Extending the UML Statecharts Notation to Model Security Aspects

Mohamed El-Attar, Hamza Luqman, Péter Kárpáti, Guttorm Sindre, *Member, IEEE*, and
Andreas L. Opdahl, *Member, IEEE*

Abstract—Model driven security has become an active area of research during the past decade. While many research works have contributed significantly to this objective by extending popular modeling notations to model security aspects, there has been little modeling support for state-based views of security issues. This paper undertakes a scientific approach to propose a new notational set that extends the UML (Unified Modeling Language) statecharts notation. An online industrial survey was conducted to measure the perceptions of the new notation with respect to its semantic transparency as well as its coverage of modeling state based security aspects. The survey results indicate that the new notation encompasses the set of semantics required in a state based security modeling language and was largely intuitive to use and understand provided very little training. A subject-based empirical evaluation using software engineering professionals was also conducted to evaluate the cognitive effectiveness of the proposed notation. The main finding was that the new notation is cognitively more effective than the original notational set of UML statecharts as it allowed the subjects to read models created using the new notation much quicker.

Index Terms—Statecharts, security modeling, extended notation, industrial survey, subject-based experiment

✦

1 INTRODUCTION

SECURITY is nowadays an indispensable quality of IT-systems. Traditional software development focuses on developing business-related functionality while leaving security as an afterthought. Security is addressed at the final stages of development by *patching* a system with generic defensive mechanisms, such as intrusion detection systems, cryptographic components and firewalls. However, software systems are increasingly becoming more complex, connected and extensible. These properties are like a double-edge sword. On one hand they provide greater potential for what software systems can actually achieve and the services they provide. On the other hand it increases the risk of the system being compromised by attackers and it renders conventional methods of addressing security concerns insufficient [34]. A *secure software engineering* process needs to be deployed in order to address security needs of complex systems. Secure software engineering prescribes that security concerns should be addressed as early as possible in the development life cycle [16]. Proper secure software engineering should begin at the requirements engineering phase, resulting in security mechanisms being *designed* into the system. Having security designed into a system, with a strategy of defense in depth [72], will make the system less vulnerable to attack. Systems developed using secure software engineering approaches are no longer solely reliant on external generic defensive mechanisms.

The UML [55] is the de-facto modeling language for developing object-oriented software systems. UML provides a catalogue of different requirements and design artifacts than can be used to model the requirements and design of a system using different views and perspectives. According to the latest version of the UML (version 2.4.1) [55], the diagrams provided by the UML still do not address important security related semantics. Consequently, UML diagrams lack notational constructs that can be used to accurately communicate and model security related semantics. This lack of preciseness can lead to confusion and misinterpretation during the requirements and design phases, ultimately leading to the developing of an insecure system. Nowadays, an insecure system is most likely deemed useless even if a system performs its business related functionality flawlessly.

To counter this limitation of the UML, many research works were directed towards extending UML diagrams with notational constructs that model security aspects [34], [35], [37], [46], [52], [68], [69], [70], [62]. Other research works were directed towards devising security modeling techniques and notations that were not based on the UML [1], [29], [36], [43], [64], [74].

The abovementioned extensions offer various useful perspectives for security analysts. As part of effective security analysis, it is important to support state-based security analysis techniques. It can be useful for security analysts to have a state-based perspective of security issues. This would

likely be the case when analyzing systems with state-dependent behavior. However, except UMLsec [34], which partially offers some security extensions related to statecharts, the modelling techniques in the literature so far have integrated security aspects into notations that do not model state based behavior. The notational extensions that UMLsec offers for statecharts are also limited. There are stereotypes that can be attached to UML diagrams (for instance state diagrams), giving security-related restrictions such as "data security" or "no down-flow", indicating that confidential information should not leak. However, UMLsec does not offer any stereotypes to distinguish between different types of states. From a security perspective, a major distinction here would be between "normal" states (when the system is not attacked or compromised in any way), and abnormal states, where the system could be, e.g., vulnerable, threatened, compromised or recovering from an attack. The ability to easily distinguish different types of states would be useful both for security experts, to help them target their efforts, and to other stakeholders to heighten their awareness of security challenges. As long as states are not explicitly distinguished, every state in a statechart would visually look the same, except that the name labels might indicate some states to be more critical than others. Such distinctions would be much more difficult to spot, especially in large statecharts.

The ability to include security concerns in statecharts or to use the modelling of states to analyze security vulnerabilities has been seen as interesting [34], [17], [4], and security-related distinctions between states—such as normal, vulnerable, and compromised states, have been proposed by, e.g., [40], [2], though in these works not in the context of diagram visualization. Many security vulnerabilities are state dependent [2], [4], i.e., the vulnerability exists only in a certain state or can only be exploited when the system is in a certain state. For instance, a time-of-check, time-of-use (TOCTOU) race condition vulnerability can only be exploited in the time interval between check and use, and a vulnerability for which a patch has been distributed may only be exploited until the patch is installed. Due to this temporal nature of many vulnerabilities, it is interesting to analyze them from a state-transition perspective. Sometimes the vulnerabilities do not result from a single insecure state in one part of the system, but from a combination of several weakened states in different parts, so that the vulnerability can only be fully understood from a thorough analysis of possible concurrent system behaviours [4]. For example, real-life hacker intrusions such as the ones described in [50] are usually complex and opportunistic, seeking out and exploiting such insecure states—often the results of multiple simultaneous weaknesses—in certain parts of a system to create new and possibly more severe vulnerabilities elsewhere, which can then be exploited further in subsequence intrusion steps. Whereas existing work [37] have extended UML sequence diagrams to be able to describe such complex attack chains in more detail, security statecharts offer an even more detailed way of describing the complex dynamics of security threats, providing a bridge between informal, user- and function-oriented misuse cases [70] and formal systems analysis. In addition to offering to understand security vulnerabilities and intrusions better, security extensions to statecharts also offer to support risk and impact analysis, by providing a way of analyzing the consequences of security violations and how a security breach cascades, likely causing further damage to a system and its environment.

In this paper we thus attempt to tackle this issue by undertaking a theoretical approach to extend the popular UML statecharts notation with a new set of notational constructs. Such a proposal must of course be carefully justified. There is a number of modeling languages around, so one can fairly question the need for yet another one. In an investigation by Fettke [21] it was shown that users (or potential users) of modeling languages felt that the plethora of modeling methods was a barrier to success. In that light, it is fair to ask if yet another language proposal, as in the current paper, is a justified effort. However, Fettke also found the two most important success factors of conceptual modeling to be the expressiveness of the modeling language, and its ability for multi-perspective modeling. Hence, providing extensions to an existing modeling language, and thus increasing its expressiveness—which is the proposal here—would be more in line with these findings than proposing an entirely new modeling language. Our notation extends one particular diagrams of the UML, which offers a multi-dimensional suite of diagrams. The essential challenge is whether the proposed extensions are important enough, insufficiently covered by available modeling languages, and whether they can be used effectively by system developers while performing relevant tasks. An argument in favor of distinguishing abnormal states (e.g., vulnerable or compromised) from normal states, is that this can help developing systems with a better *defense in depth* [72]. In an ideal world, if a system manages to fulfill all its security requirements (as specified for instance through UMLsec stereotypes [34]) it should never end up in e.g. a "compromised" state. However, it is well-known that more or less successful security attacks happen all the time. Having the ability to keep on defending even if one line of defense is broken, or if part of the system is compromised and under control of an attacker, a system may be better able to limit the harm of partially successful attacks. Notational extensions thus to highlight abnormal yet sometimes unavoidable states may help developers find new threats. While the above text argues for the intuitive importance of distinguishing different types of security-related states in a statechart, the more detailed argument that this proposal is sufficiently different from existing ones must be dealt with in the discussion of related work, and the claim that developers would be able to use the extended notation (EN) will be backed by empirical data presented later in the paper.

Model comprehension and construction are two distinct activities. Therefore, they would require two separate experiments to properly investigate; they cannot be combined together into one experiment. This paper focuses on the model comprehension aspect of the new notation. We believe that this conjuncture should be investigated prior to the model construct aspect simply because if the new notation cannot be effectively read, then there is no point in advocating its use to perform model construction.

The remainder of this paper is organized as follows: Section 2 discusses related work on model driven security, with a general overview of other security-related graphical

modeling techniques, and a more detailed comparison with UMLsec, which is the most closely related one since it also addresses statecharts. Section 3 presents the theoretical approach undertaken to develop the new notational constructs. In Section 4, a real-world case study pertaining to the AndroidOS.FakePlayer malware affecting mobile phones is used to demonstrate the application and expressiveness of the new notation. In Section 5 we present the results of an industrial survey that was conducted using professional security analysts to determine the perception and coverage of the proposed notation. In Section 6 we present an empirical study that evaluates the cognitive effectiveness of the new notation using software engineering professionals as subjects. Finally, Section 7 concludes and discusses future work.

2 RELATED WORK ON MODEL DRIVEN SECURITY

Graphical models have been used in secure software engineering for a long time. In the related area of safety analysis, fault trees were introduced already the early 1960's [19]. In the 1990's similar approaches were proposed in the security area, namely threat trees [1] and attack trees [64]. More recently such tree notations have been extended further, for instance to attack-defense trees [39] showing both the envisioned attacks and their possible countermeasures. Common for all these tree-based techniques is a top-down analysis focusing on events, where an unwanted or maybe even disastrous top level event is decomposed to smaller events that may cause it through AND/OR gates. A finished tree will thus depict how an unfortunate combination of small leaf node events could make the top-level event come true. The naming of an event may implicitly indicate states and state transitions, e.g., in the attack tree example of a burglar opening a safe, the event "Learn Combo" would implicitly hint about a state change from "Combo secure" to "Combo compromised". However, states and state transitions are not shown explicitly in these trees, so the view they give of a system is very different from that of a statechart, and the usage of these models in the system analysis and design process will thus also be very different. Hence, trees and statecharts supplement each other rather than being competitors.

