Property Specification and Models for Risk: Towards Risk Propagation Graphs

Stefano M Nicoletti, Mattia Fumagalli, Milan Lopuhaä-Zwakenberg, E Moritz Hahn, Giancarlo Guizzardi, Mariëlle Stoelinga

Property Specification and Models for Risk: Towards Risk Propagation Graphs

Stefano M. Nicoletti* Mattia Fumagalli§ Milan Lopuhaä-Zwakenberg*
E. Moritz Hahn* Giancarlo Guizzardi‡ Mariëlle Stoelinga*†

*University of Twente, Formal Methods and Tools, Enschede, the Netherlands.
†Radboud University, Department of Software Science, Nijmegen, the Netherlands.
‡University of Twente, Semantics, Cybersecurity & Services, Enschede, the Netherlands.
§Free University of Bozen-Bolzano, In2Data and CORE, Bozen, Italy.
{s.m.nicoletti,m.a.lopuhaa,e.m.hahn,g.guizzardi,m.i.a.stoelinga}@utwente.nl, mattia.fumagalli@unibz.it

***Abstract*—Safety-critical infrastructures must operate safely and securely. Fault tree and attack tree analysis are widespread methods used to assess risks in these systems: fault trees (FTs) are required — among others — by the Federal Aviation Administration, the Nuclear Regulatory Commission, in the ISO26262 standard for autonomous driving and for software development in aerospace systems. Attack trees (ATs) are hierarchical diagrams that offer a flexible modelling language used to assess how systems can be attacked. ATs are widely employed both in industry and academia: they are referred to by many system engineering frameworks, e.g. *UMLsec* and *SysMLsec*, and are supported by industrial tools such as Isograph's *AttackTree*. In this paper we will briefly present advancements on *logics for property specification* on FTs and ATs and pitch the idea of an extended model that combines FTs and ATs: *risk propagation graphs*.**

I. Introduction

Our self-driving cars, power plants, aerospace infrastructures and transportation systems must operate in a safe and secure way. Risk assessment is a key activity to identify, analyze and prioritize the risk in a system, and come up with (cost-)effective countermeasures.

Fault tree analysis (FTA) [1, 2] is a widespread formalism to support risk assessment w.r.t. failure-related events. FTA is applied to many safety-critical systems and the use of fault trees is required for instance by the Federal Aviation Administration (FAA), the Nuclear Regulatory Commission (NRC), in the ISO 26262 standard [3] for autonomous driving and for software development in aerospace systems. A fault tree (FT) models how component failures arise and propagate through the system, eventually leading to system level failures. Leaves in a FT represent *basic events* (BEs), i.e. elements of the tree that need not be further refined. By events, we mean here "event type" as opposed to "event tokens". Henceforth, we use the term "event" simpliciter to refer to the former. Once these fail, the failure is propagated through the *intermediate events* (IEs) via *gates*, to eventually reach the *top level event* (TLE), which symbolizes system failure. The other side of the coin [4] in risk assessment is represented by security. Attack trees (ATs) are hierarchical diagrams that represent various ways in which a system can be compromised [5, 6]. Due to their popularity, ATs are referred to by many system engineering frameworks, e.g. *UMLsec* [7] and *SysMLsec* [8, 9], and are supported by industrial tools such as Isograph's *AttackTree* [10]. Much like FTs, the root — or *top level event* (TLE) — of an AT represents the attacker's goal, and the leaves represent *basic attack steps* (BASes): actions of the attacker that can no longer be refined. Intermediate nodes are labeled with gates that determine how basic actions of the attacker can propagate to reach higher-complexity elements in the attack. FTs and ATs that do not capture dynamic behaviours present only OR and AND gates — we call these *static fault trees* and *static attack trees* respectively: we will focus on these variants in the present contribution. Fig. 1 presents two simple examples of an AT (in red) and a FT (in violet): they respectively represent a way to break into a house and a way to be trapped into said house due to a fire breaking out and impossibiity to escape. Many extensions of ATs and FTs exist, in order to model more elaborate attacks/failure dynamics [6, 11, 12, 13, 14].

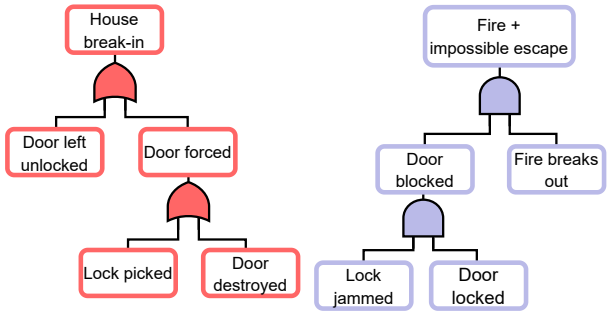

Figure 1: A simple AT (in red) and FT (in violet).

