Applying the Physics of Notation to the Evaluation of a Security and Privacy Requirements Engineering Methodology

Vasiliki Diamantopoulou and Haralambos Mouratidis

Centre for Secure, Usable and Intelligent Systems (CSIUS)
School of Computing, Engineering and Mathematics
University of Brighton, Brighton, UK
{v.diamantopoulou,h.mouratidis}@brighton.ac.uk

Abstract. Security and Privacy Requirements Engineering Methodologies are considered an important part of the development process of systems, especially for the ones that contain and process a large amount of critical information and inevitably need to remain secure and thus, ensuring privacy. These methodologies provide techniques, methods, and norms for tackling security and privacy issues in Information Systems. In this process, the utilisation of effective, clear and understandable modelling languages with sufficient notation is of utmost importance, since the produced models are used not only among IT experts or among security specialists, but also for communication among various stakeholders, in business environments or among novices in an academic environment. This paper evaluates the effectiveness of a Security and Privacy Requirements Engineering Methodology, namely Secure Tropos, on the nine principles of the Theory of Notation. Our qualitative analysis revealed a partial satisfaction of these principles.

Keywords: Security Requirements Engineering, Privacy Requirements Engineering, Physics of Notation, Evaluation

1 Introduction

The main objective of security and privacy requirements engineering methodologies is to provide techniques, methods and norms for dealing with each task, during the early stages of the Information Systems (IS) development cycle. Security and privacy requirements engineering methodologies supply researchers with existing information about security and privacy requirements in a thorough manner, providing the necessary context to operate [27]. Thus, it is imperative that security and privacy requirements should be specified at the early stages of an IS development process, since by conducting this analysis at an early stage, the building of such requirements is more efficient and also brings about more robust designs [23].

Visual notations, which are considered as a main element of each methodology, are used in all stages of the Software Engineering (SE) process [28], from

requirements engineering through to maintenance. They play a particularly critical role in communicating with end users and customers as they are believed to convey information more effectively to non-technical people than text [2], facilitating human communication and problem solving [16]. Visual representations are based on the exploitation of the capabilities of the human visual system. Diagrams can convey information more concisely [8] and precisely than ordinary language [4, 26]. Information presented visually is also more likely to be remembered due to the picture superiority effect [6, 11].

Despite the major contribution that visual syntax has on the understanding of each methodology, it has been argued that the researchers have ignored or undervalued its role. However, there are findings from various empirical studies which confirm the significant role of the visual form of notations and their positive affection to the comprehension of such methodologies, especially by novices [18, 20, 32, 35]. In this direction, it has been reported [12, 25] that more effort is spent on designing semantics of the methodologies, i.e. what concepts to include and what they mean, while visual syntax, i.e. how to visually represent these concepts, is often considered at a later stage. Notations are usually evaluated based mainly on their semantics, not paying the necessitated attention to visual syntax [33, 38]. Visual syntax should have been paid more attention to [1] since a successful representation of a system facilitates its solution.

Design rationale is the process of documenting design decisions made and the reasons they were made. This provides traceability in the design process and helps justify the final design [18]. Such rationale is conspicuously absent in design of methodologies visual notations. The graphical conventions that have been chosen are typically defined without any reference to respective theory or empirical evidence, or any other justification. However, the definition of explicit principles that transform [28] visual notation design from an *unselfconscious* process into a *self-conscious* process is imperative.

The diagram notation which is used during modelling has received little or no attention, and is regarded to be of secondary importance, probably a matter of taste rather than of science. This could be explained by the fact that researchers consider visual notations as being informal, and that therefore they analyse them only from the perspective of their semantics. However, this can be considered as a misunderstanding, since visual languages are no less formal than textual ones [4, 17]. Also, methods used for analysing visual representations are less mature than those for analysing verbal or mathematical representations [15, 40]. Finally, a third explanation could be that researchers and notation designers consider visual syntax to be insignificant, i.e. decisions about semantics (*content*) are paid high attention, while decisions about visual representation (*form*) are often considered to be a matter of aesthetics rather than effectiveness [18].

Taking all the above into consideration, we evaluate an already existing security and privacy requirements engineering methodology, namely Secure Tropos, regarding the visual notation that is being used. The aim of this study is to examine the graphical notation of Secure Tropos modelling language in order to further improve it at a later stage. To achieve that, we make use of the Physics

of Notation theory [28] since this work defines a set of principles for designing cognitively effective visual notations and moreover, it is considered as the most prominent and well accepted theory for the evaluation of software engineering methodologies. This work is an extended version of [9] which further analyses Secure Tropos in relation to the Physics of Notation theory.

The remainder of the paper is set out as follows: Section 2 discusses related work while Section 3 presents Secure Tropos security and privacy requirements engineering methodology, describing the main concepts of it, but mainly focusing on the visual notation that is being used. Section 4 provides the visual notation guidelines, as they have been defined by the relevant literature. Section 5 evaluates the aforementioned methodology, using the Physics of Notation principles. Section 6 presents issues that have been revealed after we conducted threats to validity for our work and finally, Section 7 concludes the paper, by raising issues for improvement of the examined methodology.

2 Related Work

In the IT field, one theory of visual notation design that the literature review revealed is the *Cognitive Dimensions Framework* [5, 13, 14]. This framework sets out a vocabulary of terms designed to capture the cognitively-relevant aspects of structure, and shows how they can be traded off against each other, being applied to visual programming environments. Nevertheless, this framework lacks to define theoretical and empirical foundations, it excludes visual representations from its analysis, it does not support evaluation of the chosen notations under evaluation.

Ontological analysis is also accepted for the evaluation of Software Engineering notations [10, 37]. This analysis is conducted through a two-way mapping between a modelling notation and an ontology. Ontological analysis supports the evaluation of the semantics of notations but specifically excludes visual representation aspects, since it focuses on content rather than on form.