In the new millennium a number of graphical modeling languages have been proposed to incorporate security considerations in the mainstream requirements and design effort of software systems. Mostly these have been adaptations of already existing modeling languages, providing them with security-related concepts and notation extensions. Many of these proposals looked at various languages in the UML family of modeling languages. An early proposal was misuse cases [68] extending UML use cases, with further extensions by [62] to capture vulnerabilities and insider threats. A notable proposal specifically targeting model-driven development was UMLsec [34], which is a set of UML profiles providing security extensions for several different types of UML diagrams, e.g., deployment, class, sequence, activity, and state diagrams. UMLsec has recently been updated to conform to UML2.3 in a profile called UMLsec4UML2 [63]. Other proposals including UML profiles are SecureUML [46] and the work by Burt et al. [10]. Whereas UMLsec provides profiles for several different UML diagrams, these two other works focus on profiles

related to class diagrams, and specifically targeting access control. In a follow-up work the originators of SecureUML presented an approach for modeling security of process-oriented systems [6]. In that paper, a meta-model including states is shown as Fig. 4, but the purpose of that meta-model is to support the capturing of the system process as a hierarchy of states for the integration with SecureUML, thus not representing a security extension of state diagrams as such.

Other more recent UML-related language extensions include mal-activity diagrams [68] extending UML activity diagrams, and misuse sequence diagrams [37] to extend UML sequence diagrams. Outside the UML family, primary examples of language extensions for security are abuse frames [43], Secure i* [45], KAOS SE (Knowledge Acquisition in automated Specification—Security Extension) [74], secure tropos [52], and misuse case maps [36]. For the latter, it should be noted that although the name might sound related to UML, they are an extension of use case maps [9], which is a notation very different from use cases.

Of all the proposals mentioned above, the only one to offer security extensions for statecharts is UMLsec [34]. However, the extensions offered by UMLsec are quite different from those targeted in this paper. UMLsec defines a number of security related stereotypes, such as "secrecy", "secure dependency", "critical", "no down-flow", "data security", and "fair exchange", and these may be added to various UML diagrams. Such stereotypes imply security requirements, and used with a statechart they will constrain the allowed behavior of the system. The purpose is then to analyze the statechart to see if its current definition violates the stereotype. An example given in [35] (p. 420) (see Fig. 1) is a small example of a banking application, the diagram with two states has been stereotyped with «**no down-flow**». This (and other UMLsec) stereotypes are not associated with single states, but with the state diagram as a whole. The «no down-flow» stereotype accompanied by the tagged value {secret = {wb,rb}} just underneath it, means that confidential information should not leak by being communicated through less secure operations. An analysis of the given diagram, as explained in Jürjens' example, shows that there is a potential leak, as parts of the information of a secure method called rb() is also made available through a less secure method called rx(), both these methods being attached to state transitions in the diagram. From this it can be seen that UMLsec stereotypes enhance statecharts by formally stating requirements that will constrain the allowed behaviour of the statecharts, and help the analyst discover defects in the model that might violate these specified requirements. This is clearly a useful technique for the systems analyst, but it covers a purpose quite different from the stereotypes proposed in this paper. Our stereotypes are connected directly to state nodes in a diagram, not to a larger collection of state nodes, and their purpose is not to imply requirements to constrain the state behavior, but to *distinguish between different types of states*. Of course, if a state is stereotyped as "compromised", this will be a clear hint for the analyst to specify requirements that can help the system avoid getting into that state—but the stereotype itself does not say what these requirements might be. In that sense, our proposal is on a higher level of abstraction than UMLsec. Also, just as important as to avoid getting into a compromised state, such
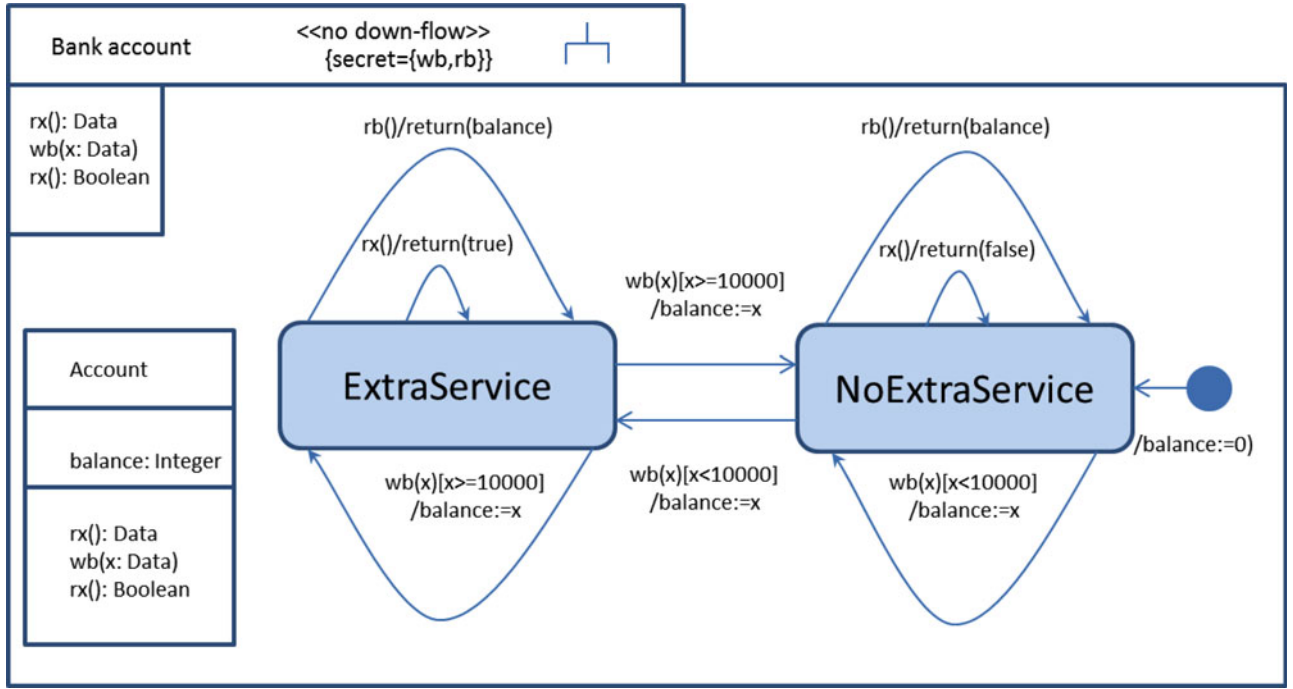
Fig. 1. Example UMLsec diagram from [35].

a statechart can also indicate where the system should go next, if getting into a compromised state anyway. Hence, rather than being competitors, UMLsec and the proposal in this paper could be seen as complementary. It would be possible to use the stereotypes to be proposed in this paper to distinguish between different types of states, and at the same time use UMLsec stereotypes to formulate more detailed requirements for the possible transitions between the states.

In addition to this conceptual difference in what the stereotypes are used for, there is also an obvious notational difference between UMLsec and the proposal in this paper. UMLsec uses standard UML statechart notation, only amending it by textual labels for stereotypes such as «no down-flow» and corresponding tagged values like {secret = {wb,rb}} from the discussed example. Of course there is nothing wrong with using the standard notation as much as possible, and this probably was a good choice for the intended usage of UMLsec. The extensions to statecharts proposed in this paper, however, have a quite different purpose. Along with the wish to distinguish abnormal states from the normal ones follows an intuitive wish to make the abnormal states stand out in the diagram, by means of graphical highlighting. As with the conceptual basis, the notational elements of UMLsec and the current proposal are also complementary and could be used together. Empirical investigations about using them together would be interesting, but are beyond the scope of this paper, as it seems natural to first evaluate the proposed extension alone.

Another main advantage of using the new notation is that it can be specified in the underlying metamodel of statecharts, hence enabling more in-depth formal analysis of the models. Formal semantics incorporated in a metamodel can will allow the development of syntax rules that stem of the security theories. Facilitating in-depth modeling analysis can allow modelers to automatically identify various issues in their models, for example, vulnerable or threatened states

that are not eventually mitigated. A more specific metamodel can be used to detect syntax errors, mistakes and antipatterns. A formalized model can also be used to define and collect various security metrics. The formal semantics of the proposed notation inherits (and extends) the formal semantics of the standard UML notation.

