@mit.bme.hu, francesco.brancati@resiltech.com

***Abstract*—Malicious attacks endanger cyber-physical systems to a drastically increasing extent. Successful attacks intruding on the physical part of the system can cause severe or even catastrophic losses. The paper presents a model-based system engineering (MBSE) solution for the assessment and mitigation strategy design tool tailored to the peculiarities of SMEs with limited human and financial resources. The proper quality of the security assessment is assured by using embedded formal methods.**

***Index Terms*—risk assessment, cyber-physical systems, qualitative reasoning**

I. Introduction

IT/OT systems are specific cyber-physical systems (CPS) or systems-of-systems (CPSoS) in which *information technology* (IT) controls the *operation technology* (OT) of a manufacturing or similar physical production process. The exposure of the economy and society to malicious attacks is increasing, as indicated by the immense damage caused. Increasing cyber-security's effectiveness has become a top priority in the US [1] and the EU [2], and actions are being taken to create a qualitatively new culture.

IT/OT systems pose specific professional challenges since intrusions through the IT can jeopardize the functionality and safety of the physical OT. Thus, attack-induced errors in the OT can also amplify the risk to catastrophic levels through the security-dependability interdependence.

Significant efforts are being made everywhere to enhance the security of large IT/OT production processes. This is a high priority for operating large critical infrastructures for which the necessary expertise and resources are available.

The issue is more contradictory for small and medium-sized enterprises (SMEs) in manufacturing and related non-IT services. IT is frequently considered "only" as much a critical resource as energy supply, and IT management is severely limited regarding awareness, staffing, and other resources. Therefore, many of these SMEs can only achieve the much-needed security consolidation in a gradual and resource-constrained way.

This work was funded by the ADVANCE (EU-RISE 823788), PrOTectME (EIT Digital 21293), and the DigitalTech EDIH (101083965) 2021-1.2.1-EIT-KIC-2021-00006 projects.

Our objective was an easy-to-use IT vulnerability to OT dependability impact analysis method and supporting tool for the SME sector complying with its specific constraints. A *lightweight MBSE framework* (Fig. 1) with *embedded formal methods* is the primary support in *risk assessment* to *mitigation* for analysts of moderate IT security expertise.

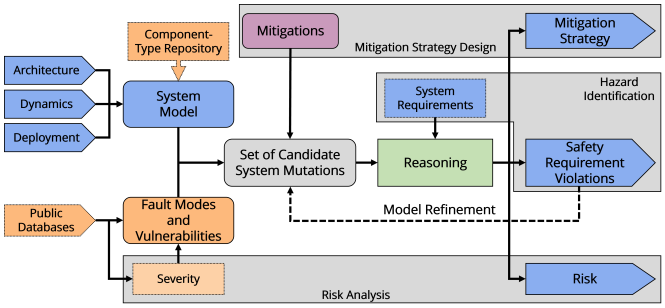

Fig. 1. Experimental Framework

1) **System Model:** The *system model* results from merging the different aspect models (like *architecture*, *dynamics*, and *deployment*) of the complete IT/OT system into a single model sharing a uniform mathematical paradigm. *Component-type libraries* support reusing already existing sub-models.
2) **Set of Candidate System Mutations:** Injecting validated information on the component security faults and the local impacts on attacks (vulnerabilities, error propagation) from validated public collections extends the system model with a set of candidate mutations to be evaluated. The subsets of the mutations define attack scenarios, including all the relevant faults and vulnerabilities.
3) **Reasoning:** Merging the candidate set of system mutations with the *system requirements* into a uniform mathematical model forms the basis of the evaluation.
4) **Hazard Identification:** All the *candidate attack scenarios* over the joint model undergo exhaustive analysis by automated formal methods generating a list of attacks that violate system safety requirements.
5) **Model Refinement:** The models in the first round of analysis are kept simple by a high level of qualitative abstraction [3]. The shortlist of potentially successful attacks

may contain spurious solutions due to over-abstraction (but the method guarantees that no actual hazardous attack is overlooked). This way, a successive iteration after CEGAR-styled (Counter-Example Guided Abstraction Refinement) model refinement and re-analysis or expert review is needed to eliminate false solutions.

6) **Quantitative Risk Analysis:** Enriching the model and the components describing attack impacts and cost facilitate a rough-granular risk analysis.
7) **Mitigation Strategy:** Finally, enriching the model with mitigation solution and their costs, the mathematical optimization-based cost-benefit analysis creates a trade-off between investments and risk reduction.

As a summary, this paper presents an *early risk assessment method* based on an environment built around an embedded algorithm for *error propagation analysis* (EPA), assessing the system-level impacts of local attacks (and unintentional faults) tailored to the peculiarities of SMEs.

II. Solution Strategy

The cybersecurity IT/OT impact analysis tool for SMEs uses the same EPA core [4] as for big enterprises. Adapting to the peculiarities of SMEs requires several algorithmic extensions to reduce the expertise threshold.

