Assessing the Understandability and Acceptance of Attack-Defense Trees for Modelling Security Requirements

Giovanna Broccia[1], Maurice H. ter Beek[1], Alberto Lluch Lafuente[2], Paola Spoletini[3], and Alessio Ferrari[1]

[1] ISTI-CNR, Pisa, Italy
{giovanna.broccia, maurice.terbeek, alessio.ferrari}@isti.cnr.it
[2] DTU, Lyngby, Denmark
albl@dtu.dk
[3] Kennesaw State University, GA, USA
pspoleti@kennesaw.edu

Abstract. *Context and Motivation* Attack-Defense Trees (ADTs) are a graphical notation used to model and assess security requirements. ADTs are widely popular, as they can facilitate communication between different stakeholders involved in system security evaluation, and they are formal enough to be verified, e.g., with model checkers. *Question/Problem* While the quality of this notation has been primarily assessed quantitatively, its understandability has never been evaluated despite being mentioned as a key factor for its success. *Principal idea/Results* In this paper, we conduct an experiment with 25 human subjects to assess the understandability and user acceptance of the ADT notation. The study focuses on performance-based variables and perception-based variables, with the aim of evaluating the relationship between these measures and how they might impact the practical use of the notation. The results confirm a good level of understandability of ADTs. Participants consider them useful, and they show intention to use them. *Contribution* This is the first study empirically supporting the understandability of ADTs, thereby contributing to the theory of security requirements engineering.

Keywords: security requirements · Attack-Defense Trees · understandability evaluation · empirical user study · Method Evaluation Model

1 Introduction

The definition of security requirements entails the representation and analysis of envisioned threats and mitigation solutions, oriented to eventually define a security policy [10]. Several notations have been proposed in requirements engineering (RE) to model and analyse security requirements, such as extensions of well-known notations (e.g., Secure I* [21] and Secure UML [22]) and other comprehensive notations with analysis capabilities (e.g., the Socio-Technical Security Modelling Language (STS-ML) [29] and the Restricted Misuse Case Modeling (RMCM) approach [23]).

Among this variety of proposals, Attack-Defense Trees (ADTs) offer a graphical notation used to model and assess the security requirements of systems or assets. They provide a representation of possible actions an attacker might take to attack a system and the measures that a defender can employ to protect the system [15]. The purposes of ADTs are multiple. In addition to providing a threat modelling methodology, they can be used for quantitatively assessing the security of a system (e.g., with model checking). Moreover, ADTs are useful for facilitating communication between stakeholders from different fields and with different backgrounds (e.g., domain experts, security experts).

Several studies have shown how graphical notations are more comprehensible by humans than textual notations [32,35]. However, although ADTs have been claimed as one of the most popular graphical models for system security analysis [11], extremely easy to use also for novice users [38], and as an easily understandable human-readable notation [8], no user study has been proposed to verify these hypotheses. Albeit this research direction holds promise and would be helpful in evaluating their effectiveness [11,19,20]. Indeed, beyond the realm of attack trees, there exists a substantial body of empirical research literature focused on security modelling and assessment [6,17,18]. These kinds of studies are particularly beneficial given the centrality of humans in system security—both for possible insider attacks and for human errors that make the system vulnerable [8].

In this paper, we present the first experiment that aims at investigating the quality of the ADT notation, both in terms of understandability and in terms of user acceptance. We designed the study based on the Method Evaluation Model (MEM) [27], a model used to evaluate information technologies, which extends the Technology Acceptance Model (TAM) [7]. We adapt MEM following the approach by Abrahão [1] and identify two classes of variables: performance-based and perception-based. The performance-based variables aim at assessing the understandability of ADTs, while perception-based variables seek to evaluate the users' acceptance of ADTs.

Our results show that: (1) ADTs are sufficiently understandable; (2) ADTs are perceived as easy to use and useful, and participants express the intention to use them; (3) there is a relationship between perceived usefulness and intention to use; (4) there are no significant relationships between various performance-based measures of understandability (effectiveness and efficiency) and perception-based variables (ease of use, usefulness, intention to use), except in the following cases: (a) perceived ease of use has a positive relationship with effectiveness, i.e., those who make fewer mistakes in different ADT understandability tasks generally consider the notation easier; (b) those who *apply* the method better in practice also consider it more useful; (c) those who make fewer mistakes when *observing* the notation used in realistic contexts, consider the method easier. Our replication package is publicly available [4].

Related Work. Several notations have been proposed in RE to model and analyse security requirements [26,13,34,37]. Some of these notations are extensions of

existing notations, like Secure I* [21], KAOS [30]), Secure UML [22], Misuse cases [33], and Secure Tropos [12].

Other attempts, some based on the languages above, also offer analysis capability. In particular, the ones mentioned in the Introduction. STS-ML [29] is an actor- and goal-oriented security requirements modelling language based on Tropos, able to capture system security needs and requirements at the organisational level and reason about corporate assets, social dependencies, and trust properties. RMCM [23] is a use case-driven modelling method that uses misuse case diagrams [33] to support the specification of security and privacy requirements of multi-device software ecosystems in a structured and analysable form. The Risk-based Security Requirements (RBSR) model [9] associates security requirements with specific weaknesses and risk profiles that can vary over time and provides mitigation accordingly to these variations. Finally, [39] introduces a threat-based security framework and its Business Process Model and Notation (BPMN) extension to model the security threat and support risk analysis.

Labunets et al. observed a difference in the representation of security risk assessment between academic proposals and industry standards. Academic approaches favour graphical notation, while the industry leans towards tabular models. Several studies were conducted to compare the effectiveness of graphical and tabular models. [17] proved that both methods are equally effective. In [18], a comparative analysis of visual and textual risk-based approaches revealed that the visual method is more effective for identifying threats, while the textual method is slightly better for eliciting security requirements.