3 EXTENDED STATECHARTS NOTATION

This paper proposes an extended statechart notation to enable visualization of security aspects such as threats and vulnerabilities. The original statechart notation only defines one type of state and event. The type of state and event is therefore generic in nature and it is most commonly used to model system-related behavior. Security analysts who need to use statecharts are limited to using the current notation. If modelers elected to model security aspects using the current notation will contain many instances of *symbol overload*. Symbol overload is defined as the use of one notational construct to visualize multiple semantic constructs [51]. Symbol overload is a very serious drawback in any notation as it leads to ambiguity and the potential for misinterpretation [23]. As such it is useful to be able to model security aspects in statecharts using new notation. The design of a new software engineering notation (or the extension of an existing notation as is the case in this paper) should not be ad-hoc or based solely on aesthetics. The design of a notation is a predominant factor of its success (widespread adoption) or failure. Therefore, notation should be designed based on scientific principles. The approach undertaken in this paper is two-phased (see Fig. 2). In the first phase, the required semantic constructs are introduced by identifying phenomena or aspects from the problem domain (Section 3.1). In the second phase, new visual constructs are introduced (Section 3.2) based on the semantic constructs developed in the former phase.
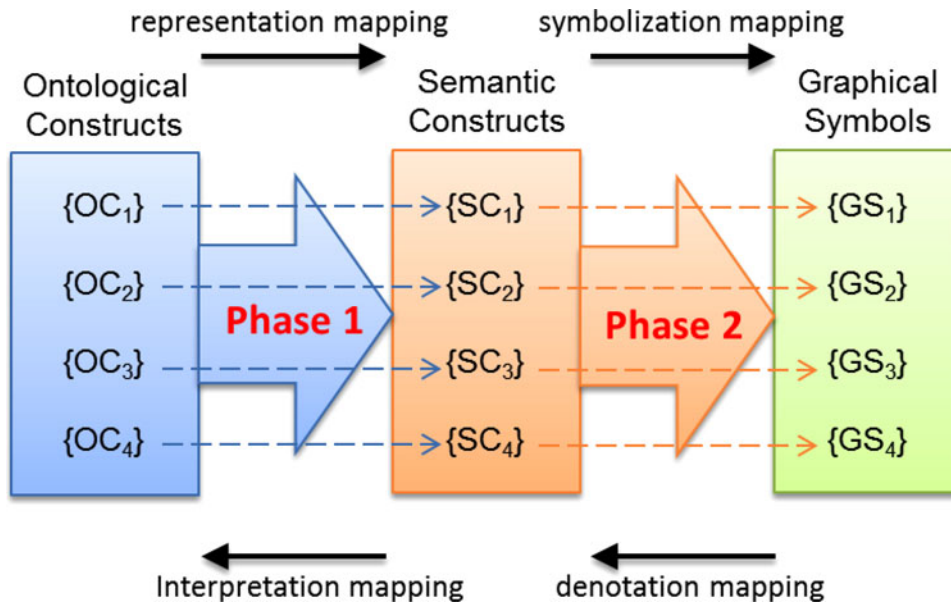
Fig. 2. Overall process required to develop the new notation [51].

3.1 Phase 1: Defining the Necessary Semantic Constructs

This paper aims to extend the statechart notation to model security aspects and hence the domain considered in this paper is the security domain. During this phase an ontological analysis of the security domain is performed in order to identify the new required semantic constructs. When introducing new semantic constructs, the goal is to achieve a one-to-one mapping between ontological and semantic constructs otherwise any of the following four anomalies can occur [75]:

- *Construct overload* occurs when a particular ontological concept does not have a corresponding construct in the notation.
- *Construct overload* exists when multiple ontological concepts can be represented by a single notation construct.
- *Construct redundancy* exists when a single ontological concept can be represented by multiple notation constructs.
- *Construct excess* exists when a construct in the notation does not correspond to any ontological concept.

A recent literature survey (in 2012) reveals that there exist many security ontologies, which can be classified into eight different categories [71]. The ontologies considered in this research work were limited to those listed as "General Security Ontologies" since the aim is to develop a generic notation to model generic security aspects for any domain. The main advantage of considering ontologies in this category is that they capture most security aspects due to being generic. Security ontologies in other categories contain a great deal of specificity pertaining to particular security domains and hence were excluded from this study. Within the "General Security Ontologies" category, two security ontologies were identified. The first ontology is proposed by Herzog et al. [30], while the other ontology is proposed by Fenz and Ekelhart [20]. A security extension of Statecharts has to couple the most central concepts from security

ontologies such as Herzog's [30] with the most fundamental Statechart constructs: state, transition/even and initial and final node. Of Herzog's top-level security concepts (Threat, Countermeasure, Vulnerability, DefenseStrategy, Product, Asset, Model, NaryRelation, Goal, TimeOfEmployment), we identified the two first as possible subtypes of transitions (or events), leading to two new semantic constructs: threat event and countermeasure event. We identified the next two concepts from Herzog's ontology, as well as threat, as possible subtypes of states, leading to three more semantic constructs: threat state, vulnerability state and defensive state. We also devised specific subtypes of initial and final nodes for describing threats. The new semantic constructs that were thus identified are summarized and defined in Table 1, along with three additional types of state that are useful for describing security breaches: compromised state, quarantine state and recovery state. Based on the ontology by [30], nine new semantic constructs were identified that are related to state-based behavior. Other ontological constructs that do not relate to state-based behavior were excluded. The new semantic constructs have a one-to-one mapping with the ontological constructs. The new semantic constructs are outlined and defined in Table 1.

3.2 Phase 2: Developing the Necessary Graphical Constructs

In this phase, we introduce new graphical constructs to correspond to the nine semantic constructs that were developed in the previous phase. Graphical notation design is ostensibly an issue of taste and aesthetics. However, empirical studies have shown that the visual form of notations significantly affects understanding especially by novices [31], [32], [33], [48], [54], [58], [59]. Research has repeatedly shown that the form of representations has an equal, if not greater, influence on cognitive effectiveness as their content [41], [65], [80]. In [51], a set of nine principles for designing cognitively effective visual notations were presented (see Fig. 3). These principles were synthesized from theory and

TABLE 1
New Security Related Semantic Constructs for Statecharts

Semantic Construct	Definition
Threatened state	A system reaches this state when exploitation is attempted while lacking the necessary defensive mechanisms.
Vulnerable state	A system reaches this state if it is vulnerable to an attack meaning that it lacks the necessary defensive mechanisms to defend itself in case an attack occurs. This state however does not imply that an attack is occurring or imminent.
Defensive state	A system reaches this state when it is expecting or experiencing an attack. The system executes the necessary defensive mechanisms to fend off attacks.
Compromised state	This state indicates that the system has been compromised. It indicates that damage has occurred and is still occurring.
Quarantine state	A system reaches this state when damage from an attack cannot be recovered. A system reaches this state as a measure of damage control.
Recovery state	If a system is compromised it can reach the "Recovery" state to regain its integrity. Once the system's integrity is restored, the system leaves this state and goes back to a normal state.
Threat event	An event that occurs with the intent to exploit or harm. These events try to transition a system into a "Vulnerable", "Threatened" or "Compromised" state.
Countermeasure event	An event that occurs to prevent a potential attack or remedy the effect of an attack. These events try to transition a system into a "Defensive", "Recovery" or "Quarantine" state.
Initial threat node	This initial node is used when the state machine is initiated due to a malicious intent.
Final threat node	This final node is used when behavior of the state machine ends while in a "Compromised" state.

empirical evidence. Prior to presenting the notation evaluation of misuse case modeling, it is important to present the justification of using the PoN instead of the cognitive dimensions (CDs) framework, perhaps the closest to a theory of visual notation design that currently exists in the IT field [11], [27]. The CDs framework is considered the predominant theoretical paradigm in visual languages research. However, the CDs framework has theoretical and practical limitations for evaluating and designing visual notations [51]. The limitations of CDs framework with respect to being a scientific basis for evaluating and designing visual notations are as follows:

- One of the main issues with CDs is that they do not have clear evaluation procedures or metrics. Hence, users of the CDs can only apply it in a subjective manner [13].
- The CDs are based solely on structural properties and hence they exclude visual representation issues [12].
- There is no notation evaluation support provided by the CDs as they simply define properties of notations without indicating if they are either "good" or "bad" [12], [26].

- The CDs framework only applies visual notations to as particular class of **cognitive artifacts** and is not specifically focused on visual notations [26].
- The dimensions are vaguely identified leading to confusion or misinterpretation or when applying them [13], [26].
- The CDs framework suffer from poorly defined theoretical and empirical foundations [27].
- The CDs do not provide design guidelines and hence they do not support notation design. Consequently, issues of effectiveness are excluded from its scope [12], [26].

The CDs framework played a vital role in the development of the Physics of Notations. The Physics of Notations can be considered as an advancement of the CDs framework, whereby the Physics of Notations provide a more powerful domain-specific theory that has originated from the CDs framework theory.

This phase requires an analysis of the relations between the new semantic constructs. An analysis of the relations between the new semantic constructs and the preexisting ones is also required. These analyses are a prelude to developing a visual language that conforms to the principles presented in [51]. Table 2 presents the proposed new graphical
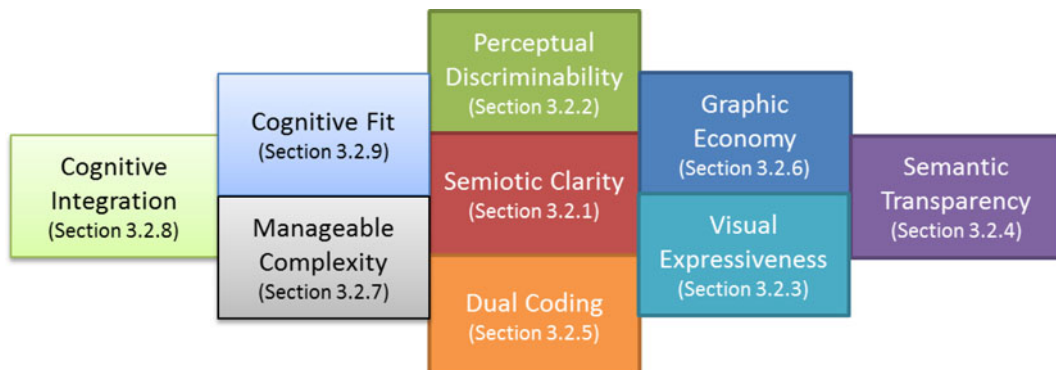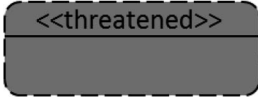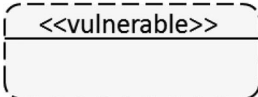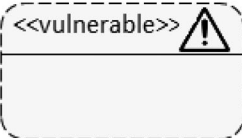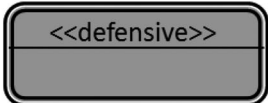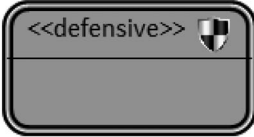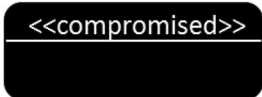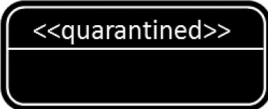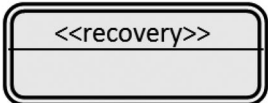

Fig. 3. The principles of designing cognitively effective visual notations [51].