A. Specific Requirements for SMEs

In particular, the *simplicity*, *interpretability* of each step, and the *explainability* of the results, assuming IT system managers of *average skills*, were of priority concern.

Moreover, the tool provides active support for the phases of the MBSE process in phases, where in-the-large analysis typically relies on security experts.

- One such phase is modeling and parametrization, where sensitivity analysis-styled support highlights the critical decisions from the point of view of the overall result of the impact analysis to reduce the impacts of human errors.
- The other phase is the elimination of spurious solutions. Several refinement options are offered as part of the CEGAR-styled model refinement and re-analysis, thus substituting complex decisions typically made by security experts with easier-to-make simpler ones.
- Optimizing the mitigation measures under cost constraints facilitates a multi-phase security consolidation by SMEs.

B. Modeling Approach

One of the significant challenges in modeling a CPS is the fundamental difference between the *signal flow* in the IT and the *flow of quantities* in physical parts. [5] *Signal (i.e., data) flows* are directional between predefined outputs and inputs of the IT components. Data sequences typically characterize their temporal behavior. Interconnected physical components share *quantities underlying some undirected conservation law(s)* (modeled as in-out variables). Continuous-time waveforms or -after sampling- time series describe their temporal behavior.

A way of faithful modeling of CPSs is hybrid modeling. However, such models are overly detailed for impact analysis (and very demanding for skills). However, the fundament of *qualitative physics* can substantiate sufficiently faithful models for their analysis. Qualitative modeling *-partitions continuous domains into different clusters of identical or similar behavior* along landmarks and represents them by a *discrete model at the granularity level of clusters*. For example, the essence of the phenomena in an IT system caused by varying workloads can be described starting from categorical ordered variables (`low, medium, high, overloaded`).

This way, qualitative abstraction and modeling provide a lingua franca for describing the IT and OT parts. Reusing the paradigm of *qualitative modeling* [3], [6] facilitates using (hidden formal) *reasoning* (in summary, QR) forms the core for our system, requirement, attack, and fault modeling and analysis.

The advantage of this approach is that it is close to natural human thinking, and the interpretation of the solutions is straightforward. As an abstract description and analysis paradigm, QR deliberates the user on going into superfluously over-detailed models and their interpretation. (For instance, a doctor evaluates a lab funding, like "The ESR in the blood test is overly `high`, *and the patient has a* `very high` *temperature, which indicates a* `very heavy` *infection.*")

On the other hand, QR extended with temporal logic has powerful capabilities, like handling uncertain information, estimation of the solution space, or -with some limitations- estimation of optimal solutions.

QR involves building a qualitative model of the system and its behavior. QR fits seamlessly for EPA because it allows the identification of the most critical fault scenarios and their potential impact on the system. By analyzing the relationships between the components of the system and their interactions, qualitative reasoning can provide the potential effects of faults in the system. This can be useful in identifying which aspects of the system are most vulnerable to failure and developing strategies to mitigate such risks.

C. Experimental Framework

The experimental framework is designed to support preliminary risk assessment using logic reasoning as its background. It supports the exhaustive qualitative risk assessment using a hidden formal method.

For the high-level engineering modeling, we used the TOGAF Archimate [7] modeling language, which provides a common language and toolkit between the analyst and the engineers and lightweight modeling of IT/OT systems. This solution allows engineers to describe at a general level the components of the system, their types, and the relations between them. Furthermore, aspects related to security can be assigned to the model [8]. This system validation model can then be used as input to the logic reasoner engine that performs the EPA and hazard identification.

Answer Set Programming [9] (ASP) is the basis of the logic reasoning component of the framework. ASP is a declarative

programming approach to model and solves search and optimization problems. It combines an expressive representation language, a model-based problem specification methodology, and efficient solving tools.

We extended our previously developed qualitative EPA library [4] to support hierarchical risk assessment and qualitative risk evaluation. In ASP, the model elements (components) and the relationships can be described. The model can also be extended with dependability and security aspects assigned to the components.

The solution builds on Telingo [10], which is an extension of ASP that allows the definition of temporal logic programs handling linear temporal logic (LTL) formulas capturing the dynamic behavior of the qualitative model. It extends ASP with model-checking capabilities by adding constraints as formal LTL expressions for the formal description and validation of requirements.

The result of the qualitative error propagation analysis in ASP is a vector that describes the violated safety constraints and gives the components' error propagation path and active fault modes. This can be extended with cost metrics. For example, when searching for the most critical consequence, the severity of the faults can be set as cost metrics.

III. Related work

Engineering of *-critical applications developed various methods for assessing fault impacts. However, most approaches for "in-the-large EPA" assume the specific skills of the domain experts, which are hardly available at our target enterprises performing "in-the-small" analysis.

This section summarises some prominent EPA approaches used in the industry, particularly emphasizing approaches to link security and dependability, and discusses the differences between qualitative EPA and these solutions.

A. Error Propagation Analysis

Several classic solutions are available for EPA, such as Fault Tree Analysis (FTA), Markov chains, and Petri Nets, which consider different aspects of system behavior and come with unique advantages and limitations.