In [19], the results of an empirical evaluation conducted to determine the effectiveness of two attack modelling techniques, an adapted attack graph method and the fault tree standard, are reported. The results indicate that the attack graph method is more effective than the fault tree method.

2 Attack-Defense Trees

The assessment of system security through graphical tree structures originated in 1960 with fault tree analysis [36], and gradually spread with the usage of similar structures such as attack trees [31,24]. To manage the dynamic nature of system security, Attack-Defense Trees (ADTs) [15] were introduced, extending attack trees with defense strategies and quantitative risk assessment [14,3]. ADTs model attack-defense scenarios, namely 2-player games between a proponent and an opponent.

Formally, ADTs are rooted trees with labelled nodes of two opposite types: attack nodes and defense nodes, representing the goals of the attacker and the defender, respectively. The root can be either type: if the root is an attack node, the proponent is an attacker; conversely, if the root is a defense node, the proponent is a defender. The main goal can be refined into sub-goals, described by its child nodes of the same type. The refinement can be either conjunctive (i.e., all sub-goals must be achieved to achieve the parent goal) or disjunctive (i.e., at least one of the sub-goals must be achieved to reach the parent goal).

A node with no children of the same type is called a non-refined node, and it represents a basic/atomic action. Each node may have one child of the opposite type, representing a countermeasure to its (sub-)goal. Essentially, an attack node may have a number of children that refines the attack and a single defense node that fends it off. Conversely, a defense node may have a number of children which refines the defense, and a single attack node that counterattacks it.

To demonstrate the features of ADTs, we present a simple fictitious scenario describing the theft of the Mona Lisa painting (cf. Fig. 1). To steal the painting, two kinds of attacks can be carried out: enter the Louvre museum by the door or by the window. Figure 1 shows in detail only the door branch (further attacks and defenses could easily be added). To secure the door, the museum can use an alarm; however, the attacker can perform a counterattack by forcing the alarm system. To do so, the attacker needs to get both the username and the password.

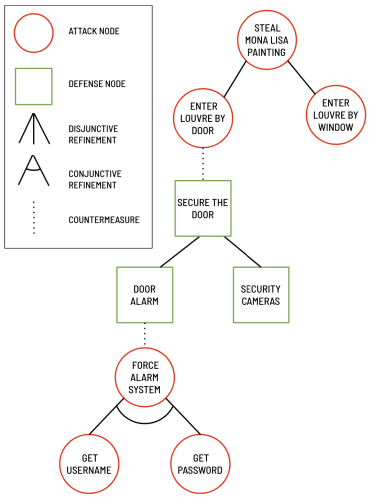


Fig. 1: ADT for theft of Mona Lisa.

Evaluation of ADTs has so far considered issues like the consistency between an ADT and the system and the impact of repeated labels on results [2,16]. As far as we know, there is no work in the literature that has focused on the assessment of the comprehensibility of ADTs (neither of attack trees). Albeit their comprehensibility is usually assessed as a factor of success [8,38,11].

3 Method Evaluation and Technology Acceptance Model

The Method Evaluation Model (MEM) [27] is a model used to evaluate new information technologies. According to MEM, the usage of new technologies is influenced by a set of *perception-based* variables and *performance-based* variables.

The perception-based variables are used to gauge the level of *acceptance* of the technology and include the perceived ease of use (PEOU), which measures how easy the technology is perceived to be, the perceived usefulness (PU), which measures how useful the technology is perceived to be, and the intention to use (ITU), which measures the extent to which users intend to use the technology in the future. The performance-based variables consist of efficiency and effectiveness, which measure the effort required to use the technology and how well the technology has been used to reach the goals, respectively. Essentially, the adoption of a new technology depends not only on whether it is actually effective but also on whether the users perceive it to be effective.

MEM has been applied in the fields of RE [1] and language comprehension [5]. In both studies, the performance-based variables (efficiency and effectiveness) have been adapted to measure the *understandability* of requirement models and language constructs, respectively. In practice, the performance-based variables are understandability effectiveness and understandability efficiency, computed based on the results obtained by sample subjects in problem-solving tasks. This paper adopts this approach and further decomposes the variables into fine-grained dimensions (cf. Sect. 4). In line with MEM, we evaluate if these variables are related to perception-based variables.

4 Study Design

Our experiment aims to study the degree of ADTs understandability and users' acceptance. We also study if there is a relationship between the degree of acceptance of the notation and its understandability.

4.1 Variables, Research Questions, and Tests

Acceptance and Understability Dimensions. Users' acceptance is based on the MEM model presented in Section 3. In particular, we evaluate *acceptance* using the three perception-based variables from the MEM (PEOU, PU, and ITU).

Understandability is evaluated in terms of effectiveness and efficiency based on the results of sample subjects in some problem-solving tasks (as suggested by the literature, e.g., [28]). For both effectiveness and efficiency, we further distinguish between fine-grained understandability, which considers three different dimensions of understandability separately, and coarse-grained understandability, which measures the average across the dimensions. The dimensions are:

UNC Understandability not in context measures the comprehensibility of ADTs *syntax*. It assesses users' ability, after ADT training, to identify correct ADT construction, recognise nodes (for attack and defense), refinements (conjunctive and disjunctive), countermeasures, and understand sequential actions and their temporal order in ADTs.

UIC Understandability in context measures the comprehensibility of ADTs *semantics*. It assesses users' ability, after training, to answer questions about both existing and instantiated ADTs and to recognise if an ADT accurately models a specific behaviour in a given scenario.

TRF Transferability measures the practical use of the notation, evaluating users' ability, after training, to create or modify ADTs. This includes recognising the appropriate elements to add to the tree for modelling specific behaviour and knowing where to place these elements.

Research Questions. We aim to answer the following research questions:

RQ1 *How well users understand ADTs?* This RQ aims to understand the level of effectiveness and efficiency with which users comprehend ADTs.