TABLE 2
New and Existing Graphic Symbols for Statecharts

Semantic Construct	Graphical Symbol	Secondary Notation
Existing Notation		
State		*No specific secondary notation stated by the UML.*
Event		*No specific secondary notation stated by the UML.*
New Notation		
Threatened state	<<threatened>>	<<threatened>>
Vulnerable state	<<vulnerable>>	<<vulnerable>>
Defensive state	<<defensive>>	<<defensive>>
Compromised state	<<compromised>>	<<compromised>>
Quarantine state	<<quarantined>>	<<quarantined>>
Recovery state	<<recovery>>	<<recovery>>
Threat event		
Countermeasure event		
Initial threat node		*Same as primary notation.*
Final compromised node		*Same as primary notation.*

symbols. Table 2 also presents the proposed secondary notation for the new graphical symbols. A secondary notation is a notation form that is optional rather than mandatory. Secondary notations are commonly devised to further clarify the meanings of primary notation. Sections 3.2.1-3.2.9 present an evaluation of the proposed new graphical symbols with respect to each of the nine principles stated in [51].

3.2.1   Principle of Semiotic Clarity

Analogous to ontological analysis, the principle of Semiotic clarity states that there should be a one-to-one mapping between semantic constructs and graphical constructs. Semiotic clarity is therefore considered to be the primary principle of Moody's nine principles [51]. Failure to satisfy this principle results in one of the following four anomalies [51]:

- *Symbol redundancy* occurs when a construct can be represented by multiple graphical symbols.
- *Symbol overload* occurs when two different constructs can be represented by the same graphical symbol.
- *Symbol excess* occurs when a graphical symbol does not correspond to any semantic construct.
- *Symbol deficit* occurs when a semantic construct does not have a corresponding graphical symbol.

There are nine new symbols used to model the nine new semantic constructs while maintaining a one-to-one mapping and hence satisfying the principle of semiotic clarity.

3.2.2   Principle of Perceptual Discriminability

The principle of perceptual discriminability is concerned with the ease and accuracy with which different symbols from the same notational set can be differentiated from each other. A diagram can be accurately interpreted only if its symbols can be accurately discriminated. A notation design should aim to increase the **visual distance** between its symbols to maximize discriminability. Discriminability is measured by the number and ranges of visual variables used. The greater the differences of the number and ranges of visual variables used by different symbols, the greater the **visual distance**. Fig. 4 presents the eight elementary visual variables which can be used to decode information [7].

The *shape* variable is the most influential of all visual variables. Shape is the predominant factor humans use to classify objects in the real world [51]. This means that different shapes are perceived to present categorically different semantic constructs, while differences in other visual variables are perceived to present different but somewhat similar semantic constructs. Therefore, similar shapes should be used to represent the same or similar constructs. The ontological analysis performed revealed a set of nine semantic constructs that can be classified into three categories: states, events and pseudostates. All three of these semantic categories preexist in statecharts. Therefore, it was clear to us that the new notation design should use the same three shapes for the semantic categories; namely rounded rectangles for states, arrows for events and circles for pseudostates (see Table 2). Not to
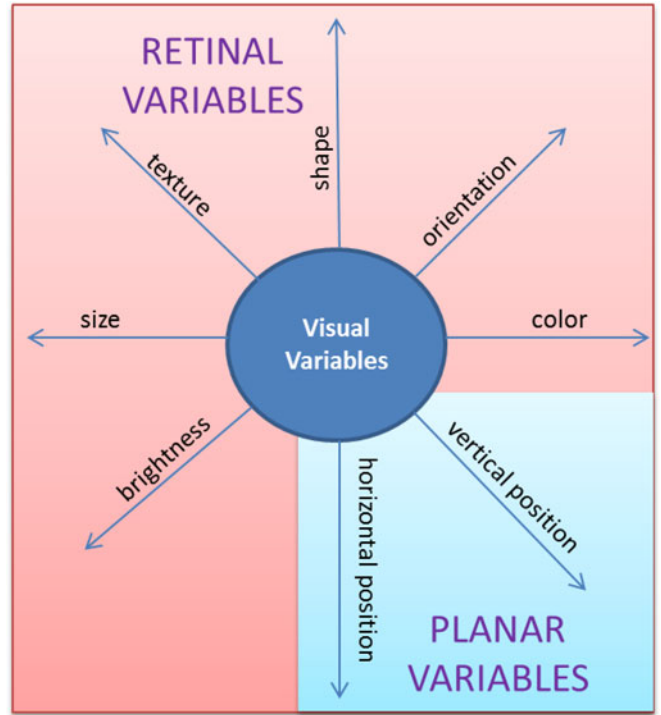
Fig. 4. Visual variables.

be deterred by this limitation, the new notation design makes effective use of other, also influential visual variables, namely color, brightness and size. A detailed discussion about the utilization of these other visual variables is provided in Section 3.2.3.

3.2.3   Principle of Visual Expressiveness

The principle of visual expressiveness is concerned with the absolute utilization of the graphic design space. A notation is considered visually expressive if it utilizes a large number of visual variables and uses a wide range of values (*capacity*) of each variable. This principle differs from the principle of perceptual discriminability, which is concerned with the visual distance between symbols within the same notation (pairwise comparisons). Although we did not use new shapes for the new notation design, we were able to leverage three other visual variables. The new notation makes use of the color, size and brightness variables. A discussion of the use of each variable is provided below:

Color. Color is considered to be one of the most cognitively effective visual variables. Studies have shown that the human visual system is highly sensitive to variations in color [47], [78]. Humans can accurately and quickly distinguish between different colors. However, if color is misused it can have a misleading effect. The field of color psychology offers some clues about the specific meaning of colors as perceived by humans. However, the field of color psychology has long suffered with the inherent difficulties in properly controlling trials of color's effect on human beings meaning that findings from the field cannot be generalized with certainty. As such, the use of color in our design uses findings from color psychology that are reinforced by anecdotal evidence. Table 3 outlines the colors used in the new design:

TABLE 3
Colors Used in the New Notation

Color	Rationale and Use in the New Notation
Red	From a sensory perspective, red is associated with love [Gorn et al., 1997], lust [Gorn et al., 1997] and excitement [O'Connor, 2011]. However, from a functional perspective, red is associated with negative issues. Red prompts a state of alarm. Red has already been used in many security modeling languages, including misuse sequence diagrams [Katta et al., 2010] and misuse case maps [Kárpáti et al., 2010], to model exploitation attempts [Kárpáti et al., 2010; Katta et al., 2010]. Red is also used by the well-known Homeland Security Advisory System (HSAS) color chart to indicate a "severe" risk of an attack. Red also indicates evil intent and therefore it is unanimously used as the color of the devil. Therefore, the new notation design uses the color red to represent states of type "Threatened", and to represent "threat" events. The color red is also commonly used in Risk Analysis matrixes. Cells that represent risk of high likelihood and high impact are colored red.
Yellow	The color yellow is synonymous with caution. The HSAS color chart uses yellow as an elevated threat level in comparison with the "guarded" level, meaning that there is a vulnerability but without there being an imminent or current threat as implied by the color red. Therefore, the new notation design uses the color yellow to represent "Vulnerable" states.
Blue	According to Color Psychology, blue means competence [O'Connor, 2011] and masculinity [Gorn et al., 1997] from a sensory perspective, and it means high quality from a functional perspective [Gorn et al., 1997]. The HSAS color chart uses blue to indicate a "Guarded" state. Therefore, the new notation design uses the color blue to represent "Defensive" states. The color blue is also used in the new notation to represent "countermeasure" events.
Black	According to Color Psychology, black indicates grief and fear [Gorn et al., 1997], and it means costliness from a functional perspective [Gorn et al., 1997]. These meanings are in-line with aspect of damage or harm that is currently taking place or has taken place. Black is the negation of white, which indicates a pure healthy state [Gorn et al., 1997]. The color black is used in many security modeling languages to indicate misuse, including misuse cases [Sindre, 2005] and mal-activity diagrams [Sindre, 2007]. Therefore, the color black is used in the new notation to model "Compromised" states. The color black is also used to model "Quarantined" states as quarantine is a damage control countermeasure meaning that damage has already happened.
Grey	In the new notation design the color grey is used to model "Recovery" states. The color grey is chosen for this purpose simply because grey is the midway color between white and black. This indicates that the system is currently progressing towards a healthy normal state (white) from a negative state (black).
White	Normal state.

A benefit of using the colors red, yellow and blue is that they are three primary colors thus exhibiting high discriminability. The color blue is often considered to be the opposite of red (such as representing hot and cold in water taps), which more accurately reflects the different meanings of the states. The color green is also often considered to be the opposite of red. However, the green also often implies that conditions are satisfactory, which is not the case in "Defensive" states and "countermeasure" events. Another advantage of not using the color green is to avoid the most common color-blindness problem, which is the inability to distinguish red from green.

We ultimately hope that the new notation, or at least a subset of the new notation, becomes standardized by the OMG (Object Modeling Group) [55]. One of the main issues that OMG has against adopting notation that utilizes color is its readability by people who are color-blind. We argue that the color-blind may still find the new notation useful since it uses many visual variables other than color, such as shape and brightness. It is highly unlikely that color-blind

people will find models created using the new notation actually less readable. Of course, another empirical study is required to provide the proof.