Fault Tree Analysis (FTA) is a top-down method for error propagation analysis that helps to identify critical points in a system and evaluate the risk associated with events. However, FTA does not examine components' behavior and interactions, and the results may be incomplete and inaccurate. Qualitative error propagation analysis can be incorporated into the FTA process to achieve a more comprehensive and accurate result. Markov chains and Petri nets are other approaches for EPA but require specific expert knowledge. However, this expertise is not always available in all SMEs. In contrast, qualitative EPA provides a general high-level approach that is easy to interpret from a decision-maker or engineering perspective.

B. Security Risk Evaluation

Several open databases are available for different focus and abstraction levels for security risk evaluation. For instance, the Common Vulnerabilities and Exposures (CVE) [11] database maintains a registry of cyber security vulnerabilities that are publicly known. The vulnerabilities in CVE are measured by the Common Vulnerability Scoring System (CVSS) [12] that denotes its severity via a calculated score. The Common Weakness Enumeration (CWE) [13] lists various software and hardware weaknesses, and the Common Attack Pattern Enumeration and Classification (CAPEC) [14] database presents a comprehensive glossary of known attack patterns utilized by adversaries to exploit identified weaknesses.

In [15], the authors proposed a domain-specific language to model the system and integrated the MITRE ATT&CK Matrix [16], [17] as an external library to model the attacks in the system. The resulting threat model can be used to generate attack graphs and use these graphs to simulate specific attack scenarios. The model also includes the mitigation from the attack matrix, allowing the simulation to examine the cases that can occur with the activation or deactivation of specific mitigation mechanisms.

In [18] an IoV security-specific ontology was built. They also combined it with a multi-level architecture model. So here, the attack can be traced and examined through multi-layers of attack paths. Based on this, ontology provides a library of components at different levels. The vulnerabilities and attacks can be assigned to these components. This paper generated an attack graph from the ontology, and the rule-based inference was performed.

These solutions can be used to evaluate how an attacker exploits vulnerabilities to reach his final target in the topological model of the system. However, these solutions do not include aspects related to dependability, i.e., it is not possible to investigate in detail how a fault caused by an exploited vulnerability violates requirements and how it propagates to the system level and degrades its service. Furthermore, these models do not include aspects necessary for quantitative analysis of faults, e.g., fault severity.

C. Dependability and Security

Security and dependability are separate concepts, although they overlap in attributes [19]. Security integrates the attributes of confidentiality, integrity, and availability.

There are many solutions for safety and security evaluation. In [20] the authors collected several approaches that combine safety and security for industrial control systems. These include generic methods, which usually develop a process for unification [21] or integration [22], [23].

Besides the generic methods, many solutions are extended versions of classical dependability evaluation methods, including extensions of FTA [21], [24], Petri nets, and Markov chain-based solutions. Additionally, there are security extensions for UML and SysML, commonly referred to as UMLsec [25] and SysMLsec [26], which are used in classical model-based system design.

IV. Risk Assessment

The pathology of cyber-attacks is similar to the one used on faults in dependability analysis. The main difference is that

faults are described by random probability in dependability, while in security, vulnerabilities are exploited by an attacker. An attacker's ability to exploit a vulnerability depends on factors such as their attack profile, skill, and motivation, and the system's ability to mitigate exploitation.

Another difference is that while in dependability, we usually talk about single faults, in security, most attacks are based on exploiting a combination of vulnerabilities. This way, the attacker activates many simultaneous or sequential faults in the system.

The main goal of a preliminary risk assessment is to provide the analyst with a broad understanding of the potential impact of the exploitation of vulnerabilities and faults that, if left unaddressed, could violate the system requirements. This information can be used to prioritize the faults and vulnerabilities based on their severity and potential impact.

For instance, if a vulnerability is discovered in a service accessible from a public network and exploiting it could have critical consequences, it would be given a higher priority for fixing compared to a less severe fault that exists in a closed internal network. By prioritizing the correction of faults based on their potential impact, limited resources and time can be allocated more efficiently to ensure the safety and reliability of the system.

The risk assessment comprises four aspects: 1. Scenario identification; 2. Quantization of risk; 3. Mitigation; 4. Cost-benefit estimation and optimization

A. Scenario Identification

The first step in the dependability evaluation process is to identify potential scenarios that could lead to a violation of the system requirements. These scenarios are typically derived from the possible fault modes of the system components. Using error propagation analysis, scenarios are constructed by combining fault modes and assessing whether the resulting combination could lead to a violation of system requirements.

For dependability evaluation, the following aspects should be considered:

1) Identify the system topology: Identify the logical or physical components that are the focus of the analysis and understand how they are interconnected.
2) Identify fault modes of components: Identify the different ways in which components can fail. This information is used to construct scenarios of potential error propagation.
3) Identify potential failures: Identify the potential failures of the system that could result in a violation of system requirements. Failures guide the analysis and prioritize corrective measures based on the severity and potential impact.