RQ2 *What is the degree of acceptance of ADTs by users?* This RQ aims to understand how much users perceive the notation as easy to use and useful and to what extent they intend to use ADTs in the future.

RQ3 *What is the relationship between ease of use/usefulness of the notation and intention to use it in the future?* Differently from RQ1, which focuses on each perception-based variable independently, this RQ aims at checking whether there is a relationhip among the variables, and in particular, if ease of use and usefulness are related to intention to use.

RQ4 *What is the relationship between the overall ADT understandability and the users' perception of ADTs' ease of use and usefulness?* With this RQ, we check whether users who perform best in understanding the notation also tend to evaluate the ADTs as easier and more useful.

RQ5 *What is the relationship between the different dimensions of understandability and the users' perception of ADTs' ease of use and usefulness?* Here, we want to check if there is an understandability dimension that is related to the perception of users in terms of ease of use and usefulness.

Variables for Acceptance and Understandability. We measure the three perception-based variables (PEOU, PU, and ITU) through an instrument adapted from MEM [27], namely a questionnaire composed of a set of statements for each variable. We shuffle the statements and add their negated version to avoid systematic response bias (i.e., both the statements "ADTs are easy to learn" and "ADTs are not easy to learn" are present) [1]. Users need to evaluate each statement on a Likert scale from 1 (*strongly agree*) to 5 (*strongly disagree*). Table 1 shows the list of positive statements for PEOU, PU, and ITU. Each variable is computed as the mean of its statements points (the points for negative statements are counted as 6 minus the points given as the answer).

Understandability dimensions are measured through specific tasks:

1. UNC is measured through a set of true/false questions on domain-agnostic ADT fragments (A, B, C instead of names), to ensure that users' responses are not influenced by knowledge of the domain.
2. UIC is evaluated through a set of yes/no questions on instantiated ADTs fragments.
3. TRF is measured through a number of instantiated ADTs fragments to extend with a set of requests.

For each of these dimensions, we compute effectiveness as the number of correct answers over the number of questions and efficiency as effectiveness over time [1]. Therefore, we have six different variables: *UNC effectiveness*, *UNC efficiency*, *UIC effectiveness*, *UIC efficiency*, *TRF effectiveness*, and *TRF efficiency*. For what concerns total understandability, we compute *understandability effectiveness* as the mean of the effectiveness of the three dimensions and *understandability efficiency* as the mean of the efficiency of the three dimensions.

Hypothesis Testing. To answer the research questions, we test a number of NULL hypotheses (cf. Table 2). Not all the combinations of variables are considered,

Table 1: Perception-based statements (positive statements).

	Statements
PEOU	1. It was easy for me to understand what the ADTs represented.
	2. ADTs are simple and easy to understand.
	3. ADTs are easy to learn.
	4. Overall, the ADTs were easy to use.
PU	1. Overall, I think that ADTs provide an effective means for describing security threats and countermeasures.
	2. I believe that ADTs have enough expressiveness to represent security threats and countermeasures.
	3. Overall, I find ADTs to be useful.
	4. I believe that ADTs are useful for representing security threats and countermeasures.
	5. Using ADTs would improve my performance in describing security threats and countermeasures.
	6. I believe that ADTs are organised, clear, concise, and unambiguous.
	7. I believe the use of ADTs would reduce the time required to represent security threats and countermeasures.
ITU	1. If I were to work for a company in the future, I would use ADTs to specify security threats and countermeasures.
	2. I intend to use ADTs in the future if given the opportunity.
	3. I would recommend the use of ADTs to security practitioners.
	4. It would be easy for me to become skilled in using ADTs.

following the approach by Abraõ [1], who relates ITU to PU and PEOU only, and not to performance-based variables.

4.2 Study Phases

The study is conducted online (material in [4]) and structured in 6 phases.

Phase 1 – Recruitment. Participants are contacted through a recruitment e-mail with all the information needed to perform the study. Specifically, links to a video training, a spreadsheet file where to get their identifier and the link to the test, the pre- and post-test questionnaires, the consent form, and study instructions.

Phase 2 – Binding. To ensure anonymity, participants are provided with a unique alphanumeric identifier via a spreadsheet file with a link to their test document (there is a different document for each participant). They are instructed to keep the identifier for the entire test, preserve the link to the test document to be used in a subsequent phase, and use incognito mode to protect their identity.

Phase 3 – Training. Before starting the test, we ask participants to watch a video that presents the ADT notation. The video is available online (https://youtu.be/KLIH-yultgI) and it contains all the information needed to complete the test. Participants are asked to use this support only once before they begin the test.

Phase 4 – Pre-test questionnaire. We ask participants to fill out an online questionnaire whose link has been sent by e-mail during the recruiting phase. The questionnaire collects information about gender, age, education, employment, work area, level of knowledge of ADTs, and education on ADTs. Participants

Table 2: Hypotheses for each research question.