The Physics of Notation [7, 28] defines a set of principles for designing cognitively effective visual notations, providing guidelines for efficient and conceivable representations of compex concepts. This study focuses on the physical properties of notations rather than their logical properties, forming thus a design theory, and it is considered as the most prominent and well accepted in the evaluation of software engineering.

3 Secure Tropos Methodology

Secure Tropos methodology [30] is based on the principle that security should be analysed and considered from the early stages of the software system development process, and not added as an afterthought. It is considered as a structured approach for goal-oriented security and privacy requirements, applicable to software systems, either to traditional ones or to cloud computing environments [31]. It is based on social hierarchies and adapts components of the i* framework [41].

The methodology provides a modelling language, a security-aware process, and a set of automated process to support the analysis and consideration of security from the early stages of the development process. This methodology is intended to support all the analysis and design activities in the software development process, supporting the fully capturing, analysis and reasoning of security and privacy requirements from the early stages of the development process. More specifically, it provides a modelling language that represents security and privacy requirements through constraints, allowing developers to model multi-agent software systems and their organisational environment. This language combines concepts i) from requirements engineering, for representing general concepts, such as actors, goals and actor dependencies, and ii) from security and privacy engineering, for representing security- and privacy-oriented concepts, such as security and privacy constraints, threats, vulnerabilities, plans, attacks security mechanisms and Privacy Enhancing Technologies (PETs).

The Secure Tropos methodology closely follows the software development life-cycle, i.e. capturing of early requirements, late requirements, architectural design, detailed design, and finally, implementation. Thus, it allows the developer to create and refine models, starting from the system-as-it-is, in order to finally develop the system-to-be, during the analysis and design stage.

3.1 Secure Tropos Model Views

The Secure Tropos modelling language is based on the concepts that have been defined in the requirements engineering discipline, combined with concepts from the security and privacy requirements engineering, all of whom are presented in Tables 1-4. The Secure Tropos produces models that contain security and privacy requirements analysis, but with the support of the corresponding tool, namely SecTro [34], the information is grouped according to three perspectives (views), i) the Organisational view, ii) the Requirements view and iii) the Attacks view. These interrelated modelling views are used in order to facilitate system design and security and privacy requirements elicitation. Each view provides specific focus of the system under analysis.

Organisational view This view represents the organisational architecture of the examined system, allowing a developer to understand the requirements of the organisation and any interactions between the organisation and external actors or systems. In addition, it displays the organisations' boundaries, where organisational actors reside; any external actors are modelled outside of this boundary. Moreover, in the Organisational view, the actors are defined along with their secure dependencies and any security and privacy constraints that might be imposed to these actors. Organisational view represents the system-as-it-is.

Requirements view This view provides a detailed representation of the Organisational view. There, system actors and their goals are designed including

the security and privacy analysis concepts. The modelling activity focuses on the responsibilities of the system and other actors, as well as the interaction of actors with the system itself. This view assist the developers to analyse the security and privacy issues of the system, by understanding the implications of security and privacy constraints, which have already been identified in the Organisational view. Additionally, this view allows the identification of threats, which are connected to specific goals, plans, or resources, that impact on. Requirements view represents the system-to-be.

Attacks View This view allows the evaluation of the system security and privacy against various attacks. The attack modelling takes place by analysing and checking whether security and privacy threats, which have already been introduced in the Requirements View, are mitigated by the security mechanisms and privacy enhancing technologies, respectively, available within the system. This view is unique for each identified threat in the Requirements view. Here, each threat is analysed to identify its potential attack methods, the system vulnerabilities they exploit and the protection provided by the proposed security mechanisms against such vulnerabilities. If the developer identifies any inability of the system to mitigate these threats, they follow an iterative process, going back to the Requirements View, and adjust the design accordingly.

3.2 Secure Tropos Process

Using the different modelling views supported by SecTro tool, security- and privacy-related features of the system can be analysed from a variety of perspectives. This subsection focuses on the process from which the models are constructed. The process is not strictly sequential, it is rather a iterative process, as the developer can return to a previous view to enhance or alter their model. The diagrams produced in one modelling activity are used as input for the other activities.

Step 1: Organisational modelling During this step, the security engineer, alongside the relevant stakeholders of the system's environment, identify:

- The actors of the system
- The goals (hard and soft) that these actors have
- The plans and the resources that are required for the realisation of the goals
- The dependencies that one actor might have on another actor, for the achievement/realisation of a goal, a plan, or a resource
- The security and privacy requirements of the examined system, which are presented in the form of security and privacy constraints, respectively, that might restrict the actors.

This diagram can also present any relationships between the examined system with external ones.

Step 2: Security and Privacy Requirements Modelling Through this step, a more deep representation of the security and privacy aspects of the system

is provided. This activity will produce a more refined version of the previous diagram, in terms of security and privacy constraints, and threat analysis. More specifically, this step contains:

- Description of the relationship between attacks expected and mitigation mechanisms for any identified threat
- Introduction of a number of resources, which represent various assets that are either created from or required for the achievement of each of the modelled goals
- Introduction of plans that indicate activities required for the achievement of certain system goals
- Modelling of threats of the systems that impact different goals and resources
- Introduction of security and privacy mechanisms that protect the system against each of the identified vulnerabilities

Step 3: Security Attacks Modelling This step allows the refinement of threats, by modelling attackers and ways to mitigate attacks on vulnerabilities. Here, the security engineer demonstrates how each threat can impact the system.