From the cybersecurity point of view, the goal of this step is to understand the cyber risk landscape from the perspective of potential attacks.

The following aspects define the cyber security logic attack scenario space:

1) Asset definition: Which company assets being analyzed could potentially be targeted?
2) Method identification: What possible methods could be used to attack these assets?
3) Threat actor identification: Who are the potential candidates that could carry out these attacks?
4) Loss event definition: What potential losses could occur as a result of these attacks?

The outcome of the step is the so-called "scenario space" that contains all potential scenarios that can lead to failures/losses. To facilitate this step, publicly available databases (CAPEC, CWE, CVE) can be utilized. MITRE ATT&CK (ICS) matrices were also used to assess what techniques and tactics are potentially exploitable in the system and create a risk.

B. Quantization of Risk

Qualitative risk assessment differs from quantitative risk assessment because it doesn't require analysts and decision-makers to compute precise numerical values. Instead, it provides a comprehensive overview of risks through descriptive means.

The IEC 61508 standard [27] provides a framework for the qualitative analysis of hazards. It is based on six categories of the likelihood of occurrence and 4 of consequence that are combined in a risk class matrix. The categories in the matrix indicate how a given risk can be classified.

Similar to safety analysis, cybersecurity also has qualitative approaches. The qualitative risk analysis in security follows the Open FAIR Risk Analysis (O-RA) standard [28]. One of the aims of qualitative assessment is to help decision-makers by classifying risk attributes (Fig. 2) into discrete categories that can be used for prioritization.

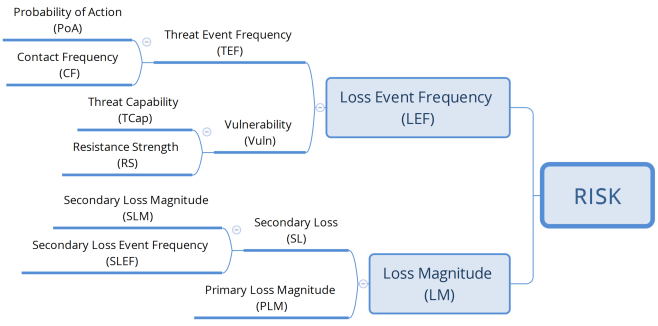

Fig. 2. Risk Attributes in [28]

The evaluation is based on a 5x5 risk matrix (Tab. I). The matrix uses a qualitative scale to determine which qualitative category the intersection of two attributes under consideration falls into. For example, Table I shows that if `Loss Magnitude (LM)` is medium (`M`) and `Loss Event Frequency (LEF)` is low (`L`), the calculated risk will fall into the low (`L`) category.

For each attribute, the qualitative categories are uniform: very low (`VL`), low (`L`), medium (`M`), high (`H`), very high (`VH`). The domain and the analyst determine which values for each attribute fall into each category.

TABLE I
O-RA - RISK MATRIX [28]

		Risk				
Loss Magnitude (LM)	*VH*	M	H	VH	VH	VH
	H	L	M	H	VH	VH
	M	VL	L	M	H	VH
	L	VL	VL	L	M	H
	VL	VL	VL	VL	L	M
		VL	*L*	*M*	*H*	*VH*
		Loss Event Frequency (LEF)				

Note that it is essential to consider how sensitive the risk evaluation outcome is to individual factors and expert consultation may be required to achieve a more accurate and reliable risk assessment. Additionally, parameters are needed to facilitate the expert estimation process, and these parameters may need to be adjusted based on the nature of the industry is evaluated.

C. Mitigations

The aim of this step is to analyze the potential measures that can be implemented in the system to minimize the impact of faults and attacks. In the context of dependability, these measures may include common fault tolerance patterns, such as redundancy, that are designed to ensure the system can continue to operate even if certain components fail. Additionally, they may also include preventative measures that aim to avoid faults from occurring, such as regular maintenance and system updates or mitigation measures to minimize the impact of system failures.

For the security evaluation, the Attack Scenario Space (which contains all potential attacks) serves as the input. By incorporating MITRE ATT&CK Mitigation, the aim is to generate a Mitigation Solution Space that includes all possible combinations of mitigation. The reasoning framework is then used to narrow the solution space and identify the best and most cost-effective mitigation solutions for a given attack scenario. The final outcome of this step is a set of mitigation that can effectively block the attack scenarios within the Attack Scenario Space.

D. Cost-Benefit Estimation and Optimization

In the cost-benefit estimation step, various cost and risk metrics are incorporated into the model to enable optimization. By doing so, for example, the most efficient mitigation can be identified. If there are constraints on the mitigation budget, the optimization engine can handle that as well. By assigning costs to the mitigation actions, the cost of mitigation can be compared to the potential losses, thus allowing for a cost-benefit analysis.

The optimization step can be integrated with the previous step. The strategic objective to minimize loss, given some budget constraints, is straightforward. However, the budget constraints from the investment and customers' perspectives are very different. As in a hardware/software system, the total cost of ownership includes maintenance; it also includes the maintenance of the protection.