RQ1	$\mathbf{H1_0}$	Users are not effective in understanding ADTs
	$\mathbf{H2_0}$	Users are not efficient in understanding ADTs
RQ2	$\mathbf{H3_0}$	ADTs are perceived as difficult to use
	$\mathbf{H4_0}$	ADTs are perceived as not useful
	$\mathbf{H5_0}$	There is no intention to use the ADT in the future
RQ3	$\mathbf{H6_0}$	There is no relationship between perceived ease of use and perceived usefulness
	$\mathbf{H7_0}$	There is no relationship between perceived usefulness and intention to use
	$\mathbf{H8_0}$	There is no relationship between perceived ease of use and intention to use
RQ4	$\mathbf{H9_0}$	There is no relationship between understandability effectiveness and perceived ease of use
	$\mathbf{H10_0}$	There is no relationship between understandability effectiveness and perceived usefulness
	$\mathbf{H11_0}$	There is no relationship between understandability efficiency and perceived ease of use
	$\mathbf{H12_0}$	There is no relationship between understandability efficiency and perceived usefulness
RQ5	$\mathbf{H13_0}$	There is no relationship between understandability not in context effectiveness and perceived ease of use
	$\mathbf{H14_0}$	There is no relationship between understandability not in context effectiveness and perceived usefulness
	$\mathbf{H15_0}$	There is no relationship between understandability not in context efficiency and perceived ease of use
	$\mathbf{H16_0}$	There is no relationship between understandability not in context efficiency and perceived usefulness
	$\mathbf{H17_0}$	There is no relationship between understandability in context effectiveness and perceived ease of use
	$\mathbf{H18_0}$	There is no relationship between understandability in context effectiveness and perceived usefulness
	$\mathbf{H19_0}$	There is no relationship between understandability in context efficiency and perceived ease of use
	$\mathbf{H20_0}$	There is no relationship between understandability in context efficiency and perceived usefulness
	$\mathbf{H21_0}$	There is no relationship between transferability effectiveness and perceived ease of use
	$\mathbf{H22_0}$	There is no relationship between transferability effectiveness and perceived usefulness
	$\mathbf{H23_0}$	There is no relationship between transferability efficiency and perceived ease of use
	$\mathbf{H24_0}$	There is no relationship between transferability efficiency and perceived usefulness

have to mark the questionnaire with the identifier received during the binding phase (Phase 2).

Phase 5 – Test. We ask participants to fill out the test in all its phases. The test is accessible through the link received during the binding phase (Phase 2); such a link leads to an editable online document (a different document for each participant). The spreadsheet accessed in Phase 2 enables us to bind each document to the ID of the corresponding user. The test is composed of 4 steps:

i **Retention.** Retention measures the comprehension of the training material and the ability to retain knowledge from it. We use this step to keep in the participants' memory the concepts presented in the training video that they will need during the test. The outcome of this step is not utilised in the calculation of understandability. In this step, a list of figures (i.e., all figures in the legend of Fig. 1) is presented and, for each figure, a table with two definition options. Participants are asked to mark the right definition for each figure.

ii **Understandability not in context.** With this step, we want to get how understandable is the syntax of the notation for the participants. In this step, 6 items are presented, and for each of them, we show one or more attack-defense tree fragments and 4 statements. Participants have to check for each of the statements whether it is true or false. Participants are asked to write down the starting (when starting step ii) and finishing time (when completing all the steps).

iii **Transfer.** Transfer measures how much is transferable the knowledge acquired through the training material. In this step, three attack-defense tree fragments are presented and, for each of them, a list of three requests. Par-

ticipants are asked to modify the tree fragments according to the requests using an editable diagram embedded in the document (the instructions to modify the diagram are written inside the diagram itself). The three ADT fragments used represent common and familiar types of attacks, namely an attack on a bank account, an attack to open a safe lock, and an attack to burgle a house. For each fragment, three requests were made, each with increasing levels of difficulty: (i) participants are asked to add a node to the tree and specify the type of node and its position; (ii) participants are asked to add all the nodes necessary to model a given situation; (iii) participants are asked to modify the tree according to given syntactic and/or semantic constraints. For each of the three items, participants are asked to write down starting and finishing times in the appropriate lines.

iv **Understandability in context.** With this step, we want to perceive to what extent users, after a training phase on ADTs, are able to answer questions about given ADTs. In this step, three attack-defense tree fragments are presented, and, for each of them, a list of three yes/no questions. Participants are asked to answer the questions by typing in the document "yes" or "no". The three ADT fragments used are extended versions of the fragments used in the Transfer step (cf. step iii). For each of the three items, participants are asked to write down starting and finishing times in the appropriate lines.

Users are not bound by a specific time frame for the test phase, but allocating 40 minutes is deemed sufficient for completing phases ii, iii, and iv (according to the authors ter Beek and Lluch Lafuente, who are ADT experts [3]). This duration considers the time required for reading and analysing questions, processing ADT fragments, providing accurate answers, and adapting to the platform used.

Phase 6 – Post-test questionnaire. We ask participants to fill out an online questionnaire whose link has been sent by e-mail during the recruiting phase. We use this phase to measure the perception-based variables (namely, PEOU, PU, and ITU) through a set of statements users need to rate from 1 to 5. The questionnaire contains 8 statements concerning PEOU, 14 statements on PU, and 8 statements concerning ITU (see Table 1). Participants have to mark the questionnaire with the identifier received during the binding phase (Phase 2).

5 Study Execution

The experimental study protocol containing the definition of the study phases, its rationale, as well as the data analysis process has been submitted to the ethical committee of the Italian National Research Council (CNR), which authorised the administration of the test. To take part in the study, participants are asked to sign an informed consent for the processing of personal data.

Participants. In total, 25 participants took part in the study: computer science students, Ph.D. students, and professors; researchers in the field of software engineering, formal methods, and security; participants belong to Kennesaw State

Table 3: Descriptive statistics.

Variables	*Median*	*Mean*	*Std. dev.*	*Min.*	*Max.*
PEOU	4.25	4.18	0.563	2.875	5
PU	4	3.92	0.37	2.929	4.571
ITU	3.875	3.88	0.403	3	5
UNC effectiveness	0.750	0.783	0.083	0.625	0.958
UNC efficiency	0.094	0.103	0.046	0.024	0.188
UIC effectiveness	0.889	0.907	0.175	0.111	1
UIC efficiency	0.250	0.264	0.135	0.009	0.500
TRF effectiveness	0.667	0.613	0.267	0	1
TRF efficiency	0.023	0.026	0.015	0	0.049
understandability effectiveness	0.792	0.768	0.134	0.287	0.986
understandability efficiency	0.118	0.131	0.059	0.011	0.241

University, CNR, University of Pisa, and the Technical University of Denmark. Participants in the study were selected opportunistically based on their availability. They were of both genders (56% men, 40% women, 4% prefers not to answer), aged between 21 and 56 years old. We asked them to self-evaluate their knowledge of ADTs before the test on a 5-point scale from 1 (*no knowledge*) to 5 (*advanced*) and whether they knew similar notations. The results are reported in Figures 2 and 3, respectively. A total of 80% of the participants did not receive any education on ADTs before the test; the remaining participants attended a university course, a seminar, or self-educated.