*This work was partially funded by the NWO grant NWA.1160.18.238 (PrimaVera), and the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 101008233, and the ERC Consolidator Grant 864075 (*CAESAR*).

Our approach: In previous work, we introduced tailor-made logics to specify powerful yet understandable analysis queries on FTs [15, 16] and ATs [17]. In this paper we briefly showcase these three logics:

1) A Boolean Logic for Fault Trees (*BFL*) [15],
2) A Probabilistic Logic for Fault Trees (*PFL*) [16],
3) A Logic for Quantitative Security Properties on Attack Trees (*ATM*) [17].

All of these three contributions present model checking algorithms that provide an automated procedure to verify that specified properties hold on the given models. Furthermore, we pitch the idea of a new model — *risk propagation graphs* — that combines and extends FTs and ATs to guarantee more expressive power and to favour joint safety-security risk analysis. The need for a model of this kind is clearly underlined by a recent publication [18] that highlights shortcomings of existent models for risk propagation via an ontological analysis. By proposing this model, we set the foundations for a comprehensive and quantitatively-informed risk analysis procedure. We aim to further enrich this procedure with a tailored logic for property specification on risk propagation graphs that rests on *BFL*, *PFL* and *ATM*.

Structure of the paper: Sec. II describes our Boolean logic for FTs, Sec. III showcases its probabilistic extension, Sec. IV presents our logic for quantitative security properties on ATs, Sec. V introduces *risk propagation graphs* and Sec. VI concludes the paper and reflects on future work.

II. Failures on Fault Trees: *BFL*

Fault tree analysis supports qualitative and quantitative analysis. Qualitative analysis aims at pointing out root causes and critical paths in the system. Typically, one identifies the *minimal cut sets* (MCSs) of a FT, i.e. minimal sets of BEs that, when failed, cause the system to fail. One can also identify *minimal path sets* (MPSs), i.e. minimal sets of BEs that - when not failed - guarantee that the system will remain operational. Quantitative analysis allows to compute relevant dependability metrics, such as the system reliability, availability and mean time to failure. *BFL* [15] is based on concrete insights and needs gathered through a series of questions targeted at a FT practitioner from industry [19]. The atomic propositions in this logic are the FT elements, i.e., both the BEs and the IEs. As usual, formulae can be combined through Boolean connectives. Furthermore, we include operators for setting evidence, and for MCSs and MPSs. In this way, we obtain a simple, yet expressive logic to reason about FTs that supports easier formulation of scenarios. With *BFL*:

1) We can set evidence to analyse what-if scenarios. E.g., what are the MCSs, given that BE A or subsystem B has failed? What are the MPSs given that A or B have not failed?
2) We can check whether two elements are independent or if they share a child that can influence their status.
3) We can check whether the failure of one (or more) element E always leads to the failure of TLE.
4) We can set upper/lower boundaries for failed elements. E.g., would element E always fail if at most/at least two out of A, B and C were to fail?

Moreover, if a property does not hold, *BFL* allows practitioners to generate *counterexamples*, to show why the property fails.

III. Fault Trees and Probabilities: *PFL*

As FTA requires the ability to perform both qualitative and quantitative analysis, *PFL* [16] extends the framework established with *BFL* allowing practitioners to reason about probabilities. With *PFL*:

1) We can check whether the probability of a given element (potentially conditioned by another one) respects a certain threshold,
2) We can set the value of one BE in complex formulae to an arbitrary probability value,
3) We can check if two BEs/IEs are stochastically independent,
4) We can also return probability values for given formulae, possibly mapping atoms to an arbitrary probability value.

Furthermore, [16] presents *LangPFL*, a domain specific language for *PFL* that propels the usability of this specification language by hiding some of the more complex aspects of our logic.

IV. Security Metrics on Attack Trees: *ATM*

ATs are often studied via *quantitative analysis*, during which they are assigned a wide range of security metrics [20, 6]. Such metrics are key performance indicators that formalize how well a system performs in terms of security and are essential when comparing alternatives or making trade-offs. Typical examples of such metrics are the minimal time [21, 22, 23, 24], minimal cost [25], or maximal probability [26] of a successful attack. With *ATM* [17] we constructed a more general framework that allows for property specification that considers these quantitative *security metrics*. More in detail, with *ATM*:

1) We can reason about *successful*/*unsuccessful* attacks;
2) We can check whether metrics, such as the cost, are bounded by a given value on single attacks;
3) We can compute metrics for a class of attacks and
4) perform quantification.