The benefit of the optimization is a multi-phase strategy where the actions can be prioritized. For example, if a company has a limited budget let's first deal with the most potential and severe risk and later focus on the other ones.

The following optimization tasks can be described in the evaluation:

- Failure Impact/Cost: potential consequences of a cybersecurity breach or failure in terms of the cost to the organization (e.g., financial losses, reputational damage, and legal liabilities). When evaluating different mitigation strategies, it is essential to consider the potential impact of failure and the associated costs to determine the most effective countermeasure.
- Mitigation Cost: expense associated with implementing protective measures to reduce the likelihood or impact of a failure or cyberattack.
- Attack Cost: resources that an attacker must expend to successfully attack the system (e.g, time, cost of hardware or software)
- Most efficient attack/mitigation: a strategy that offers the greatest benefits in terms of risk reduction while also minimizing the overall cost.
- Constraint on the mitigation budgets: limitations on the amount of money that an organization can allocate to cybersecurity measures. organizations must carefully weigh the costs and benefits of different mitigation strategies to ensure that they are maximizing their resources and achieving the greatest possible protection.

V. UNCERTAINTIES IN RISK ASSESSMENT

In risk assessment and EPA, several uncertainty factors may be present in the analysis. One of these factors is that not all information about the system and its faults and vulnerabilities is known. Occasionally, especially in security analysis, attacks can change the *system's structure* in unforeseen ways. Typical examples include parasitic couplings between functionally independent elements in the system or side-channel attacks. Or can be difficult to determine the probability of failures or attacks, particularly as new vulnerabilities emerge or attack methods become more widespread. Another source of uncertainty is qualitative modeling, where the lack of numerical values in error propagation evaluation creates uncertainty due to abstraction.

A. Uncertainty in Risk Evaluation

A solution to deal with uncertainties and to support the evaluation is the use of approximate values and sensitivity analysis.

One solution for dealing with approximate values is Rough Set Theory (RST) [29], [30]. RST is a mathematical paradigm to deal with imprecise, inconsistent, incomplete, uncertain information and knowledge. The result of the RST approximation consists of three sets. One set is the positive region, which contains certain solutions of the approximation. The other set is the negative region, which contains elements that are certainly not elements of a solution. The third set

is the boundary region. In this region are the elements that cannot certainly be determined from the available information whether they are part of the solution.

RST enables the identification of viable solutions by determining the potential values of risk factors. This helps to narrow down the set of potential solutions, and by examining the boundary (uncertain) region, it is possible to filter out any spurious solutions.

On the other hand, sensitivity analysis examines how uncertain factors impact the output by altering its values. For instance, in the context of O-RA, `Loss Magnitude`, and `Loss Event Frequency` are factors used to determine `Risk`. Let's consider that the `Loss Event Frequency` is `Low (L)` If there is uncertainty in the factor `Loss Magnitude (LM)`, with `VL` or `L` being the possible values. It can be concluded that the output is not sensitive in `LM` as the calculated `Risk` remains `VL` for both potential input values. However, if `LM` is known to range between `L–VH`, the output will vary with each change, indicating that `Risk` is sensitive to the possible values of `LM`. If a sensitivity analysis reveals that a factor of the risk is sensitive, further evaluation is required.

B. Uncertainty in EPA

During the EPA, uncertainties may arise, which may affect the outcome - this way, the validity of the analysis - and lead to escaping faults. Uncertainties can originate from two primary sources [31]. Firstly, epistemic uncertainties express that parts of the analyzed system may be unknown to the domain expert. Secondly, aleatory uncertainties may arise from incorrect or incomplete modeling of the system or even from the non-deterministic operation (physical processes). The phenomenon of *error propagation* itself may be non-deterministic; for instance, if the propagation of data errors is influenced by intermediate data operations and faithful modeling of these is impossible at the given level of abstraction or would require an overly detailed model. In [32], we proposed an approach that extends the EPA by handling the uncertainties via RST.

VI. Hierarchical Evaluation

Risk assessment is an iterative process. The analyst first examines the system at a high level and then drills down from the critical points to examine details in a more refined model.

Our hierarchical evaluation approach (Fig. 3) allows the analyst first to examine the system at a high level and then refine the assets and threats to get a more detailed picture to support their analysis. At a high level, the analyst only wants to examine the system's assets in broad terms. For example, with this high level of analysis, the analyst can focus on high-risk problems. And with refinement, it is possible to refine the model further and evaluate the system more precisely and accurately.

In Figure 4, an example of asset refinement is presented, demonstrating how the model of a system can be refined. The high-level description outlines the system asset `Engineering Workstation`. At a more refined level, the model includes a more detailed representation of the components and the relation between them in terms of information, data, and attack flow (e.g., `E-mail Client` → `Browser` → `Infected Computer`). Finally, mitigation (e.g., `User Training, Endpoint Security`) can attach to the specific aspects of the model.