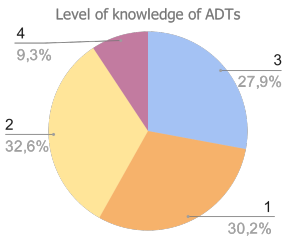


Fig. 2: Level of knowledge of ADTs.

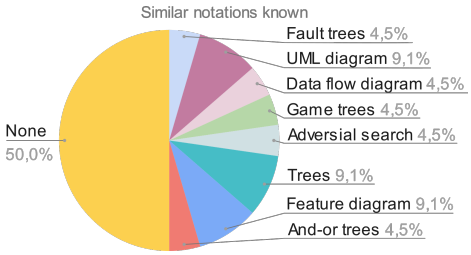


Fig. 3: Similar notations known.

Results. Table 3 shows descriptive statistics for all the variables gathered with the test, i.e., the perception-based variables (PEOU, PU, and ITU) and the performance-based variables: (1) understandability not in context effectiveness and (2) efficiency; (3) understandability in context effectiveness and (4) efficiency; (5) transferability effectiveness and (6) efficiency; and (7) understandability effectiveness and (8) understandability efficiency.

The perception-based variables are all above the average value of the Likert scale (i.e., 3), which thus suggests a general degree of acceptance of the notation.

The results indicate that users perceive ADTs as easy to use (mean score of 4.18) and as useful (mean value 3.92). Results also suggest that users intend to use the notation in the future (ITU has an average score of 3.88).

The results indicate a generally good level of understandability of ADT notation with an average total understandability effectiveness above 0.76, meaning that $\sim 77\%$ of the questions of the test are correctly answered.

Regarding the different dimensions composing understandability, the results show that understandability in context is the measure that provides the highest contribution (average effectiveness of 0.907), followed by understandability not in context (effectiveness = 0.783) and transferability (effectiveness = 0.613). This suggests that while participants understand the syntax and semantics of ADT fragments, they have more difficulty applying them in practice. For what concerns efficiency, we observe a similar trend, thereby confirming that ADTs "in action" are perceived as more difficult.

Table 4 summarises the relation between variables expressed in the hypotheses addressing each RQs presented in Section 4. For each hypothesis, the column "Reject" reports a "T" if the hypothesis has been rejected and an "F" otherwise. Below we discuss in detail only the rejected NULL hypotheses because no conclusions can be made for the others.

RQ1. To answer RQ1, we applied a Wilcoxon signed rank test to check whether effectiveness and efficiency are significantly above the target values of 0.6 (indicating a sufficient performance according to ter Beek and Lluch Lafuente, two ADT experts [3]) and of 0.015 (i.e., 60% of 1 (maximum effectiveness)/40 min (expected completion time)), respectively. We apply a non-parametric test (i.e. Wilcoxon signed rank) because the normality check, performed with the Kolmogorov-Smirnov test, fails for all the variables with p-value well below the 0.05 significance level. The test results show that both variables are significantly higher than the target values for $\alpha = 0.05$, with p-values of 0.000139 and 7.381e-06, respectively, with large effect-size (cf. Table 4). Therefore rejecting $H1_0$ and $H2_0$, and attesting a **sufficient overall understandability of the ADT notation**. Fine-grained effectiveness and efficiency measures are all significantly greater than the respective reference values for both effectiveness and efficiency, with the exception of transferability; we refer to [4] for detailed information.

RQ2. To answer RQ2, we applied a Wilcoxon signed rank test to check whether PEOU, PU, and ITU are significantly above the average value of the Likert scale (i.e., 3). The test results show that all the variables attesting the acceptance are significantly higher than 3 for $\alpha = 0.05$, with p-values of 1.077e-05, 7.109e-06, and 9.282e-06, respectively, with large effect-size (cf. Table 4). Therefore rejecting $H3_0$, $H4_0$, and $H5_0$ and confirming the overall degree of acceptance of the ADT notation as high. As the boxplot in Figure 4 shows, while ITU and PU have comparable values, PEOU receives the highest score. This suggests that **ease of use is the main characterising quality of ADTs**.

RQ3. To answer RQ3, we fit a regression linear model between PEOU and PU, and between both PEOU and PU and ITU. As shown in Figure 5a, the test results attest that there is a significant positive relationship between PU

Table 4: Statistics summary.
Blue rows indicate NULL hypotheses that have been rejected (p-value < 0.05)
The term "effv" indicates effectiveness and the term "effc" indicates efficiency.