Size. Size is an ordinal visual variable which can be leveraged to encode different information. For the "Defensive", "Quarantine" and "Recovery" states, a thicker boundary is used to signify a protected state. In this sense, size is used to indicate different protection levels. To avoid confusion with the existing UML notation of *active objects*, the borders of these states are double-padded, which also signifies extra protection. The boundary size of the "Threatened", "Vulnerable" and "Compromised" states are the same as the existing notation for a state. This design decision is made to indicate that the system is not receiving any protection while in any of these states. The arrow sizes of the "Threat" and "Countermeasure" events are bigger than a regular event. In this sense, size is used for emphasis, whereby "Threat" and "Countermeasure" events should be emphasized given the importance of security in a system. The "Threat" and "Countermeasure" events both share the same size to indicate that they are of equal importance. Note that the arrows were not made much thicker as they may clog the overall view of the diagrams.

Brightness. The term *brightness* is commonly interpreted as the strength of a particular color. However in the relative literature, the term *brightness* is used to refer to solid versus dashed lines. The brightness visual variable is used to indicate the lack of protection in the "Threatened" and "Vulnerable" states. These state boundaries are depicted using dotted lines rather than solid lines. The use of dotted-lines implies gaps and hence vulnerabilities in a defense. Dotted-lines have been used in other security modeling languages to indicate vulnerability, such as misuse sequence diagrams [37] and misuse case maps [36]. The "Compromised" state is also open to further attacks, however, when damage is already occurring or has occurred, then encoding information about vulnerability is useless because the current situation has exacerbated. This concept emphasized visually by the "Compromised" state, whereby if it would have had a dotted-line boundary, then the boundary would be invisible given the black background.

3.2.4   Principle of Semantic Transparency

The principle of semantic transparency refers to the use of graphical representations whose appearances are suggestive of their meaning. For example, using an icon of a medical doctor in a use case model is a clear indication that the actor represented by the icon is a real-world medical doctor. Using semantically transparent graphical representations speed up recognition and improves intelligibility, especially for novice users [8], [48]. However, caution should be exercised when using semantically transparent graphical representations as a wrong graphical representation can actually be misleading rather than helpful, which is worse than not having any semantically transparent graphical representations in the first place. Some graphical representations can be cultural or domain dependent. The design of the proposed notation extension aims to be globally applicable and therefore it intentionally avoids using representations that are better understood in some countries than others. For

example, a red-cross symbol is used by hospitals in Christian countries, while a red half-moon symbol is used by hospitals in Muslim countries. However, the 'H' sign is a common sign for hospitals in all countries.

Icons. The design of the new notation makes use of icons. However, icons are only limited to being a secondary notation. The rationale behind this design decision is that graphical icons require a high degree of artistic skills which is unlikely to be possessed by users of software engineering notations. Consequently, mandating the use of graphical icons will hinder the adoption and the standardization of the new notation. If tool support is provided for the new notation then using icons will have an added value. In fact, icons and 3D shapes have been shown to be cognitively and perceptually very effective [5], [32], [78]. The following icons are used in the new notation:

- *A shield.* A shield is a sign that is synonymous with defense. It is used by popular antivirus and firewall software to indicate that the operating systems and other virtual assets are protected from intrusion. The shield is therefore used to annotate the "Defensive" state and "Countermeasure" event symbols.

- *A caution traffic sign.* A caution traffic sign is used to draw attention to a potentially hazardous situation. However, as it is the case with actual traffic laws, a caution traffic sign does not mandate or forbid any particular action; it only warns and recommends caution. The caution traffic sign shield is therefore used to annotate the "Vulnerable" state symbol in order to draw attention for countermeasure but not in the severe sense as an imminent threat.

- *A lightning bolt sign.* In the real-world, a lightning bolt is a scary (for many people) and life threatening phenomenon. A lightning bolt is therefore commonly used to indicate threat and danger in general. The lightning bolt is used to annotate the "Threatened" state and "Threat" even symbols. The lightning bolt sign is used in KAOS models to represent conflicting requirements meaning that each requirement poses a threat to the other [14].

- *A skull and bones sign.* This sign is indicative that damage has already occurred or is currently occurring.

- *A hospital sign.* Hospitals are a place for healing. Once a patient is healed they no longer stay at the hospital. Analogously, once recovery is complete the system transitions to another state that is not concerned with recovery. Therefore, the hospital sign is used for the "Recovery" state symbol.

- *The caution tape sign.* Caution tape is used to quarantine an area where an incident has occurred. The tape is strung to prevent any further incidents occurring whether intentionally or unintentionally. The caution tape sign is used to annotate the "Quarantine" state symbol.

Double-padded state borders. Padding is a natural mechanism to repel and defend against external entities such as cold weather or a flooding. The "Defensive", "Quarantined" and "Recovery" states have double-padded borders. In these states, protection is applied to repel and defend against further attacks.

3.2.5 Principle of Dual Coding

While the use of text in diagrams is ostensibly forbidden, it is actually encouraged to use text to supplement, *not* replace graphics. According to *dual coding theory*, using a combination of graphics and text is more effective than using either of them alone [56]. Therefore, a general guideline to text use in notation is that text should be used as a form of redundant coding to reinforce and clarify meaning. Text should not be used as the sole basis for differentiating between symbols. The new state types introduced in the new notation include stereotypes that indicate their type. The use of stereotypes was chosen as it is a formal extension mechanism provided by the UML. However, in order to determine whether or not the principle of dual coding is adhered to or violated, an evaluation of the entire notation (new and original) is necessary. This evaluation is performed by ridding all symbols of all text then determining if they can be distinguished from one another. The result of this evaluation has shown that each symbol in the entire notation is distinguishable without text.

3.2.6 Principle of Graphic Economy

The principle of graphic economy refers to the number of different symbol categories used in a notation. Studies have shown that the human ability to discriminate between perceptually distinct symbols is around six categories [49], which define an effective upper limit for graphic complexity. It is beyond the scope of this paper to evaluate the original UML statechart notation with respect to the graphic economy principle (which it does satisfy because it includes only three categories). Instead, the focus here should be to evaluate whether the newly added notation makes the entire notational set exceed the upper limit of six categories. However, counting the number of categories is not necessary since all the symbols introduced in Table 2 belong to one of three categories in the original notation. Therefore, with the new notational set included, the number of perceptually distinct symbols remains at 3. If icons were included in the evaluation then the count would reach 4, which is still within the upper limit. However, icons were assigned as secondary notation so they will not count.

3.2.7 Principle of Complexity Management

The principle of complexity management states that a notation should contain mechanisms within its visual vocabulary to effectively control complexity. The new notation, namely the new states types introduced, inherit the complexity mechanisms already defined by UML statecharts. UML statecharts define the concept of *superstates*. A superstate is essentially a container of other state machines. This means that statecharts containing the new state types can have their complexity managed effectively. The semantic concept of superstates can be formally added to the newly added state types.

3.2.8 Principle of Cognitive Integration

The principle of cognitive integration states that a notation should provide mechanisms to achieve conceptual and perceptual integration. Conceptual integration is a mechanism that will allow the reader to fit statecharts within an overall cognitive map of the whole system. Perceptual integration is a mechanism that will allow the reader to effectively navigate between diagrams. The new notation does not explicitly introduce any new mechanisms to achieve conceptual and perceptual integration. The new notation inherits conceptual integration mechanisms from the original notation (ON) and the UML in general. For example, the UML allows use cases to be depicted as a package containing a state machine that visualizes its textual behavior. UML lacks any mechanisms for perceptual integration. The UML also has Interaction Overview Diagrams (IODs) which are essential activity diagrams but whose nodes can contain other types of interaction diagrams. According to the UML, nodes in IODs do not contain statecharts. IODs were however formally extended by [77] to contain statecharts. To date, this extension has not been standardized by the UML.

3.2.9 Principle of Cognitive Fit

The principle of Cognitive Fit states that different *visual dialects* should be used for different tasks and audiences. A visual dialect is a visual form that represents a certain set of semantic constructs. Some notations can exist in multiple visual dialects such as Data Flow Diagrams (DFDs). DFDs exist in two dialects: the DeMarco dialect [15] and the Gane and Sarson dialect [22]. Both dialects are semantically equivalent. Certain tasks may be easier with one particular dialect than the other. A particular dialect may be easier to understand by novices than another. Therefore, different visual dialects should be used when most suitable. Although there are competing notations that are semantically similar to statecharts, determining which dialect is the most suitable for a particular task or audience will require an empirical evaluation which is beyond the scope of this paper. Considering the notation of UML statecharts exclusively, UML statecharts only have one visual dialect. The design of the extended notation does provide a means for cognitive fit via the use of the secondary notation. Usage of icons can aid understanding by stakeholders who are not experienced with abstract modeling languages. Icons can later be removed when the models are used by developers whom do not need them. Another dialect can be created to cater for color-blind users by using different pattern fills. Pattern filling however requires a high degree of artistic skills and as such was not incorporated in the notation design but rather left as an option to be provided via tool support.

A summary of the proposed improvements and how they meet (or partially meet) the principles of PoN is shown in Appendix B, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TSE.2015.2396526 available online.