By providing more comprehensive and precise information on how a system processes faulty input, this information enables the detection of potential vulnerabilities and the creation of more effective mitigation strategies. Additionally, as many databases of known vulnerabilities are version-specific (e.g., CWE [13]), having a precise description of the components in a system can greatly facilitate the analysis.

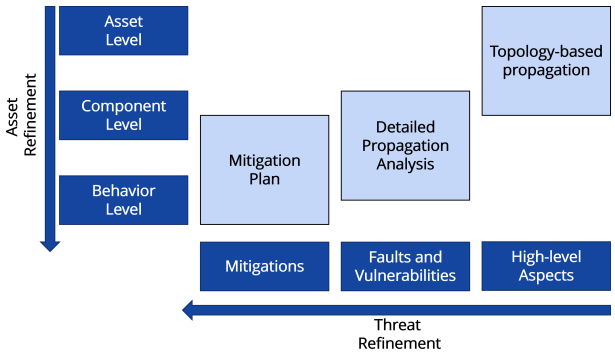

Fig. 3. Hierarchical Evaluation

The objective is to refine the system not only in terms of its assets but also in its threats. Therefore, a refinement strategy has been developed that introduces three threat refinement levels. The first level is concerned with high-level aspects such as reliability, availability, and timeliness. At the second level, specific faults and vulnerabilities in the system are identified. Finally, at the lowest level, mitigation mechanisms are introduced.

In Figure 3, a matrix integrates the refinements of assets and threats, with asset-type refinements arranged vertically from top to bottom and threat refinements arranged horizontally from right to left. Three key evaluation focuses have been identified for hierarchical evaluation.

1) *Topology-based propagation*: This method focuses on main assets and high-level aspects for a preliminary evaluation when detailed component information is unavailable. It is useful for early system development or initial risk assessments. After identifying critical subsystems/components, further investigation can refine the analysis and identify mitigation strategies.
2) *Detailed propagation analysis*: The aim of the analysis is to use the qualitative EPA to provide a detailed picture of the consequences of potential faults and vulnerabilities in the system. Besides the modeling of the information flow of the components, the behavior of the components can also be modeled to provide a more detailed analysis.
3) *Mitigation plan*: This analysis method can be used to create a plan for mitigating risk by adding mitigation

solutions to the model. Different cost metrics can be assigned to each mitigation strategy (e.g., financial costs). These costs can be used to optimize the mitigation plan. Additionally, the analysis method leverages the qualitative risk attributes identified in earlier steps. This approach ensures that the mitigation plan is tailored to the specific risks of the system and also takes into account the potential trade-offs.

At a high level of abstraction, the results obtained may contain spurious solutions as many aspects are over-generalized. As the solution is refined, the analyst can exclude these spurious solutions. For instance, in topology-based propagation analysis, only the consequences of attacks at a high level are known, with no knowledge of the component-level interactions. On the other hand, when creating a mitigation plan, the vulnerability of components can be determined based on their software version.

VII. Case Study

The case study is presented on a *water tank system* (Fig. 4). It was inspired by the Tennessee Eastman Process (TEP) [33] that is used to compare and benchmark fault and anomaly detection algorithms. Our example system consists of a main water tank component with input and output valve actuators and their respective controllers. The water tank includes a *water level sensor* that measures the water level in the tank. The water tank *controller* sends control messages to the *valve controllers* based on the measure of the water level sensor. The system also contains a *Human-Machine Interface* (HMI) where the operator can check the status of the system. The case study was first presented in [4]; however, we extended the model with specific IT aspects, including the *Engineer Workstation* where the user can manually reconfigure the input and output valve actuators.

The *safety requirements* for the system are 1) R1: the water tank should not be overflow; 2) R2: alert should be sent to the operator in case of water tank overflow. In this example, we consider the following possible failure modes of the components: 1) F1: Input Valve: `Stuck-at-Open` 2) F2: Output Valve: `Stuck-at-Closed` 3) F3: HMI: `No signal` 4) F4: Infected Computer can cause F1, F2, and F3.

On a high abstraction level, we can define high-level vulnerabilities and potential attacks on the component (e.g., Exploitation of Remote Services from the MITRE ATT&CK). To demonstrate hierarchical modeling, the Engineering Workstation has been extended. This finer decomposition describes a possible attack scenario where a user opens a link in a spam email and then downloads malware from the website, which infects the computer.

We can add mitigations (`M1: User Training, M2: Endpoint Security`) to the model to reduce the risk associated with each scenario. One mitigation solution against opening the link is user training. A potential solution against malware is the use of enterprise endpoint security. Different cost metrics can be attached to the mitigation solutions, such as their price or the time investment required to implement and execute each solution. Based on these metrics, the inference framework can optimize according to stakeholders' needs.