RQs	*Hyp.*	*Variables*	*Reject*	*p-value*	*Effect-size*
RQ1	$H1_0$	Effectiveness	T	0.000139	1.255714
	$H2_0$	Efficiency	T	$7.381E-06$	1.983621
RQ2	$H3_0$	PEOU	T	$1.08E-05$	2.097433
	$H4_0$	PU	T	$7.11E-06$	2.485847
	$H5_0$	ITU	T	$9.28E-06$	2.185815

RQs	*Hyp.*	*Relation between variables*	*Reject*	*Eq.*	*p-value*
RQ3	$H6_0$	PEOU → PU	F	PU = 3.6 + 0.073 * PEOU	0.5962
	$H7_0$	PU → ITU	T	ITU = 0.5 + 0.86 * PU	2.44E-06
	$H8_0$	PEOU → ITU	F	ITU = 2.9 + 0.24 * PEOU	0.108
RQ4	$H9_0$	und. effv → PEOU	T	PEOU = 2.8 + 1.8 * und. effv	0.03677
	$H10_0$	und. effv → PU	F	PU = 3.2 + 0.97 * und. effv	0.08483
	$H11_0$	und. effc → PEOU	F	PEOU = 3.8 + 2.7 * und. effc	0.1752
	$H12_0$	und. effc → PU	F	PU = 4 - 0.5 * und. effc	0.8492
RQ5	$H13_0$	UNC effv → PEOU	F	PEOU = 4.5 - 0.74* UNC effv	0.7578
	$H14_0$	UNC effv → PU	F	PU = 4.5 - 0.44 * UNC effv	0.4241
	$H15_0$	UNC effc → PEOU	F	PEOU = 3.9 + 2.9 * UNC effc	0.2606
	$H16_0$	UNC effc → PU	F	PU = 3.9 - 0.22 * UNC effc	0.8952
	$H17_0$	UIC effv → PEOU	T	PEOU = 2.8 + 1.5 * UIC effv	0.02051
	$H18_0$	UIC effv → PU	F	PU = 3.5 + 0.43 * UIC effv	0.3332
	$H19_0$	UIC effc → PEOU	F	PEOU = 3. 9 + 1.1 * UIC effc	0.2168
	$H20_0$	UIC effc → PU	F	PU = 4 - 0.16 * UIC effc	0.7812
	$H21_0$	TRF effv → PEOU	F	PEOU = 3.7 + 0.73 * TRF effv	0.08802
	$H22_0$	TRF effv → PU	T	PU = 3.5 + 0.62 * TRF effv	0.02494
	$H23_0$	TRF effc → PEOU	F	PEOU = 3.9 + 10 * TRF effc	0.2105
	$H24_0$	TRF effc → PU	F	PU = 3.8 + 3.9 * TRF effc	0.4685

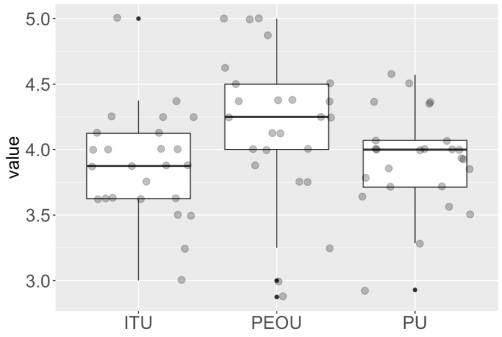
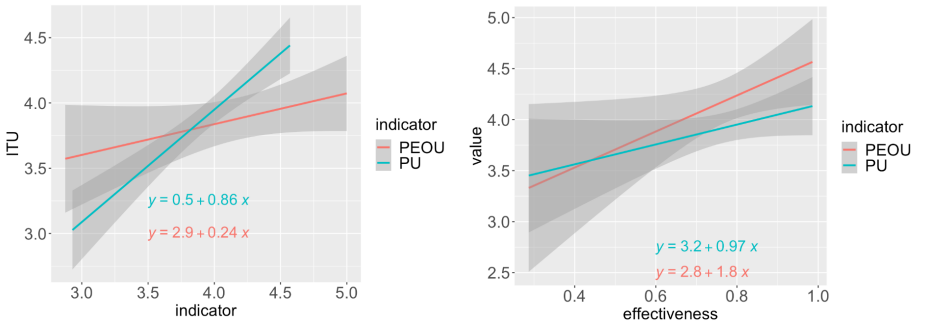


Fig. 4: Result for RQ2: boxplot of users' acceptance variables

and ITU (p-value = 2.44e-06). We can thus reject $H7_0$ and suggest that **users intend to use the notation in the future more for its usefulness than for its easiness**.

RQ4. To check if there is a relationship between the understandability of the notation and the users' perceptions about its easiness and usefulness, we test $H9_0$–$H12_0$ by fitting a linear model between PEOU and PU and understandability effectiveness, and between PEOU and PU and understandability efficiency. Our results show a significant positive relationship between effectiveness and perceived ease of use (cf. Fig. 5b), suggesting that **users who perform best in the test tend to evaluate better the notation in terms of easiness**.



(a) Relationship between PEOU and PU and ITU.

(b) Relationship between understandability effectiveness and PEOU and PU.

Fig. 5: Results for RQ3 and RQ4.

RQ5. Finally, to understand whether one of the understandability dimensions affects most the perceived easiness and usefulness, we fit a regression linear model between the perception-based variables (PEOU and PU), and the effectiveness and efficiency of the three understandability dimensions (understandability not in context, understandability in context, and transferability). Our results show that the effectiveness of understandability in context and of transferability both have a significant positive relationship with PEOU and PU, respectively (cf. Figs. 6a and 6b). We can thus reject $H17_0$ and $H22_0$, and confirm that users who observed instantiated trees and understand their meaning tend to evaluate the notation as easier, while users who apply the method better by extending the tree correctly tend to evaluate it as more useful. This suggests that **users who successfully use the notation in practice, tend to appreciate it more**.

5.1 Threats to Validity

Construct Validity. Users' acceptance was assessed through existing models [27] and adapted to the ADT notation according to [1]. The usage of effectiveness and efficiency for understandability performance is widely used in the literature (cf., e.g., [1,27,5]). For what concerns the understandability dimensions, retention

(a) Relationship between understandability in context effectiveness and PEOU and PU.

(b) Relationship between transferability effectiveness and PEOU and PU.

Fig. 6: Results for RQ5.

and transferability are adapted from [25,1], even if here we use retention as a means to retain the information gathered from the training phase rather than a dimension to be measured. Understandability not in context and in context are measures adapted from [1] to address the evaluation of syntax and semantics. The tasks used for each dimension have been revised by two ADT experts and considered appropriate to evaluate the understandability of the notation.