- Identification of the attack methods that a threat can utilise
- Identification of the vulnerabilities that the above attack methods can exploit
- Identification of the system resources and goals that the above vulnerabilities can affect
- Refinement of the security and privacy mechanisms, should this analysis phase reveals any vulnerabilities of the system that the mechanisms of step 2 cannot protect

4 Visual Notation Principles according to the Theory of Notation

For the effective approach of the evaluation of the graphics of notation, the reader should be aware of specific definitions. A **visual notation** (or visual language, graphical notation, diagramming notation) consists of a set of **graphical symbols** (**visual vocabulary**), a set of **compositional rules** (**visual grammar**) and definitions of the meanings of each symbol (**visual semantics**). The visual vocabulary and visual grammar together form the **visual** (or **concrete**) **syntax**. Graphical symbols are used to **symbolise** (perceptually represent) **semantic content**, typically defined by a **metamodel**. The meanings of graphical symbols are defined by mapping them to constructs they represent [28]. A valid expression in a visual notation is called a **visual sentence** or **diagram**. Diagrams are composed of **symbol instances**, arranged according to the rules of the visual grammar. What has to be addressed in visual notation design is the clear **design goal**. Goals such as simplicity, aesthetics, expressiveness, and naturalness are often mentioned in the literature. In addition, to be most effective in facilitating human communication and problem solving, visual notations need to be optimised for processing by the human mind. Thus, **cognitive effectiveness**

is defined as the speed, ease, and accuracy with which a representation can be processed by the human mind [26]. Cognitive effectiveness determines the ability of visual notations to both communicate with business stakeholders and support design and problem solving by software engineers.

According to [28], there are nine principles for designing cognitively effective visual notation. For the development of these principles, information from theory and empirical evidence about cognitive effectiveness of visual representations has been synthesised. More specifically, the nine principles that will be the guide for the evaluation of security requirements methodology are the following:

1. **Principle of Semiotic Clarity**: This principle mentions that there should be an one-to-one correspondence between semantic constructs and graphical symbols. The notations aim at precision, expressiveness, and parsimony, in order for users to effectively design the examined systems.
2. **Principle of Perceptual Discriminability**: This principle mentions that different symbols should be clearly distinguishable from each other. The concepts should have been represented with accurate graphical symbols, easily distinguishable. Consequently, this can lead to the accurate interpretation of the model as a whole [40]. This principle is determined by i) the *visual distance* between the symbols, i.e. the different visual variables that have been used for the representation of each concept, ii) the *primacy of shapes*, which contributes to the identification of the objects within a diagram, iii) the *redundant coding* which contributes to the elimination of errors, iv) the *perceptual popout* which suggests a unique value on at least one visual variable, and v) the *textual differentiation*, when the discrimination among the concepts is basically achieved with the use of text and typographic characteristics (fond styles such as bold, italics and underlining).
3. **Principle of Semantic Transparency**: This principle highlights the utilisation of visual representations whose appearance suggests their meaning. The notation that is used should be such, that the user can comprehend the content of the symbol only by its appearance, by providing cues to their meaning. This principle aims to minimise the demanded effort for the understanding of the meaning of a concept.
4. **Principle of Complexity Management**: This principle focuses on diagrams' notation, mentioning that explicit mechanisms for dealing with complexity should be included. The complexity level of a diagram plays an important role in its comprehension, especially when dealing with novices. Excessive complexity is considered a barrier for users to understand SE diagrams [29, 36]. Modularisation and hierarchy are mechanisms that can be used in order to manage complexity in SE notations. More specifically, modularisating SE diagrams could result to the improvement of end-users' comprehension. This can be achieved through certain semantic constructs, i.e. subsystem constructs or decomposable constructs. Also, diagrammatic conventions for the decomposition of diagrams should be defined. Regarding hierarchy, it allows systems to be represented at different levels of abstrac-

tion and detail, allowing thus, developers to control the complexion at each level.

5. **Principle of Cognitive Integration**: This principle mentions the inclusion of explicit mechanisms to support integration of information from different diagrams. The representation of a system through multiple diagrams demands additional effort by the end user to integrate information from different sources (diagrams). This state has been addressed through i) conceptual and ii) perceptual integration. Conceptual integration refers to mechanisms that support the assembling of information from different diagrams into contiguous system representation. Perceptual integration aims to provide the navigation and transition from the one diagram to the other in a simpler and easy for the reader to follow way.
6. **Principle of Visual Expressiveness**: This principle suggests the full range and capacities of visual variables. More specifically, this principle measures visual variation across the entire visual vocabulary [4]. The expression of each concept with the use of a range of visual variables results in the enrichment of the representation that exploits multiple visual communication channels. This principle, which is also related with the one of Perceptual Discriminability, can contribute to the improvement of models understandability. The choice of visual variables should be based on the nature of information that needs to be conveyed [4].
7. **Principle of Dual Coding**: In continuation to both the visual expressiveness and complexity management, this principle suggest the use of text in the modelling process, when the text is used supplementary, rather than as a substitute, i.e. as a form of *redundant coding to reinforce and clarify meaning*. This principle is also based in the differentiated characteristics that humans have regarding their ability to comprehend a meaning. The use of dual coding aims at capturing the human abilities across their full spectrum of spatial and verbal abilities [39].
8. **Principle of Graphic Economy**: This principle refers to the careful number of different graphical symbols that should be used in a methodology. It is argued [24] that the cognitive limits on the number of visual categories that the human mind can effectively recognise are limited. Consequently, the reasonable use of visual categories is proposed, otherwise the users' understandability is negatively affected.
9. **Principle of Cognitive Fit**: This principle highlights the use of different visual dialects for the representation of information, either in case that we deal with different audiences, or in case that we have different representational medium. In the first case, the representation should cover both the expert users and the novices, since they have different level of understandability. In this direction, the approach of the 'lowest common denominator', by using notations understandable by both two types of audience should be avoided, since it can negatively affect the effectiveness for both of the types of users [22]. Regarding the representational medium, this also can affect the communication of the model with the user. More specifically, since there is the option of the representation of a model without the assistance of a

CASE tool, the representation of the concepts should be such, to be able to be transferred in 'a piece of paper'. This aspect of the principle of cognitive fit can explain the absence of techniques such as colour, icons, and 3D shapes.