V. Towards Risk Propagation Graphs

FTs and ATs are indubitably useful in assessing risks on a given system. The expressive power that they offer, however, is not enough to allow a comprehensive analysis for joint safety-security risks. A survey on model-based techniques for joint safety-security risk assessment [14] reveals that models combining FTs and ATs constitute a promising starting point but are however still inadequate to capture all facets of safety-security interactions. This

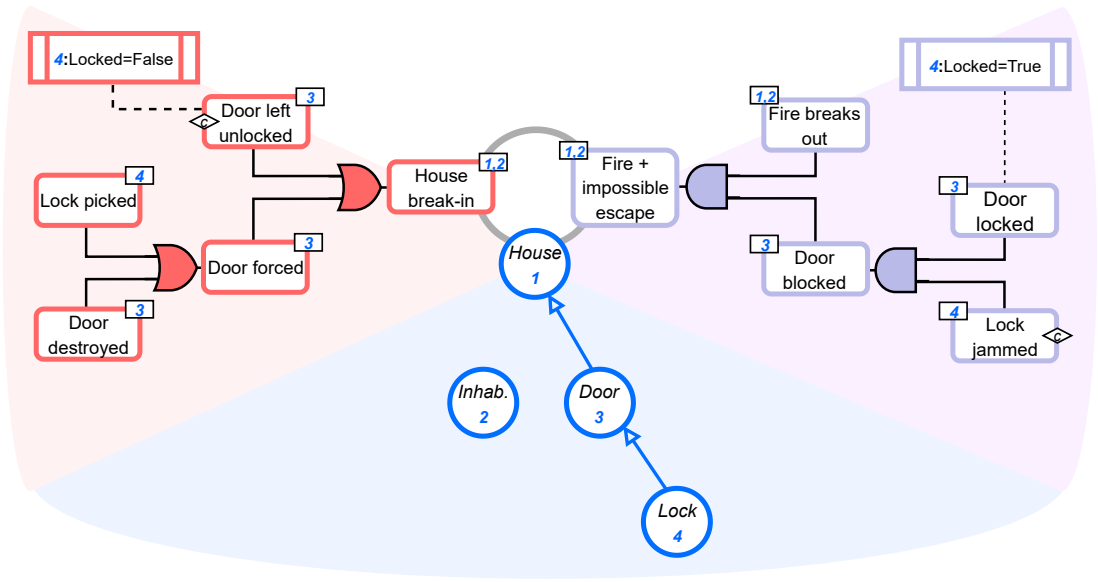

Figure 2: A risk propagation graph.

finding is further confirmed by a recent paper [18] that highlights shortcomings of current models capturing risk propagation mechanisms, FTs included. In fact, none of the formalisms considered in [18] can formulate queries such as:

1) "Infer the risk of an object, given the event(s) in which it participates";
2) "Infer the risk of an event, given the event(s) to which it is connected";
3) "Infer the risk of an object, given the object(s) to which it is connected";
4) "Infer the risk of an event, sharing an object with other events";
5) "Infer the risk of an object in an event with another object, which is in another event";
6) "Infer the risk of an event, given different properties characterising the correlated event".

In fact, a downside of, e.g., FTs is that they only consider events, and for a comprehensive study of risk propagation, both events, objects, and their interplay need to be considered. These shortcomings highlight the necessity of a new formalism that combines both the ease of modelling and familiarity of ATs and FTs with extended capabilities to model more complex scenarios and to query these elaborated models.

Risk propagation graphs: To this end, we sketch the idea of *risk propagation graphs* (RPGs): RPGs are graphs that incorporate elements from the ontological analysis in [18], while combining and enriching conventional FT and AT analysis. Fig. 2 showcases an example of an RPG: this graph is constructed starting from *objects at risk*, represented by blue circles. Relationships between these objects are constructed via edges that represent *Parthood*, e.g., *Lock* is part of *Door*, which is part of *House*. Once objects at risk are represented, one can draft *Threat Events* and *Loss Events*, i.e., events that represent risks concerned with intentional damage (*security*) and with unintentional failures/malfunctions (*safety*). Events are represented by rectangles with rounded edges. These events are detailed in a AT- and FT-like manner, starting from events that *involve* the *top level object* (in this case, *House*). Each event is then equipped with objects that it *involves* (listed in the top right of each event): for example, the event *House break-in* involves the objects *House* and *Inhabitant*. If an object is involved in an event, then the *parts* of this object are also involved in the same event. Furthermore, events can be *correlated* with each other (represented with a small diamond). Finally, events can be detailed with *Properties* (squared rectangles on the graph): states in which an object must be, in order for the event to occur. For example, for *Door left unlocked* to happen, the object *Lock* must not be in the state *Locked*. On the contrary, for *Door locked* to happen, *Lock* must be *Locked*.