3.3 Accessibility

The new notation will have inherent accessibility problems. It is important note that the newly proposed notation is not intended to replace the original notation. The proposed notation is not intended to be used exclusively. Users of statechart modeling who cannot access the new notation
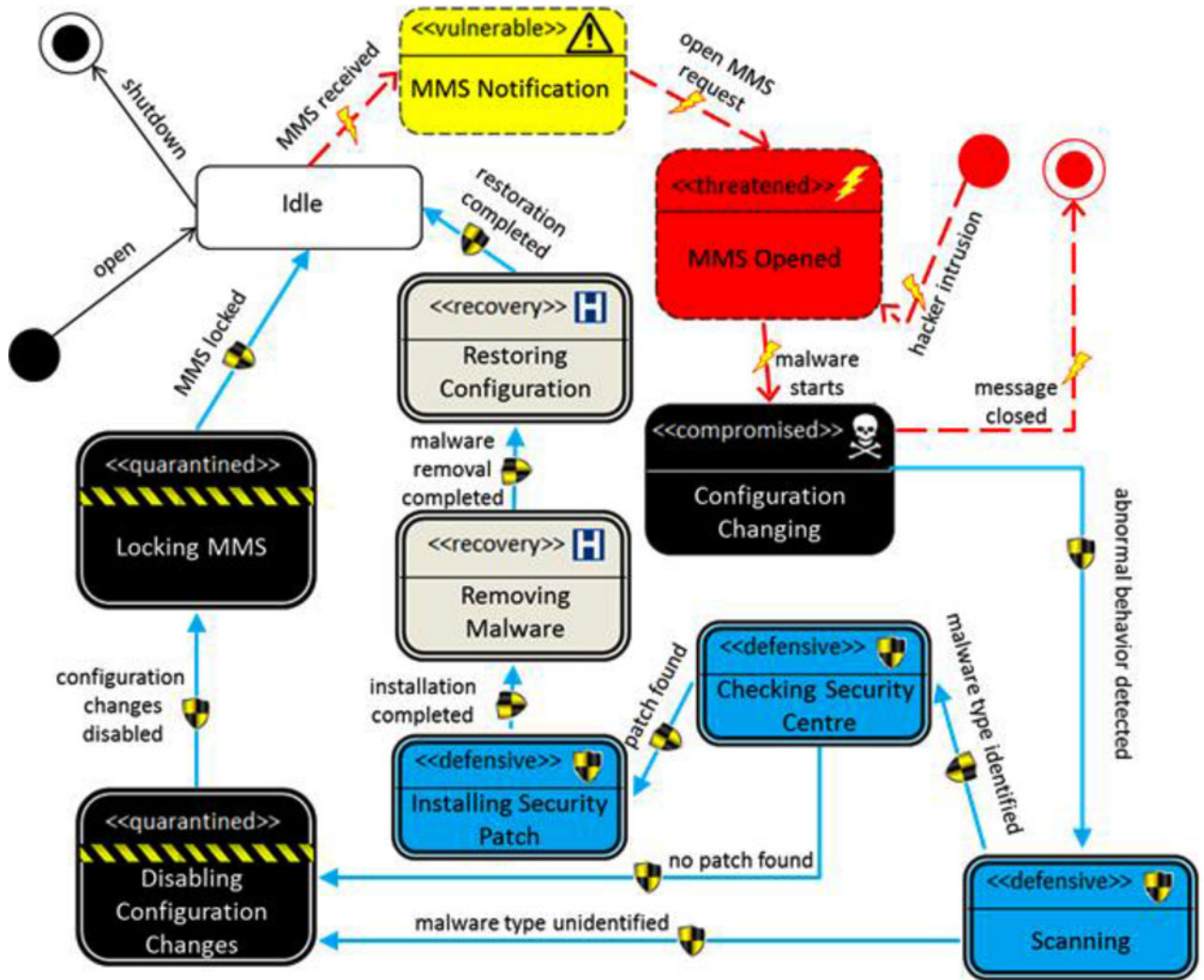
Fig. 5. The AndroidOS.FakePlayer malware threat state machine modeled using the new notation.

can simply revert to the original notation. As such, when it comes to hand sketching or black-and-white printing, we suggest users should use the original notation. If a user finds the original notation more readable than our proposed notation, then the user is encouraged to use the original notation. The same applies to users from the color-blind community.

Having mentioned these accessibility limitations, there remains numerous opportunities to leverage the benefits of the new notation. More specifically, when statecharts are viewed on computer screens, it is arguable that all computer screens use high-resolution color. Many users may also have color printers. Most printers that are designed to print large-sized posters to be displayed on a wall for teamwork purposes, do print in color. Existing UML tool support can be used to create notation that resembles our proposed notation, as evident by the diagrams we present in the paper which were developed by existing tools. In case the specific tool support is made for the notation (as already planned for future work), then users who prefer using UML modeling tools can use the developed tool.

Therefore, given that fact that the original notation will not be removed, users can leverage the benefits of our notation when permissible.

4   THE NEW NOTATION IN ACTION: JUXTAPOSING WITH THE ORIGINAL NOTATION

This section presents a real-world case study that motivates the introduction of the proposed security-extended state-chart notation.

4.2   Case Study Definition and Motivation
The main purpose of this case study is to compare the effect of using the new and original notations to model security aspects in-situ setting. Evaluation is mainly based on the modeling capabilities of the two notations rather than the perception of the model users. Evaluating the effect of the new notation on model users (creators and readers) is performed empirically in Sections 5 and 6.

4.3   Case Study Formulation
The context of this case study pertains to modeling the state-based functional security behavior of the AndroidOS.Fake-Player malware (see Fig. 5) [73]. The AndroidOS.FakePlayer malware is one of the eight worst pernicious Android malware attacks. Discovered in 2010, AndroidOS.FakePlayer allows an attacker to easily steal from users without their knowledge. The malware masquerades as a media player
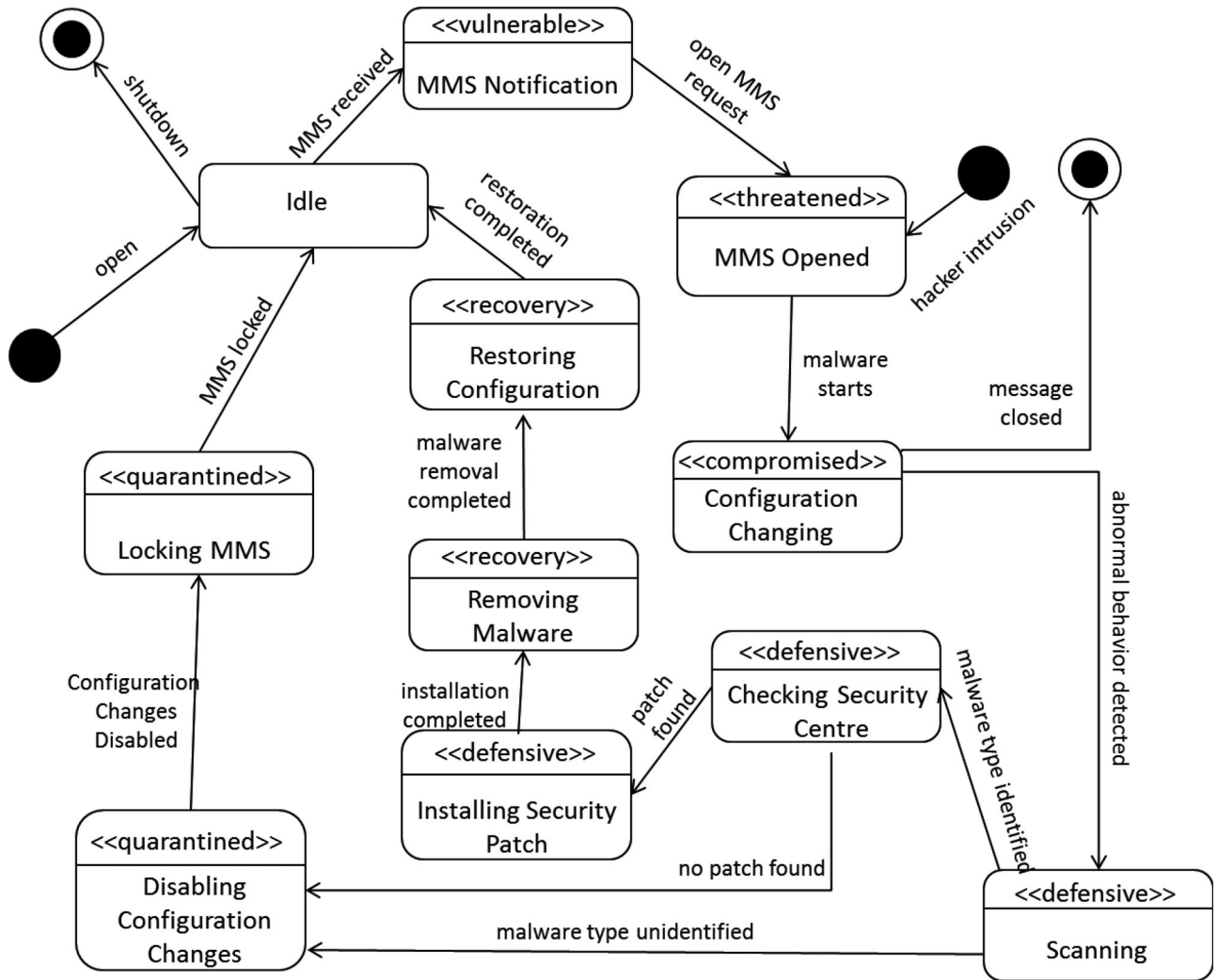
Fig. 6. The AndroidOS.FakePlayer malware threat state machine modeled using the original notation.

application ("*app*"). Upon installation the malware begins to silently send costly SMS (Short Message Service) messages to predetermined SMS numbers [73]. The purpose behind this case study is to model the state based behavior of an Android based system infected with the malware and to design countermeasures that allows it to defend against this malware. The requisite design describes the countermeasures also from a state based perspective. As a countermeasure, once abnormal behavior is detected due to unexpected change in configurations, the security center is then checked for anti-malware software. After installing the anti-malware code, the malware is removed and the configuration settings are restored. If no anti-malware is found, then further changes to configuration settings are disabled and the corresponding SMS message is locked. The scope of attacks considered in this case study is a mixture of cyber security and social engineering attacks. Figs. 5 and 6 show the corresponding statecharts using the extended and original notations, respectively.