5 Methodology Evaluation

The Secure Tropos graphical notation has not followed specific justification regarding the design choices of the symbols that are used. These design choices have been asserted, following unself-conscious design culture [1], which it is not based on explicit design principles but on instinct, imitation, and tradition. Nevertheless, we proceed with the evaluation of its graphical notation based on the nine principles of the Theory of Notation [28].

Principle of Semiotic Clarity The concepts and the relationship elements of Secure Tropos, which are presented in Tables 1-4, reveal that there is one-to-one correspondence between symbols and their referent concepts. This correspondence contributes to the precision and the efficient expressiveness of the symbols, avoiding the ambiguity and their misinterpretation by the users. Thus, the principle of Semiotic Clarity is fully satisfied.

Principle of Perceptual Discriminability Regarding the shapes that have been used in order to represent the various concepts in Secure Tropos, the visual distance between the symbols is substantial enough. The identification of the various objects is achieved through the utilisation of the most of the concepts (see Tables 1 and 2) different shapes and colours. The shapes that have been used for the representation of the communication links (see Tables 3 and 4) consist of lines, but with elements that discriminate them (i.e. arrows, dashed lines). It is argued [28] that most SE notations use a perceptually limited repertoire of shapes, mostly rectangle variants. In the examined methodology we can identify the use of clearly discriminable shapes that represent different constructs; they all come from different shape families and differences between them can be detected pre-attentively. Furthermore, the variable of colour is also used in the concepts of the methodology, improving discriminability between entities, satisfying the *redundant coding* sub-principle. However, the same colour for more than one concepts is being used and this can cause misunderstandings that might incommode the perceptual processing of the user. In addition, Secure Tropos uses text (labels) to differentiate between most of the relationship types. Textual differentiation of symbols is a common but cognitively ineffective way of dealing with excessive graphic complexity, as text processing relies on less efficient cognitive process. Textual differentiation of symbols also confounds the role of text in diagrams. Labels play a critical role at the sentence level in distinguishing between symbol instances and defining their correspondence in the real world. Also, when labels are used to distinguish between relationship types, it precludes the use of user-defined and domain-relevant names. Text is an effective way to distinguish between symbol instances but not between symbol types. Thus, the principle of Perceptual Discriminability is partially satisfied.

Table 1: Concept Types on Secure Tropos methodology - Organisational and Requirements View

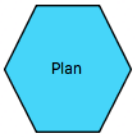
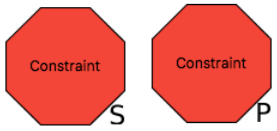
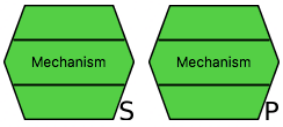
Concept	Description	Notation
Actor	Active entities that carry out actions to achieve goals by exercising its know-how. It refers generically to any unit to which intentional dependencies can be ascribed. Actors depend on each other to achieve goals, perform tasks, and furnish resources. While each actor has strategic goals to pursue, they are achieved through a network of intentional dependencies	
Hard Goal	A condition or state of affairs to be achieved. An actor can choose freely among different ways to achieve a goal. Thus, a goal represents an intentional desire of an actor. The specifics of *how* the goal is to be satisfied is not described by the goal but through task decomposition.	
Soft Goal	A goal that does not have clear-cut definition or criteria on whether it has been achieved. It represents quality attributes for which there are no a priori, clear criteria for satisfaction, but actors have to fulfil. Soft goals are typically used to model non-functional requirements.	
Plan	Represents a way of doing something. The fulfilment of a plan can be a means of satisfying a goal. As such, different alternative plans that actors might employ to achieve their goals, are modelled to enable software engineers to reason about the different ways that actors can achieve their goals, and decide upon the optimal way.	
Resource	Represents a physical or informational entity that an actor requires. The main concern is whether the resource is available and who is responsible for its delivery.	
Constraint	A restriction on an actor's function. There are two types of Constraints, namely Security and Privacy. Additionally, a Constraint is related to an objective that needs to be fulfilled, which is expressed through the constraint, such as Confidentiality, Integrity, Authentication, etc.	
Mechanism	Represents a system mechanism that supports the satisfaction of a security or privacy constraint. It can be any of two types, Security or Privacy.	
Threat	Represents a circumstance that has the potential to cause damage to the system.	

Table 2: Concept Types on Secure Tropos methodology - Security Attacks View

Concept	**Description**	**Notation**
Attacker	A malicious actor who tries to endanger the security of the system through attacking its resources, goals and plans.	
Vulnerability	A weakness of the system or the organisation.	
Attacks method	A method by which a Threat is realised.	

Principle of Semantic Transparency Among all the graphical notations of Secure Tropos, there is one, the security or privacy "Constraint" which is depicted as a "Stop" sign and satisfies this principle. Stop sign is a familiar signal which can be interpreted as the criticality of a situation. In the same way, the concept of constraints represents a set of restrictions that do not permit specific actions to be taken (see Table 5). Attack link also satisfies this principle, since its depiction is accompanied by two symbols, i.e. a red exclamation mark and a green tick. The first symbol aims to gain user's attention since an identified vulnerability has not been mitigated by a security or a privacy mechanism, while the second symbol confirms that all possible attacks have been mitigated. Thus, the principle of Semantic Transparency is partially satisfied.