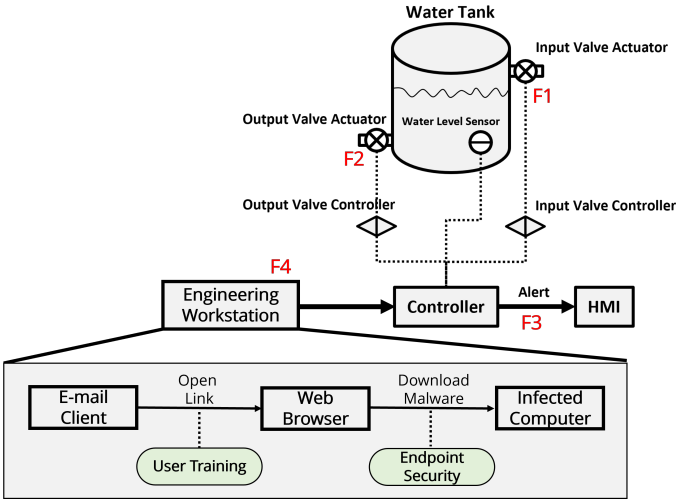

Fig. 4. Case Study Model

We used Archimate to model the system and the corresponding metadata, and then we transformed the model to Answer Set Programming to run the evaluation.

Listing 1 shows a code snippet that describes that if there is no active mitigation (`M`) for a fault (`F`) activated on a component (`C`) then the fault will be a potential fault considered in the evaluation. The engine selects the active faults from the potential fault set.

Listing 1. Fault Activation

```
potential_fault(C, F) :-
    component(C),
    fault(F),
    mitigation(F, M),
    not active_mitigation(C, M).
```

Listing 2 shows that the state of a component (`C`) does not change when the `stuck_at_x` fault mode is active. This way, all the faults modes on the model can be described.

Listing 2. Fault Model

```
component_state(C, X) :-
    prev_component_state(C, X),
    active_fault(C, stuck_at_x).
```

After the modeling part, the analysis checks the violations of the safety constraint for all the scenarios (combination of fault modes). The results of the evaluation can be examined in a form of a Jupyter Notebook.

Table II shows an extract (some of the fault mode combinations were neglected from the table) of the solution of the analysis of the extended model. For he fault modes the asterisk symbol represents the active fault modes. The second scenario (S2) shows the scenario where the attacker compromised the Engineering Workstation without any mitigation. This way, the propagation of the attack caused an active fault in the physical system as the attacker could directly reconfigure the actuators.

TABLE II
ANALYSIS RESULTS

	Fault Modes				Mitigations		Requirements	
	F1	F2	F3	F4	M1	M2	R1	R2
S1					Active	Active	-	-
S2				*			Violated	Violated
S3	*				Active	Active	-	-
S4		*			Active	Active	Violated	-
S5		*	*		Active	Active	Violated	Violated
S6	*		*		Active	Active	-	-
S7	*	*	*		Active	Active	Violated	Violated

If the analyst activates the potential mitigation in the model, it allows excluding this specific scenario from the evaluation.

Further analysis indicated that the most severe fault combination is when the output valve is stuck in the `closed` state, and the HMI does not get an `alert` message (S5). Both safety requirements (R1, R2) are violated in this case. If all physical fault modes (F1,F2,F3) are activated (S7), the analysis results in the same violation results, but the potential probability of the simultaneous occurrence of all faults is much lower.

VIII. CONCLUSION

The tool sketched in the paper shows that high-quality professional solutions can be achieved if a non-professional user gets sophisticated support by hiding the methodological complexity. The task's understandability and interpretability are the key elements of efficiency and quality [34].

The tool, which is still partly under development, with its simple but straightforward modeling language, reuse and integration of validated up-to-date internal and external information and workflow, and embedded exhaustive formal verification, can achieve reassuring quality results without learning, even for non-security domain system managers.

REFERENCES

[1] Joseph R. Biden Jr. Executive Order on Improving the Nation's Cybersecurity. [Online]. Available: https://www.whitehouse.gov/wp-content/uploads/2023/03/National-Cybersecurity-Strategy-2023.pdf

[2] European Commission. Proposal for a Regulation on cybersecurity requirements for products with digital elements - Cyber resilience Act. [Online]. Available: https://digital-strategy.ec.europa.eu/en/library/cyber-resilience-act

[3] K. D. Forbus, "Qualitative Process Theory," *Artificial intelligence*, vol. 24, no. 1-3, pp. 85–168, 1984.

[4] A. Földvári, G. Biczók, I. Kocsis, L. Gönczy, and A. Pataricza, "Impact Assessment of IT Security Breaches in Cyber-Physical Systems," in *2021 10th Latin-American Symposium on Dependable Computing (LADC)*. IEEE, 2021, pp. 1–4.

[5] Object Management Group, "SysML Extension for Physical Interaction and Signal Flow Simulation, v1.1," 2021.

[6] K. D. Forbus, "Qualitative Modeling," *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 2, no. 4, pp. 374–391, 2011.

[7] H. Jonkers, E. Proper, and M. Turner, "TOGAF™ and ArchiMate®: A future together," *White Paper W*, vol. 192, 2009.