Properties on RPGs: With this model, we allow for the formulation of queries that not only can reason about *objects* in addition to *events* but that can also aggregate both safety- and security-related risks on these objects, given certain states that characterize the system under analysis. E.g., with $Risk = Probability \times Impact$, one could formulate the following properties on Fig. 2:

1) What is the most risky (threat/loss) event that involves *Door*, given that *Lock* is *Unlocked*?
2) What is the total risk associated with *House*, given that *Lock* is *Locked*?
3) What is the optimal states configuration on all the objects at risk to minimise risk on *Inhabitant*?
4) What is the risk of *Door left unlocked*, given that *Lock* is *Unlocked*, and the correlated event *Lock jammed*?

VI. Conclusions and Future Work

We shortly presented advancements on logics for property specification on ATs and FTs, briefly showcasing how these languages can enhance analysis of these models. Furthermore, we sketched the idea of *risk propagation graphs*: models that combine ATs and FTs with extended expressivity, in order to tackle shortcomings highlighted by [14] and [18]. With such a model we will move towards joint safety-security co-analysis, without disregarding the need to reason about *objects* at risk, in addition to the central role of *events* typical of canonical AT and FT analysis.

Future work: This seminal step opens promising directions for future work. Firstly, we will render this sketch concrete by providing a formal definition of risk propagation graphs. This would allow us to reason more precisely about them and to have a formally sound basis to ground further work. Secondly, we will develop a formal logic to express properties of interest, here highlighted in Sec. V via natural language. This logic will ideally be a superset of the logics we presented in Sec. II, Sec. III and Sec. IV: by doing so, we will allow practitioners to express properties specific to risk propagation graphs, while at the same time granting the option to specify properties only on the ATs and FTs composing the RPG. Furthermore, starting from what we sketched in Sec. V, we will investigate how event propagation interacts with mereology. Lastly, model checking algoritms will be devised in order to verify specified properties on RPGs.

In summary:

1) We briefly introduce previous work on logics to specify properties on ATs and FTs, i.e., *BFL* [15], *PFL* [16] and *ATM* [17];
2) We point to recent works [14, 18] highlighting shortcomings of ATs and FTs, and underlining the need of a more expressive model-based formalism to reason about *risk*;
3) We propose a sketch of such a formalism, that we label *risk propagation graph*;
4) We highlight potential advantages of this model and specify some example properties in natural language;
5) We swiftly discuss future work, by envisioning a formal definition for RPGs, connections between event propagation and mereology, a logic that allows reasoning on RPGs, and the need for model checking algoritms.

References

[1] E. Ruijters and M. Stoelinga, "Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools," *Comp.Sci.Rev.*, vol. 15–16, pp. 29–62, 2015.

[2] M. Stamatelatos, W. Vesely, J. Dugan, J. Fragola, J. Minarick, and J. Railsback, "Fault tree handbook with aerospace applications," *NASA Office of Safety and Mission Assurance*, 2002.

[3] International Standardization Organization, "ISO/DIS 26262: Road vehicles, functional safety," https://www.iso.org/standard/68383.html, 2018.

[4] C. E. Budde, C. Kolb, and M. Stoelinga, "Attack trees vs. fault trees: two sides of the same coin from different currencies," in *QEST*, pp. 457–467, 2021.

[5] B. Schneier, "Attack trees," *Dr. Dobb's journal*, vol. 24, no. 12, pp. 21–29, 1999.

[6] M. Lopuhaä-Zwakenberg, C. E. Budde, and M. Stoelinga, "Efficient and generic algorithms for quantitative attack tree analysis," *IEEE TDSC*, 2022.

[7] J. Jürjens, "UMLsec: Extending UML for secure systems development," in *UML*, ser. LNCS, vol. 2460, pp. 412–425, 2002.