Principle of Complexity Management A common problem that is encountered in goal oriented diagrams is about their complexity when the diagrams are too overloaded with information. This problem is even greater in Secure Tropos, since the models capture not only system's requirements information, but also contain security and privacy requirements. However, since this issue has already been identified, in the diagrams of Secure Tropos the design can follow *hierarchy* structure for the representation of goals, in order for the model to be well-structured, and thus contributing to the readability of each model. Moreover, the concept of *modularisation* finds application in Secure Tropos, since, as we described in Section 3, there are different views, i.e. Organisational view, which represents the organisational architecture of the system, Requirements view, where a deeper representation of the Organisational view is presented, al-

Table 3: Relationship Types on Secure Tropos methodology - Organisational and Security Requirements View

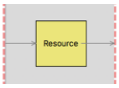
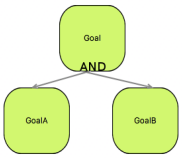
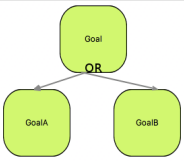
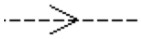
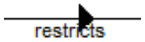
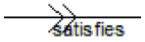
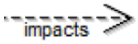
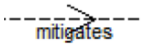
Relation class	Description	Notation
Dependency	The depender depends on the dependee to bring about a certain state of affairs in the world. The dependum is expressed as an assertion statement. The dependee is free to and is expected to make whatever decisions are necessary to achieve the goal (namely, the dependum). The depender does not care how the dependee goes about achieving the goal.	
And	Allows the decomposition of an element to more fine grained elements. All the sub-elements need to be fulfilled in order the parent element to be fulfilled as well. The elements that can be decomposed are a goal, a plan, a resource, a mechanism, an attack method.	
Or	Allows the decomposition of an element to more fine grained elements. The difference with the 'And' relationship is that only one element is needed for the fulfilment of the parent element.	
Contribution	Shows a contribution toward satisfying a soft goal, typically from a task or another soft goal. Any of these Contribution links can be used to link any of the elements to a soft goal to model the way any of these Elements contributes to the satisfaction or fulfilment of the soft goal.	
Restricts	Shows the goal that is restricted by a Security Constraint.	
Satisfies	Shows the security or privacy Constraint that a mechanism satisfies.	
Impacts	Shows the Goal that is affected by a Threat	
Mitigates	Shows the Threat that is mitigated by a Security Mechanism.	

Table 4: Relationship Types on Secure Tropos methodology - Security Attacks View

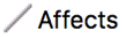
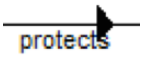
Relation class	**Description**	**Notation**
Attacks	Shows the Vulnerability that an Attack Method is exploiting.	
Affects	Shows what goals and/or resources a vulnerability puts at risk.	
Protects	Shows what mechanisms work as countermeasure.	

Table 5: Partial satisfaction of "Semantic Transparency" principle

	
	The red exclamation mark indicates that the identified vulnerability has not been mitigated by a security/privacy mechanism
	The green tick confirms that all possible attacks have been mitigated

lowing the further analysis of the security and privacy issues of the examined system, and Attacks view, which further analyses each threat to identify its attack methods, the system vulnerabilities they exploit and the protection provided by the proposed security and privacy mechanisms against such vulnerabilities. Consequently, the information is grouped according to these three perspectives, contributing to the presentation of the information in a more grouped way, facilitating thus the designers to focus each time on a different perspective of the examined system. Through this approach, each model is presented to the user from different viewpoints, improving their understanding. In addition, each identified threat is presented in an additional view, so as the created models are more readable. Thus, the principle of Complexity Management is partially satisfied.