Internal Validity. To prevent systematic response bias in user acceptance questionnaires, we mixed positive and negative statements. Moreover, to minimise participant response bias and limit the possible tendency of users to provide positive answers to please the researchers, the experiment was conducted completely online; thus, none of the users met the experimenter. This approach not only preserved participant anonymity but also created a more naturalistic setting, minimising biases introduced by participants' awareness of being observed and diminishing the Hawthorne Effect. The support used during the test (e.g., the editable online document and diagram) may have influenced users' performance. To study this hypothesis, further investigation with users must be carried out to grasp their difficulties with the support.

External Validity. The selected participants encompass diverse genders and experience levels, enhancing generalisability. Participants were opportunistically chosen from the academic field, varying in seniority. However, their representation may not fully encompass all ADT user classes, influencing study results. Further research involving users from different fields is needed to confirm the applicability of conclusions across all user classes. It should also be noted that this study is a controlled experiment, which aims to maximise internal validity and does not evaluate ADT users in a realistic setting, where contextual factors play a relevant role. Therefore, case studies are needed to confirm that our conclusions apply in a real-life security analysis environment.

6 Conclusion and Future Work

In this paper, we presented the first empirical study to assess the quality of ADTs in terms of users' acceptance and understandability. Our evaluation measures how well the notation can be used in practice. In particular, our study focused on assessing users' perceptions variables that attest the notation appreciation in terms of ease of use, usefulness, and intention to use, and of performance variables that attest the degree of understandability of the notation in terms of effectiveness and efficiency. Understandability has also been studied according to three different fine-grained dimensions, and the relation between all these variables has been evaluated through multiple statistical tests.

Our results suggest that the ADT notation is sufficiently understood and greatly appreciated by users, specifically, the main aspect characterising its quality is its ease of use. Overall, the notation has a good level of understandability with a total average effectiveness above 0.76. Among its dimensions, we note better performance in more practical tasks (i.e., those related to observing and extending instantiated trees). Concerning relationships among the variables, we note that general understandability and understandability in context have a relationship with the perceived ease of use and that the ability to apply ADT in practice has a relationship with the perceived usefulness.

In future research, we plan to address user challenges in the test by conducting interviews to assess the impact of the platform on performance. To enhance result accuracy, we will broaden our subject pool, including users from diverse classes, such as those in the security field. We also intend to compare user performance and perceptions across ADTs and other security requirements modelling techniques, preferably textual methods. Additionally, our analysis will encompass various commercial and academic ADT tools.

Acknowledgements. Research supported by the Italian MUR–PRIN 2020TL3X8X project T-LADIES (Typeful Language Adaptation for Dynamic, Interacting and Evolving Systems); by Innovation Fund Denmark and the Digital Research Centre Denmark, through the bridge project "SIOT – Secure Internet of Things – Risk analysis in design and operation"; by Industriens Fond through the project "Sb3D: Security-by-Design in Digital Denmark"; and by the EU Project CODECS GA 101060179. The authors would like to thank all the participants of the study.

References

1. Abrahão, S., Insfrán, E., Carsí, J.A., Genero, M.: Evaluating requirements modeling methods based on user perceptions: A family of experiments. Inf. Sci. **181**(16), 3356–3378 (2011)
2. Audinot, M., Pinchinat, S., Kordy, B.: Is My Attack Tree Correct? In: ESORICS. LNCS, vol. 10492, pp. 83–102. Springer (2017)
3. ter Beek, M.H., Legay, A., Lluch Lafuente, A., Vandin, A.: Quantitative Security Risk Modeling and Analysis with RisQFLan. Comput. Secur. **109**, 102381 (2021)

4. Broccia, G., ter Beek, M.H., Lluch Lafuente, A., Spoletini, P., Ferrari, A.: Assessing the Understandability of Attack-Defense Trees for Modelling Security Requirements: an Experimental Investigation - Supplementary Material. https://doi.org/10.5281/zenodo.10136730
5. Broccia, G., Ferrari, A., ter Beek, M., Cazzola, W., Favalli, L., Bertolotti, F.: Evaluating a Language Workbench: from Working Memory Capacity to Comprehension to Acceptance. In: Proceedings 31st International Conference on Program Comprehension (ICPC). pp. 54–58. IEEE (2023)
6. Buyens, K., De Win, B., Joosen, W.: Empirical and statistical analysis of risk analysis-driven techniques for threat management. In: Proceedings 2nd International Conference on Availability, Reliability and Security (ARES). pp. 1034–1041. IEEE (2007)
7. Davis, F.D.: Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. MIS Q. pp. 319–340 (1989)
8. Eisentraut, J., Holzer, S., Klioba, K., Křetínskỳ, J., Pin, L., Wagner, A.: Assessing Security of Cryptocurrencies with Attack-Defense Trees: Proof of Concept and Future Directions. In: ICTAC. LNCS, vol. 12819, pp. 214–234. Springer (2021)
9. Ezenwoye, O., Liu, Y.: Risk-Based Security Requirements Model for Web Software. In: Proceedings 30th International Requirements Engineering Conference Workshops (REW). pp. 232–237. IEEE (2022)
10. Fabian, B., Gürses, S., Heisel, M., Santen, T., Schmidt, H.: A comparison of security requirements engineering methods. Requir. Eng. **15**, 7–40 (2010)
11. Gadyatskaya, O., Trujillo-Rasua, R.: New Directions in Attack Tree Research: Catching up with Industrial Needs. In: GraMSec. LNCS, vol. 10744, pp. 115–126. Springer (2017)
12. Giorgini, P., Mouratidis, H., Zannone, N.: Modelling Security and Trust with Secure Tropos. In: Integrating Security and Software Engineering: Advances and Future Visions, chap. 8, pp. 160–189. IGI Global (2007)
13. Iankoulova, I., Daneva, M.: Cloud Computing Security Requirements: a Systematic Review. In: Proceedings 6th International Conference on Research Challenges in Information Science (RCIS). pp. 1–7. IEEE (2012)
14. Kordy, B., Kordy, P., Mauw, S., Schweitzer, P.: ADTool: Security Analysis with Attack-Defense Trees. In: QEST. LNCS, vol. 8054, pp. 173–176. Springer (2013)
15. Kordy, B., Mauw, S., Radomirović, S., Schweitzer, P.: Foundations of Attack-Defense Trees. In: FAST. LNCS, vol. 6561, pp. 80–95. Springer (2010)
16. Kordy, B., Wideł, W.: On Quantitative Analysis of Attack-Defense Trees with Repeated Labels. In: POST. LNCS, vol. 10804, pp. 325–346. Springer (2018)
17. Labunets, K., Massacci, F., Paci, F.: On the Equivalence Between Graphical and Tabular Representations for Security Risk Assessment. In: REFSQ. LNCS, vol. 10153, pp. 191–208. Springer (2017)
18. Labunets, K., Massacci, F., Paci, F., Tran, L.M.S.: An Experimental Comparison of Two Risk-Based Security Methods. In: Proceedings 7th International Symposium on Empirical Software Engineering and Measurement (ESEM). pp. 163–172. IEEE (2013)
19. Lallie, H.S., Debattista, K., Bal, J.: An Empirical Evaluation of the Effectiveness of Attack Graphs and Fault Trees in Cyber-Attack Perception. IEEE Trans. Inf. Forensics Secur. **13**(5), 1110–1122 (2018)
20. Lallie, H.S., Debattista, K., Bal, J.: A review of attack graph and attack tree visual syntax in cyber security. Comput. Sci. Rev. **35**, 100219 (2020)
21. Liu, L., Yu, E.S.K., Mylopoulos, J.: Secure-I*: Engineering Secure Software Systems through Social Analysis. Int. J. Softw. Inform. **3**(1), 89–120 (2009)