[8] L. Apvrille and Y. Roudier, "SysML-sec: A sysML environment for the design and development of secure embedded systems," in *APCOSEC*, 2013.

[9] Y. Roudier and L. Apvrille, "SysML-Sec: A model driven approach for designing safe and secure systems," in *MODELSWARD*, pp. 655–664. IEEE, 2015. ISBN 978-989-758-136-6

[10] Isograph, "AttackTree," https://www.isograph.com/software/attacktree/, Accessed Mar. 2023.

[11] D. Basgöze, M. Volk, J. Katoen, S. Khan, and M. Stoelinga, "Bdds strike back - efficient analysis of static and dynamic fault trees," in *(NFM)*, ser. LNCS, vol. 13260, pp. 713–732, 2022.

[12] H. Boudali, P. Crouzen, and M. Stoelinga, "Dynamic fault tree analysis using input/output interactive Markov chains," in *DSN*, pp. 708–717. IEEE Computer Society, 2007. DOI: 10.1109/DSN.2007.37

[13] M. Volk, S. Junges, and J. Katoen, "Fast dynamic fault tree analysis by model checking techniques," *IEEE Trans. Ind. Informatics*, vol. 14, no. 1, pp. 370–379, 2018. DOI: 10.1109/TII.2017.2710316

[14] C. Kolb, S. M. Nicoletti, M. Peppelman, and M. Stoelinga, "Model-based safety and security co-analysis: Survey and identification of gaps," *arXiv preprint arXiv:2106.06272*, 2021.

[15] S. Nicoletti, E. Hahn, and M. Stoelinga, "BFL: a Logic to Reason about Fault Trees," in *(DSN)*, pp. 441–452. IEEE/EUCA, 2022. DOI: 10.1109/DSN53405.2022.00051

[16] S. M. Nicoletti, M. Lopuhaä-Zwakenberg, E. M. Hahn, and M. Stoelinga, "Pfl: A probabilistic logic for fault trees," in *25th International Symposium on Formal Methods, FM 2023*, pp. 199–221. Springer Nature, 2023.

[17] S. M. Nicoletti, M. Lopuhaä-Zwakenberg, E. M. Hahn, and M. Stoelinga, "ATM: a Logic for Quantitative Security Properties on Attack Trees," 2023, manuscript under review.

[18] M. Fumagalli, G. Engelberg, T. P. Sales, Í. Oliveira, D. Klein, P. Soffer, R. Baratella, and G. Guizzardi, "On the semantics of risk propagation," in *International Conference on Research Challenges in Information Science*, pp. 69–86. Springer, 2023.

[19] Anonymized, "A Logic to reason about Fault Trees. Interview Report," URL removed for anonymization purposes.

[20] C. E. Budde and M. Stoelinga, "Efficient Algorithms for Quantitative Attack Tree Analysis," in *(CSF)*, pp. 1–15, 2021.

[21] R. Kumar, E. Ruijters, and M. Stoelinga, "Quantitative Attack Tree Analysis via Priced Timed Automata," in *FORTE*, ser. LNCS, vol. 9268, pp. 156–171. Springer International Publishing, 2015. DOI: 10.1007%2F978-3-319-22975-1_11

[22] F. Arnold, H. Hermanns, R. Pulungan, and M. Stoelinga, "Time-dependent analysis of attacks," in *POST*, ser. LNCS, vol. 8414, pp. 285–305, 2014.

[23] R. Kumar, S. Schivo, E. Ruijters, B. Yildiz, D. Huistra, J. Brandt, A. Rensink, and M. Stoelinga, "Effective Analysis of Attack Trees: A model-driven approach," in *FASE*, ser. LNCS, vol. 10802, pp. 56–73, 2018. DOI: 10.1007/978-3-319-89363-1_4

[24] M. Lopuhaä-Zwakenberg and M. Stoelinga, "Attack time analysis in dynamic attack trees via integer linear programming," *arXiv e-prints*, vol. abs/2111.05114, 2021.

[25] F. Arnold, D. Guck, R. Kumar, and M. Stoelinga, "Sequential and Parallel Attack Tree Modelling," in *SAFECOMP*, ser. LNCS, vol. 9338, pp. 291–299. Springer International Publishing, 2015. DOI: 10.1007/978-3-319-24249-1_25

[26] R. Jhawar, B. Kordy, S. Mauw, S. Radomirović, and R. Trujillo-Rasua, "Attack Trees with Sequential Conjunction," in *SEC*, ser. IFIPAICT, vol. 455, pp. 339–353, 2015.