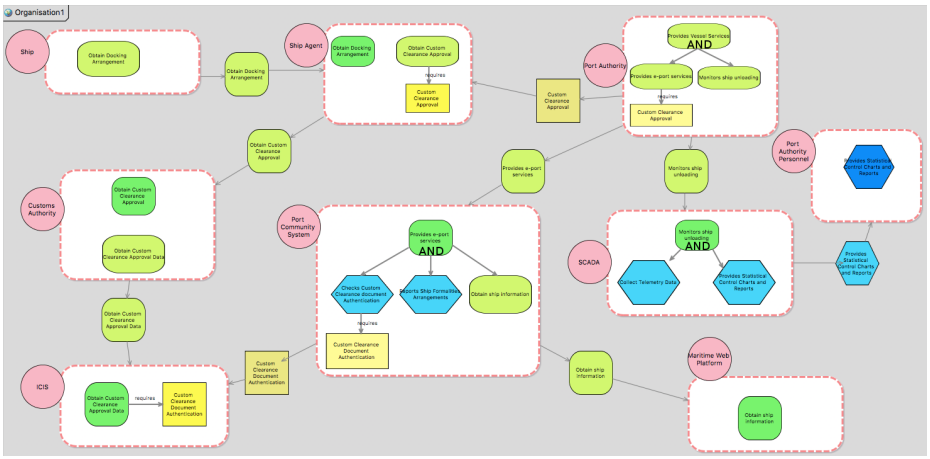


Fig. 1: Organisational View

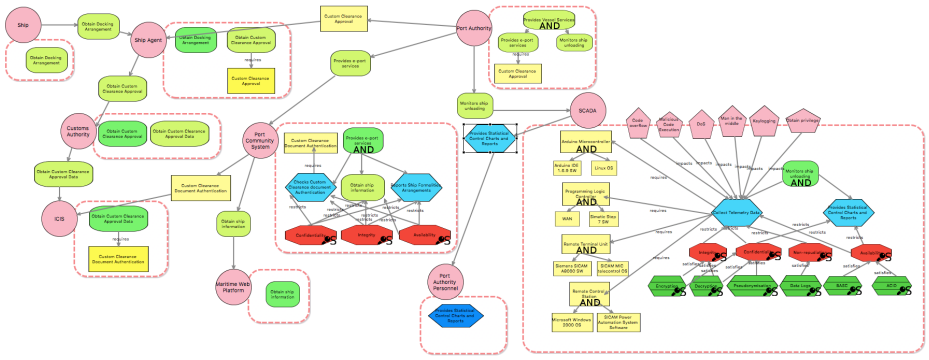


Fig. 2: Requirements View

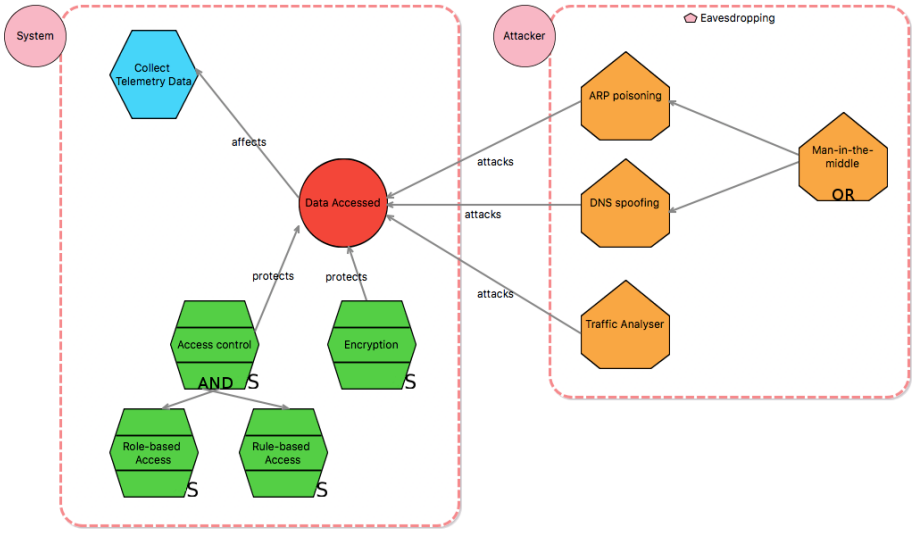


Fig. 3: Attacks View

Principle of Cognitive Integration As we described in the previous principle, in Secure Tropos multiple diagrams are used to represent one system. Each view is responsible for specific analysis of the system-as-it-is and also the system-to-be. Consequently, an end-user needs to parse all the information that has been recorded in each view, in order to have a holistic knowledge of the examined system. Despite this complexity, the notation that the methodology uses is presented in this way that contributes to the elimination of the effort that is demanded by the reader in order to keep track of where they are. The transition from one view to the other can be achieved more smoothly and can constitute to the connection point between different views. As it has been highlighted in Fig 4, separated tabs support user orientation by indicating where they are in the system of diagrams, allowing easy navigation. Moreover, the concepts that are introduced in the Organisational view (the first view) and are essential for the further analysis to the next two views, are automatically introduced. This results to the facilitation of the user to realise the core concepts of the analysed system. Thus, the principle of Cognitive Integration is fully satisfied.

Principle of Visual Expressiveness Secure Tropos uses colours in order to distinguish each concept. Colour is not the only identifiable characteristic of each concept, shape is another one. They together facilitate comprehension of the models, avoiding misunderstandings, technical or human related (e.g., black-and-white printing, colour blindness, respectively). In addition, Secure Tropos uses a variety of shapes, i.e. rectangle, rounded rectangle, cycle, hexagon, heptagon, octagon, diamond shape, and ellipse. The literature refers that this variety of shapes is the less effective one regarding human visual processing, and thus curved, 3D, and iconic shapes have to be preferred [3, 40]. Regarding the ratio

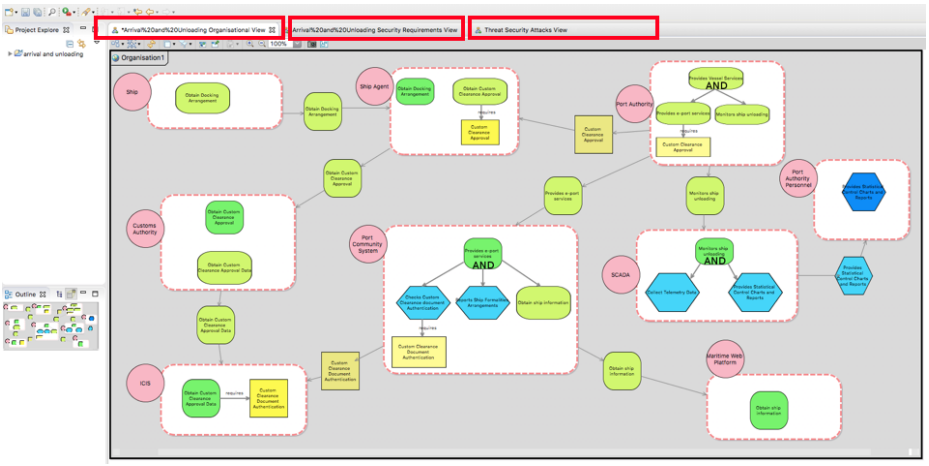


Fig. 4: Satisfaction of "Cognitive Integration" principle

of graphical encoding versus textual encoding, Secure Tropos fails to satisfy this balance (it is argued that the more visual variables that are used, the greater the role of perceptual processing [28]), since textual encoding is used in all of the relationship notations; a point that is not preferred if a model aims to maximise their visual expressiveness. Thus, the principle of Visual Expressiveness is partially satisfied.

Principle of Dual Coding The SecTro tool supports the depiction of each concept of Secure Tropos both by a graphic symbol and their corresponding label. The labels are used only in the pallet of the tool (see Fig. 5), contributing to the learnability [19] and memorability [21] usability criteria. Moreover, when a concept is inserted to the design space, a Properties panel (see Fig. 6) provides information regarding the specific concept, which can also contribute to the satisfaction of the Dual Coding. In this way, the interpretation of each concept can be achieved with confidence by the user. Thus, the principle of Dual Coding is fully satisfied.