22. Lodderstedt, T., Basin, D.A., Doser, J.: SecureUML: A UML-Based Modeling Language for Model-Driven Security. In: UML. LNCS, vol. 2460, pp. 426–441. Springer (2002)
23. Mai, P.X., Goknil, A., Shar, L.K., Pastore, F., Briand, L.C., Shaame, S.: Modeling Security and Privacy Requirements: a Use Case-Driven Approach. Inf. Softw. Technol. **100**, 165–182 (2018)
24. Mauw, S., Oostdijk, M.: Foundations of Attack Trees. In: ICISC. LNCS, vol. 3935, pp. 186–198. Springer (2005)
25. Mayer, R.E.: Models for Understanding. Rev. Educ. Res. **59**(1), 43–64 (1989)
26. Mellado, D., Blanco, C., Sanchez, L.E., Fernández-Medina, E.: A systematic review of security requirements engineering. Comput. Stand. Interfaces **32**(4), 153–165 (2010)
27. Moody, D.L.: Dealing with Complexity: A Practical Method for Representing Large Entity Relationship Models. Ph.D. thesis, University of Melbourne (2001)
28. Oliveira, D., Bruno, R., Madeiral, F., Castor, F.: Evaluating Code Readability and Legibility: An Examination of Human-centric Studies. In: Proceedings 36th International Conference on Software Maintenance and Evolution (ICSME). pp. 348–359. IEEE (2020)
29. Paja, E., Dalpiaz, F., Giorgini, P.: Modelling and reasoning about security requirements in socio-technical systems. Data Knowl. Eng. **98**, 123–143 (2015)
30. Salehie, M., Pasquale, L., Omoronyia, I., Ali, R., Nuseibeh, B.: Requirements-Driven Adaptive Security: Protecting Variable Assets at Runtime. In: Proceedings 20th International Requirements Engineering Conference (RE). pp. 111–120. IEEE (2012)
31. Schneier, B.: Attack Trees. Dr. Dobb's J. (1999)
32. Sharafi, Z., Marchetto, A., Susi, A., Antoniol, G., Guéhéneuc, Y.G.: An Empirical Study on the Efficiency of Graphical vs. Textual Representations in Requirements Comprehension. In: Proceedings 21st International Conference on Program Comprehension (ICPC). pp. 33–42. IEEE (2013)
33. Sindre, G., Opdahl, A.L.: Eliciting security requirements with misuse cases. Requir. Eng. **10**, 34–44 (2005)
34. Souag, A., Mazo, R., Salinesi, C., Comyn-Wattiau, I.: Reusable knowledge in security requirements engineering: a systematic mapping study. Requir. Eng. **21**, 251–283 (2016)
35. Stein, D., Hanenberg, S., Unland, R.: A Graphical Notation to Specify Model Queries for MDA Transformations on UML Models. In: MDAFA. LNCS, vol. 3599, pp. 77–92. Springer (2004)
36. Vesely, W.E., Goldberg, F.F., Roberts, N.H., Haasl, D.F.: Fault Tree Handbook. Tech. Rep. NUREG-0492, Nuclear Regulatory Commission, USA (1981)
37. Villamizar, H., Kalinowski, M., Viana, M., Fernández, D.M.: A Systematic Mapping Study on Security in Agile Requirements Engineering. In: Proceedings 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). pp. 454–461. IEEE (2018)
38. Wideł, W., Audinot, M., Fila, B., Pinchinat, S.: Beyond 2014: Formal Methods for Attack Tree-based Security Modeling. ACM Comput. Surv. **52**(4),75:1–75:36 (2019)
39. Zareen, S., Akram, A., Khan, S.A.: Security Requirements Engineering Framework with BPMN 2.0.2 Extension Model for Development of Information Systems. Appl. Sci. **10**(14), 4981 (2020)