4.4 Case Study Evaluation

As shown in this case study, there is only one normal state while the remaining 10 states relate to various security aspects. Using the original notation, the other types of the

other 10 states can only be identified using textual stereotypes; a generic extension mechanism used throughout the entire suite of UML diagrams. The original notation offers to visual mechanism to specify the different types of security related states. It can also be shown that there are only two regular events in comparison to 15 various security related events. As shown in Fig. 6, security related events can only be modeled as regular events. A modeler will have to be careful in choosing the event labels in order to signify that they security related. Moreover, there is one security related start node and one security related final node. As shown in Fig. 6, a modeler cannot specify these specific types of semantics. A modeler is forced to resort to the normal start and final nodes.

It is certainly important to specify formal semantics before the new notation comes into full operation. As it stands, the new notation simply inherits the formal semantics of the original notation. This means that the users of the new notation have the flexibility to use the new notation as they find suitable while adhering only to the base semantics of UML statechart. The specification of formal semantics however is considered part of the model construction aspect of the new notation. This paper focuses on the model comprehension aspects of the new notation. Whilst model construction is equally as important, ensuring effective model
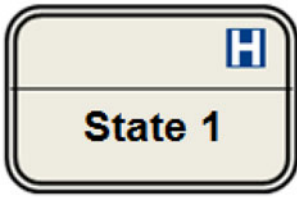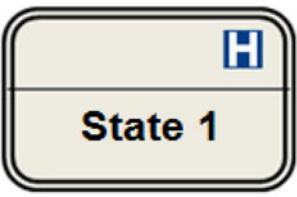
Page n	
State 1 (H)	**What does this symbol mean?** o The system is vulnerable to an attack. o The system is currently being attacked. o The system is quarantined to avoid further attacks and damage. o The system is recovering and fixing damage. o The system is performing defensive actions that will later allow fixing damage, or stopping further or new attacks. o This is an ordinary business-related state. o Other
Please provide the reasons for your choice:	

Page n+1		
State 1 (H) ☐	☐	**Which of these two symbols better indicates that the current state is a recovery state, i.e. a state where the system is fixing damages it has sustained?** (Place a check mark adjacent to the symbol of your choice)
Please provide the reasons for your choice:		

Fig. 7. An excerpt of the questionnaire used.

comprehension is a prelude to this next phase of research. This next phase of research is a major undertaking that will require separate empirical validations and will be the subject of future research.

5   DO THE NEW SYMBOLS PROVIDE A CLEARER INDICATION OF THEIR SEMANTIC MEANING? AN INDUSTRIAL SURVEY

In this section we validate whether the design of the new notation symbols actually provide a clearer indication of their semantic meaning in comparison to the original notation, which is a matter of great subjectivity. Naturally, a designer of a notation may be biased towards a favorable evaluation of the notation. The following research questions were specifically addressed by this study:

RQ1: Is the design of the notation indeed semantically transparent?
RQ2: Is the design of the notation more semantically transparent than the original notation?

To this end, we created an online survey to collect the independent opinions of professional security analysts, which can be accessed from this research's companion website [18]. The survey method is recommended when self-reported data from a large number of participants is to be elicited [57].

5.1   Survey Design

A questionnaire was developed to conduct the survey. The questionnaire evaluates the meaning of each new symbol in isolation first, and then it evaluates each new symbol against its counter symbol from the original notation for comparison purposes. Qualitative data about each symbol is presented to request the survey participant to provide an explanation for their answers to both parts of a symbol evaluation. To ensure that visual aspects of the new notation are the reason behind a participant's choice, the new symbols are handicapped by removing their corresponding textual stereotypes. The following is an excerpt from the questionnaire that shows a two-part question about the *recovery state* symbol (see Fig. 7).

In the actual questionnaire, the two parts of a question pertaining to a particular symbol are presented on two webpages. The rationale behind this is to avoid providing the participant with a clue about the correct meaning of the symbol (first part), which is shown in the second part of the question. To mitigate against learning effects, the order of the symbols to be evaluated is randomized amongst participants. This was achieved by adding a randomization function in the PHP (Hypertext Preprocessor) script that links to the survey. Initially, a pilot study was conducted with graduate students to validate the questionnaire. As a result of the pilot, study the questionnaire was subject to minor improvements, particularly with respect the wording of some questions.

The questionnaire begins with eliciting information about the participant followed by a small tutorial of the basic notation elements of statecharts, namely the concepts of a state and an event, and the semantic definition of states and events. Only professional security analysts were invited

TABLE 4
Survey Responses

Notation	Expected the intended meaning	Felt it was more suggestive of the underlying security aspect than the original notation as stated by the UML
Threatened state	86.2%	100%
Vulnerable state	87.9%	94.8%
Defensive state	96.5%	100%
Compromised state	79.3%	100%
Quarantined state	82.7%	100%
Recovery state	96.5%	96.5%
Threat event	100%	100%
Countermeasure event	100%	100%
Initial threat node	100%	94.8%
Final compromised node	100%	93.1%

to participate in the survey. It was considered important to only invite professional security analysts to partake as they would have a greater knowledge and deeper understanding of security related aspects. Such professionals are believed to be better judges between the new and original notation. They are also more likely to determine if there is important security aspects that were overlooked in the new notation. Most respondents were recruited via the LinkedIn website [44], a social-media website for professional. There were also some who we reached via direct referrals.

5.2 Survey Results

In total we received 68 responses; however 10 responses were excluded as they were deemed invalid mainly because the respondent was not actually a security analyst professional. All of the respondents indicated they were familiar (with varying degrees) with the concept of finite state machines and some were also familiar with the UML statecharts diagram notations. The data from the remaining 58 respondents were analyzed and the results are shown in Table 4. Table 4 shows the ratio of correct answers for first part of each question, as well as the ratio of respondents who felt that the new notation was more expressive and suggestive of its corresponding security aspect than the original notation. Familiarity with statecharts was expected to create bias towards the original notation, however the results showed that despite their familiarity, the respondents still preferred the new notation.

With respect to guessing the correct meaning of a symbol, all respondents answered correctly for the initial and final node notations as well as all event notations. Almost all respondents answered correctly for the recovery state symbol as well. The remaining four symbols were not as intuitive as the other symbols; however the majority of the respondents still answered the corresponding questions correctly. The compromised state symbol was the least intuitive of all symbols, yet 79.3 percent respondents were able to guess the intended meaning.

With respect to comparing the new notation with the original notation; the new notation scored much better than the original notation, despite being rid of textual stereotypes. The new notation was preferred 100 percent of the time for six symbols. For the remaining symbols,

the new notation was preferred by at least 93.1 percent respondents.

The survey results show that the respondents find the new notation more descriptive than the original notation. The survey results also show that new symbols were highly suggestive of their underlying security semantics. For the compromised and quarantine state symbols, they did not score overwhelmingly high as other symbols did; this however is not a serious deterrent. After all, users of the new notation are expected to undergo some training of the new notation before actually using it, similar to when they were trained to use the original notation of statecharts. Given the survey results, it can be argued that users can get to firmly grasp the concepts of the new notation without extensive training.

Qualitative information from the survey attests the quantitative data. The following is a categorization of the answers provided:

Drawbacks of the new notation. The respondents mainly indicated that the quarantine and compromised states were easy to confuse with each other. In fact, further analysis has shown that the majority of incorrect answers for the quarantine state is when the respondent thought it was the compromised state, and vice versa. Therefore, the score for the quarantine and compromised states were similar. The same situation occurred with the threatened and vulnerable states, where respondents confused one for the other and hence the scores for these two symbols were very similar.

Coverage of security related semantics. One of the most important findings from the qualitative data is that all security analysts felt that the modeling language encompasses the most important security aspects. Other security aspects indicated by the respondents as not covered by the new notation were indeed a sub-type of one of the security aspects already covered. Recall that it may be counterproductive to create a symbol for every aspect as that will lead to too many symbols being created, hence actually reducing cognitive effectiveness and increasing the difficulty of learning and use.

Call for tool support. Many security analysts indicated that they would definitely prefer using the new notation but only if tool support is available. State, node and arrow coloring, and creating custom stereotypes, are features that are available in just about all major UML modeling tools.

5.3 Threats to Validity

This section presents the threats to the validity of the study in accordance with the standard classification [79].

5.3.1 Conclusion Validity

Like other researchers of many studies involving experience or opinion data [3], [28], [38], [53], we also have full confidence in our results, as we have elicited data from experts working in diverse cultures and organizations. We also have no reason to believe that a respondent would intentionally provide arbitrary answers to sabotage the results since participation is absolutely voluntary. Finally, we have confidence that the respondents are indeed professional security analysts as they indicated in the survey. This confidence is based on the lack of reason to partake in the survey by intentionally falsifying their actual profession. This confidence is also based on the fact that respondents were required to provide their professional contact information, which can be used to verify their professions. The 10 respondents who were excluded from the analysis most likely participated in the survey because they were somehow unaware of the survey's profession restriction.

5.3.2 Internal Validity

Internal validity provides confidence in the overall assessment of the results. The questionnaire was developed in consultation with IT security experts. A pilot study was conducted to validate the questionnaire and its results provide an acceptable level of validity.

5.3.3 Construct Validity

A common problem in questionnaire-based surveys is that they consist of closed-ended questions. Closed-ended questions tend to pre-empt a particular solution from respondents. To mitigate this threat, the ordering of solutions is randomized throughout all questions in the questionnaire. Each question is designed to provide an extra choice in which a respondent can indicate that they simply do not know (or have no basis for judgment), or they can indicate that there is no difference between a particular aspect of a particular new symbol and its original counterpart. The last question in each set of questions pertaining to a particular symbol is an open-ended one. The purpose of the open-ended question is to elicit qualitative information from respondents that validates their choices. Validation in this sense refers to a respondent making the correct answer selections based on their actual intentions.

5.3.4 External Validity

External validity is concerned with the generalization of results to contexts and situations other than the one in which original study was conducted. The survey results are based on the opinions of 58 security experts from 18 different countries spanning all continents. Although it cannot be assumed that the results obtained from this survey represent the opinions all security experts across the globe, we do believe that they provide a reasonable representative sample.