Principle of Graphic Economy By using the different views (Organisational view, Requirements view, Attacks view) of the Secure Tropos tool, the user is able to focus on a specific perspective of the examined system. The graphic economy is achieved and thus the diagrams are effectively presented to the users. With the use of different views, Secure Tropos does not concentrate vast amounts of information in the same model, but distinguishes information according to the focus of each part of the analysis. For example, the Organisational view focuses on the elements (actors) that compose the architecture of the examined organisation, the Requirements view focuses on security and privacy constraints, and finally, the attacks view analyses separately each identified threat, in relation to the affected concepts (goals, plans or resources of an actor). Thus, the principle of Graphic Economy is fully satisfied.

Fig. 5: Pallete of Concepts of SecTro tool of Requirements View

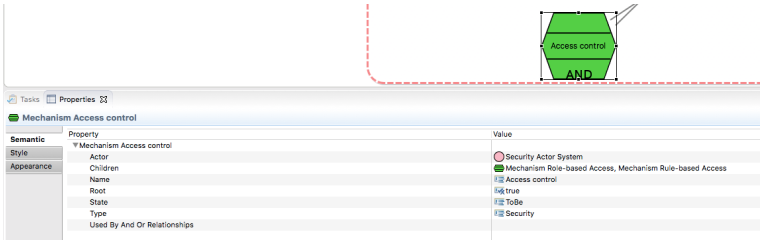


Fig. 6: Properties panel of SecTro tool

Principle of Cognitive Fit Secure Tropos modelling language is not provided in two versions, as it is suggested by this principle. There is the requirement that the language should be provided in different versions, covering mainly the level of expertise of users, as due to its wide applicability, it is used by students, IT security experts, project managers, and also simple users. Thus, the principle of Cognitive Fit is not satisfied. However, Secure Tropos methodology adopts concepts from well-known requirements engineering languages and therefore, it is easier for the users who are not security experts, but have some understanding of Requirements Engineering, such as the concepts of actors, goals, and dependencies, to also understand Secure Tropos concepts.

Moreover, Secure Tropos adopts well-known Security Engineering concepts, such as the concepts of threats, vulnerabilities, and attacks. In this way, it helps a user who has general knowledge of Security Engineering to comprehend Secure Tropos methodology. Finally, since Secure Tropos is used in academic environments, there have been developed tutorials that support novices to capture the main idea and the concepts of the methodology, as well as there are detailed tutorials that support the usage of SecTro tool.

From the above analysis, it is revealed that Secure Tropos modelling language fully satisfies four out of nine principles, four of them are partially satisfied, while one principle is not satisfied at all. For the one principle that is not satisfied at all, there area already some foundations that can contribute towards its satisfaction.

These results can be used in order to better improve the language, focusing in the revision of specific elements which can contribute to the overall communication of the language with its users.

6 Threats to validity

The evaluation of Secure Tropos methodology presented in this paper is subject to threats to validity, since in this way we can denote the trustworthiness of the results of our analysis. Moreover, we will be able to identify to what extent these results are true, and not biased by the researchers' personal and subjective perspective.

Internal validity: On this stage, focus is given on the team that evaluated this methodology. Some of the members of this team were quite familiar with the methodology - one of them was one of the founders of the original methodology. On the other hand, there was also a member of the team, whose involvement in the methodology was rather low, that evaluated the methodology from a more critical and objective perspective. This synthesis offered the necessary balance within the team, which was able to bring objective and useful results.

Construct validity: This aspect examines to what extent the operational measures that are studied represent actually what the researchers had in their minds, and what is investigated. What we have identified in our evaluation is that it suffers from the so-called mono-method bias, i.e. the subjects were treated only with one methodology. For this aspect, future work includes the evaluation of other security and/or privacy requirements engineering methodologies, where the results will allow us to compare Secure Tropos methodology to others.

External validity: This work has been conducted in a theoretical level, since, due to time limitations, we couldn't have feedback from external users. However, this methodology has been recently used in an EU H2020 project[1], where pilots from various domains were run, i.e. the Public Administration domain and the Healthcare domain, and they applied it. Despite the fact that the evaluation that had been conducted didn't specifically focus specifically on the Secure Tropos, we didn't receive any negative comments from users who were considered novices in using this methodology.

7 Conclusions

The effectiveness of a methodology to efficiently communicate its content with the users is of equal importance to the semantics of it. In this paper we evaluate a security and privacy requirements engineering methodology, namely Secure Tropos, based on the most well-known theory, the Physics of Notation, which has been synthesised from theory and empirical comparison and can be used for the evaluation, comparison and improvement of visual notations. Our qualitative analysis resulted in valuable lessons learned, which are thoroughly discussed in

[1] `http://www.visioneuproject.eu/`

Section 4, and can also be applied to other security and privacy requirements engineering methodologies. This application, which is one of our future works, will allow us i) to evaluate them and proceed to comparisons among them, and ii) to develop guidelines for the improvement of their visual syntax.

Moreover, empirical analysis is also another future step, in order to identify to what extend the proposed outcomes of the analysis of this paper can improve the communication between the analysts and end users. The users have to be distinguished between experts and novices and the aim is to record their perception regarding the **design goals**, such as simplicity, aesthetics, expressiveness and naturalness, and also, regarding **cognitive effectiveness**, such as speed, ease, and accuracy.

Finally, in order to further strengthen the validity of our results, external practitioners will be involved in the study. This could substantially raise the subjectiveness of the evaluation part of this research.