[8] I. Band *et al.*, "How to model enterprise risk management and security with the ArchiMate language," *The Open Group white paper (W172)*, vol. 9, 2019.

[9] V. Lifschitz, *Answer Set Programming*. Springer Berlin, 2019.

[10] P. Cabalar, R. Kaminski, P. Morkisch, and T. Schaub, "telingo = ASP + time," in *International Conference on Logic Programming and Nonmonotonic Reasoning*. Springer, 2019, pp. 256–269.

[11] "Common Vulnerabilities and Exposures," https://cve.mitre.org/.

[12] FIRST.Org, "Common Vulnerability Scoring System (CVSS) Specification, Version 3.1," Forum of Incident Response and Security Teams (FIRST), Leesburg, VA, Technical Report, 2020. [Online]. Available: https://www.first.org/cvss/specification-document

[13] R. A. Martin, "Common weakness enumeration," *Mitre Corporation*, 2007.

[14] "CAPEC: Common Attack Pattern Enumeration and Classification," https://capec.mitre.org.

[15] W. Xiong, E. Legrand, O. Åberg, and R. Lagerström, "Cyber Security Threat Modeling Based on the MITRE Enterprise ATT&CK Matrix," *Softw. Syst. Model.*, vol. 21, no. 1, p. 157–177, feb 2022. [Online]. Available: https://doi.org/10.1007/s10270-021-00898-7

[16] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "MITRE ATT&CK: Design and philosophy," *Technical report*, 2018.

[17] O. Alexander, M. Belisle, and J. Steele, "MITRE ATT&CK® for industrial control systems: Design and philosophy," *The MITRE Corporation: Bedford, MA, USA*, 2020.

[18] S. Hou, X. Chen, J. Ma, Z. Zhou, and H. Yu, "An Ontology-Based Dynamic Attack Graph Generation Approach for the Internet of Vehicles," *Frontiers in Energy Research*, p. 808, 2022.

[19] A. Avizienis, J.-C. Laprie, and B. Randell, "Fundamental concepts of dependability," *Department of Computing Science Technical Report Series*, 2001.

[20] S. Kriaa, L. Pietre-Cambacedes, M. Bouissou, and Y. Halgand, "A survey of approaches combining safety and security for industrial control systems," *Reliability engineering & system safety*, vol. 139, pp. 156–178, 2015.

[21] A. J. Kornecki and M. Liu, "Fault tree analysis for safety/security verification in aviation software," *Electronics*, vol. 2, no. 1, pp. 41–56, 2013.

[22] A. J. Kornecki and J. Zalewski, "Safety and security in industrial control," in *Proceedings of the sixth annual workshop on cyber security and information intelligence research*, 2010, pp. 1–4.

[23] T. Novak and A. Gerstinger, "Safety-and security-critical services in building automation and control systems," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 11, pp. 3614–3621, 2009.

[24] I. N. Fovino, M. Masera, and A. De Cian, "Integrating cyber attacks within fault trees," *Reliability Engineering & System Safety*, vol. 94, no. 9, pp. 1394–1402, 2009.

[25] J. Jürjens, "Developing safety-and security-critical systems with uml," in *DARP workshop, Loughborough*, 2003.

[26] L. Apvrille and Y. Roudier, "Towards the model-driven engineering of secure yet safe embedded systems," *arXiv preprint arXiv:1404.1985*, 2014.

[27] International Electrotechnical Commission, "IEC 61508 Functional safety of electrical/electronic/programmable electronic safety-related systems," 2010.

[28] The Open Group, "Risk Analysis (O-RA) Standard, Version 2.0.1," 2019. [Online]. Available: https://pubs.opengroup.org/security/o-ra/

[29] Z. Pawlak, "Rough sets," *International journal of computer & information sciences*, vol. 11, no. 5, pp. 341–356, 1982.

[30] S. Akama, T. Murai, and Y. Kudo, "Reasoning with rough sets," *Logical Approaches to Granularity-Based Framework*, 2018.

[31] M. Yazdi, S. Kabir, and M. Walker, "Uncertainty handling in fault tree based risk assessment: State of the art and future perspectives," *Process Safety and Environmental Protection*, vol. 131, pp. 89–104, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957582019305555

[32] A. Földvári and A. Pataricza, "Handling Uncertainty in Error Propagation Analysis," in *30th PhD Minisymposium of the Department of Measurement and Information Systems*. Budapest University of Technology and Economics, 2023.

[33] A. Bathelt, N. L. Ricker, and M. Jelali, "Revision of the Tennessee Eastman Process Model," *IFAC-PapersOnLine*, vol. 48, no. 8, pp. 309–314, 2015.

[34] A. Pataricza, L. Gonczy, F. Brancati, F. Moreira, N. Silva, R. Esposito, A. Bondavalli, and A. Esper, "Cost estimation for independent systems verification and validation," *Certifications of Critical Systems-The CECRIS Experience*, p. 117, 2017.