References

1. Alexander, C.: Notes on the Synthesis of Form, vol. 5. Harvard University Press (1964)
2. Avison, D., Fitzgerald, G.: Information systems development: methodologies, techniques and tools. McGraw Hill (2003)
3. Bar, M., Neta, M.: Humans prefer curved visual objects. Psychological science 17(8), 645–648 (2006)
4. Bertin, J.: Semiology of graphics: diagrams, networks, maps (1983)
5. Blackwell, A., Green, T.: Cognitive dimensions of notations resource site. URL http://www. cl. cam. ac. uk/afb21/CognitiveDimensions (2009)
6. Butler, J., Holden, K., Lidwell, W.: Universal principles of design: A cross-disciplinary reference (2003)
7. Caire, P., Genon, N., Heymans, P., Moody, D.L.: Visual notation design 2.0: Towards user comprehensible requirements engineering notations. In: Requirements Engineering Conference (RE), 2013 21st IEEE International. pp. 115–124. IEEE (2013)
8. DeMarco, T.: Structured analysis and system specification. Yourdon Press (1979)
9. Diamantopoulou, V., Pavlidis, M., Mouratidis, H.: Evaluation of a security and privacy requirements methodology using the physics of notation (2017)
10. Gehlert, A., Esswein, W.: Toward a formal research framework for ontological analyses. Advanced Engineering Informatics 21(2), 119–131 (2007)
11. Goolkasian, P.: Pictures, words, and sounds: From which format are we best able to reason? The Journal of General Psychology 127(4), 439–459 (2000)
12. Grady, B.: Object-oriented analysis and design with applications (1994)
13. Green, T.R.G., Petre, M.: Usability analysis of visual programming environments: a ?cognitive dimensions? framework. Journal of Visual Languages & Computing 7(2), 131–174 (1996)
14. Green, T.R.: Cognitive dimensions of notations. People and computers V pp. 443–460 (1989)
15. Gurr, C.A.: Effective diagrammatic communication: Syntactic, semantic and pragmatic issues. Journal of Visual Languages & Computing 10(4), 317–342 (1999)

16. Harel, D.: On visual formalisms. Communications of the ACM 31(5), 514–530 (1988)
17. Harel, D., Rumpe, B.: Meaningful modeling: what's the semantics of" semantics"? Computer 37(10), 64–72 (2004)
18. Hitchman, S.: The details of conceptual modelling notations are important-a comparison of relationship normative language. Communications of the Association for Information Systems 9(1), 10 (2002)
19. Holzinger, A.: Usability engineering methods for software developers. Communications of the ACM 48(1), 71–74 (2005)
20. Irani, P., Ware, C.: Diagramming information structures using 3d perceptual primitives. ACM Transactions on Computer-Human Interaction (TOCHI) 10(1), 1–19 (2003)
21. Jackson, M., Crouch, S., Baxter, R.: Software evaluation: criteria-based assessment. Software Sustainability Institute (2011)
22. Kalyuga, S., Ayres, P., Chandler, P., Sweller, J.: The expertise reversal effect. Educational psychologist 38(1), 23–31 (2003)
23. Kim, J., Kim, M., Park, S.: Goal and scenario based domain requirements analysis environment. Journal of Systems and Software 79(7), 926–938 (2006)
24. von Klopp Lemon, A., von Klopp Lemon, O.: Constraint matching for diagram design: Qualitative visual languages. Theory and Application of Diagrams pp. 589–603 (2000)
25. Lankhorst, M.: Enterprise architecture at work: Modelling, communication and analysis (the enterprise engineering series) (2009)
26. Larkin, J.H., Simon, H.A.: Why a diagram is (sometimes) worth ten thousand words. Cognitive science 11(1), 65–100 (1987)
27. Mellado, D., Blanco, C., Sánchez, L.E., Fernández-Medina, E.: A systematic review of security requirements engineering. Computer Standards & Interfaces 32(4), 153–165 (2010)
28. Moody, D.: The "physics" of notations: toward a scientific basis for constructing visual notations in software engineering. IEEE Transactions on Software Engineering 35(6), 756–779 (2009)
29. Moody, D.L.: Complexity effects on end user understanding of data models: An experimental comparison of large data model representation methods. ECIS 2002 Proceedings p. 10 (2002)
30. Mouratidis, H.: A natural extension of tropos methodology for modelling security (2002)
31. Mouratidis, H., Argyropoulos, N., Shei, S.: Security requirements engineering for cloud computing: The secure tropos approach. In: Domain-Specific Conceptual Modeling, pp. 357–380. Springer International Publishing (2016)
32. Nordbotten, J.C., Crosby, M.E.: The effect of graphic style on data model interpretation. Information Systems Journal 9(2), 139–155 (1999)
33. Opdahl, A.L., Henderson-Sellers, B.: Ontological evaluation of the uml using the bunge–wand–weber model. Software and systems modeling 1(1), 43–67 (2002)
34. Pavlidis, M., Islam, S.: Sectro: A case tool for modelling security in requirements engineering using secure tropos. In: CAiSE Forum. pp. 89–96 (2011)
35. Purchase, H.C., Carrington, D., Allder, J.A.: Empirical evaluation of aesthetics-based graph layout. Empirical Software Engineering 7(3), 233–255 (2002)
36. Shanks, G., Darke, P.: Understanding corporate data models. Information & Management 35(1), 19–30 (1999)
37. Shanks, G., Tansley, E., Weber, R.: Using ontology to validate conceptual models. Communications of the ACM 46(10), 85–89 (2003)

38. Siau, K., Cao, Q.: Unified modeling language: A complexity analysis. Journal of Database Management (JDM) 12(1), 26–34 (2001)
39. Wiegmann, D.A., Dansereau, D.F., McCagg, E.C., Rewey, K.L., Pitre, U.: Effects of knowledge map characteristics on information processing. Contemporary educational psychology 17(2), 136–155 (1992)
40. Winn, W.: Encoding and retrieval of information in maps and diagrams. IEEE Transactions on Professional Communication 33(3), 103–107 (1990)
41. Yu, E., Liu, L., Mylopoulos, J.: A social ontology for integrating security and software engineering. Integrating Security and Software Engineering: Advances and Future Actions pp. 70–105 (2006)