6 BUT CAN THE NEW NOTATION BE READ QUICKER? A SUBJECT-BASED EMPIRICAL EVALUATION

In this section we report a controlled subject-based experiment that was used to validate the cognitive effectiveness of the new notation. In particular, this experiment investigates if statecharts using the new notation can be read quicker and more accurately than graphs that use the original notation. The experiment involved software professionals from Saudi Arabia. The experiment follows the experimentation process proposed by Wohlin et al. [79]. The following sections describe the experiment's definition, context, hypothesis formulation, subject selection, design, instrumentation and measurement techniques, and validity evaluation, respectively.

6.1 Experiment Definition

The main research question posed by this experiment is whether the proposed notation can be used to develop statecharts that can be syntactically read more effectively than statecharts built using the original notation. Syntax readability in this sense refers to the speed and accuracy of which subjects can read the graphs. A better notation is one that can be read quicker and one that will lead its readers to fewer cases of misreading. The experiment is defined to have one independent variable, which is the use of the proposed notation. If the changes applied by the new notation are eliminated then what will be left are diagrams that use the original notation. Hence two treatments exist, the use of the extended notation and the use of the original notation. To assess the effect of using the EN, two dependent variables were recorded: the response time for the subjects to answer questions related to one diagram type (T), and the errors committed when answering questions related to one diagram type (E).

6.2 Experiment Context

The experiment involved software engineering professionals from various companies. The experiment was conducted over many sessions in order to accommodate the busy and varying schedules of the professionals and to gather a sufficient number of subjects as required to perform meaningful statistical analysis. The experiment was conducted as a voluntary exercise without financial compensation. Instead, the subjects received educational value for participating. The experiment was divided into two components. The first component was a tutorial about the involved notations to familiarize the subjects with them and their underlying semantics. The tutorial provided the subjects with all the necessary background about statecharts in order to perform the prescribed experimental tasks. The second component of the exercise consisted of an unlimited time session in which the subjects undertook the assigned experimental tasks. The subjects were never informed about the hypotheses under investigation to prevent the threat of bias towards either of the two notations.

6.3 Hypotheses Formulation

Two hypotheses were produced to account for the potential effects of using the extended notation. For the response time

TABLE 5
The Dependent Variables and Their Corresponding Hypotheses

Dependent Variable	Null Hypothesis (Ho):	Alternative Hypothesis (Ha):
Response Time	(Ho1): T (EN) $\geq$ T (ON)	(Ha1): T (EN) $<$ T (ON)
Errors	(Ho2): E (EN) $\geq$ E (ON)	(Ha2): E (EN) $<$ E (ON)

variable (*T*), the alternative hypothesis indicates that subjects reading statecharts built with the extended notation will be able to answer questions about them faster than subjects reading statecharts built with the original notation. For the errors committed variable (*E*), the alternative hypothesis indicates that subjects will make fewer incorrect answers when they are concerned with statecharts using the extended notation. Both variables were set as one-tailed hypotheses. A one-tail test allots the alpha to testing the statistical significance to one direction while completely disregarding the possibility of a relationship in the other direction. This direction of the hypotheses were set as one-tailed as the new notation was specifically designed to be an improvement of the original, hence a better result is expected in the experiment. The dependent variables and their corresponding hypotheses are shown in Table 5.

6.4 Subject Selection

The professionals who participated in this experiment worked in different organizations across the eastern province of Saudi Arabia. All of them had less than 3 years of industrial experience with statecharts. It should be noted that having less than 3 years of experience was not a selection criterion, coincidently all the subjects that volunteered were under this category. They were randomly divided into two groups (A and B). In total there were 39 subjects.

6.5 Experimental Statecharts and Tasks

A 2 × 2 partial factorial design with repeated measure was used to mitigate the effect of individual and group abilities. Accordingly, all subjects were required to consider two distinct state machines; one that uses the new notation while the other uses the original notation (see Table 6).

There exist two types of graphs: *abstract* and *domain-specific* graphs. *Abstract* graphs contain no information that references the real world (i.e., the underlying domain). *Abstract* graphs are commonly referred to as *syntactic* graphs. *Domain-specific* graphs contain information about the underlying domain. *Domain-specific* graphs are commonly referred to as *semantic* graphs. For this experiment, we are evaluating how fast subjects can obtain syntactical information from the graph. The underlying domain is therefore irrelevant. Hence,

the type of graphs used in this experiment is only syntactical graphs. Prior experimental research in notation understandability demonstrated the effect of diagrammatic layout features on comprehension [58]. Therefore, each pair of structurally identical statecharts (one using the new notation, while the other using original notation) were presented using identical layouts in order to eliminate any effects caused by different diagrammatic layouts. Besides the type and layout of the statecharts, the statecharts used in this experiment were required to satisfy the following criteria:

1. The statecharts used must include the entire notational set of the new notation.
2. The statecharts used must be of similar complexity. Similar syntactical complexity is achieved by using statecharts that have a similar number of nodes and edges in both statecharts. Similar syntactical complexity is also achieved by using statecharts that contain similar numbers of each type of nodes and edges used. This criterion is important to prevent the relative complexity levels of the statecharts used from influencing the results.
3. The statecharts used must be of considerable size and complexity in order to draw meaningful conclusions. Certainly, the bigger the artifacts the better. Bigger artifacts provide greater confidence in any statistically significant (or insignificant) result. But how big is big enough? To answer this question we referred to other empirical evaluations conducted that evaluate notation readability and understandability by subjects, such as [58], [59], [24], [60]. In this experiment, we used artifacts that were roughly twice the size of the artifacts used in [58], [59], [24], [60] in terms of the number of nodes and edges.
4. The statecharts used must be developed by authors independent of this research work. This requirement is important to eliminate bias towards the new notation since it is proposed by the authors of this paper. The graphs used were developed by two graduate student groups as part of their course project work. Some minor modifications were made to the statecharts in order to satisfy criterion (2). The impact of the minor modifications is believed to be negligible

TABLE 6
Experimental Design

	Group A	Group B
Session 1	EN Statechart Diagram 1	ON Statechart Diagram 1
Session 2	ON Statechart Diagram 2	EN Statechart Diagram 2

with respect to biasing subjects towards any particular notation.

Satisfaction of the aforementioned four criterion can only be validated upon completing the experiment and analyzing the results statistically.

The subjects were given one statechart at a time and were required to answer questions presented as a questionnaire. Both sets of questionnaires contained an identical set of system-independent questions. The questions only require identifying the syntactical properties of the symbols. For example, "Identify the threatened states". The order of these questions was randomized in both questionnaires to mitigate learning effects. Since the questionnaires contained the same set of questions, the questionnaires have the same maximum score. Because we used abstract graphs from which the domain-specific information has been removed, it will be impossible for subjects to identify any security constructs conveyed by the symbols. Therefore, some advantages were handed to the original notation. First, the textual stereotypes of states in the new notation are not removed. Second, for threatening events, they were annotated with the textual tag (t) and depicted on the event arrows. Finally, for the "initial threat" and "final compromised" nodes, they were also annotated with the textual tag (t) and depicted as part of the node. For "countermeasure" events, they were annotated with the textual tag (d). Due to the large size of the statecharts used in the experiment, the statecharts are shown in Appendix A, available in the online supplemental material.

The subjects were given handouts containing the legends for both notations printed in color. Providing the legends as handouts allows them to be displayed in front of subjects at all times. This reduces the cognitive load needed to consciously maintain meanings of symbols in working memory and avoid having to flip back and forth between different pages.

The time required to answer a set of questions related to one particular statechart was recorded for each subject (see details in the next section). The subjects were requested to complete the experimental tasks within one session. This means that the subjects were not allowed to take the experimental artifacts away and return them later. However, there was no time limit for the session so that the subjects will not face any timing pressures. A post-experiment questionnaire included a question that specifically asked if the subject felt any timing constraints. The responses of all subjects to that question confirm that they did not feel any timing pressures. Regardless of the number of subjects involved in a session, the experiment conductors were present at all times. The presence of the conductor helps prevent the threat of subjects completing their tasks but forgetting to stop the timer. The presence of the conductor is useful for answering questions that subjects may have during the experiment.

6.6 Instrumentation

All the experimental artifacts were printed in color on paper. Black and white models were also printed using the same size paper. The subjects were requested to answer questions on paper using pencils, which were provided to all subjects. Subjects also used stopwatches available in their own mobile phones to keep time.

6.6.1 Why Not Conduct the Experiment Online?

Certain limitations prevented us from conducting the experiment online whereby a larger set of subjects can participate. One reason was to ensure that subjects who are involved in this experiment did not participate in the survey presented in Section 5, in order to eliminate learning effects or bias that may have developed from participating in the survey. Another reason is to ensure that each subject completed their prescribed tasks individually without the help of other individuals. Another reason is the uncontrolled effect of the size of the display devices used by subjects. Some subjects may use large screen monitors while others may use a small handheld device. The size of the display device will surely have an effect on the speed of browsing and reading of the diagrams. Moreover, the physical presence of the conductors prompts subjects from losing their focus on the experimental tasks. Finally, the availability of the experiment conductors allows them to answer any on-the-spot inquiries by the subjects.

6.7 Analysis Procedure

Quantitative data related to errors was considered as discrete count data since there was no evidence that would lead us to believe that the questions had unequal unit weighting. Quantitative data related to response time was also considered as discrete data. The data sets were examined for their compliance to the normality assumptions using the Shapiro-Wilk test [65]. An advantage of this test in comparison to other common "normality" tests is that it tends to be more powerful and it does not require the mean or variance of the hypothesized normal distribution to be specified in advance. This feature of the Shapiro-Wilk test is required since there was no causal explanation as to the nature of the distribution that the data points are sampled from. Data sets were first compared to detect any linear correlation between the two dependent variables. They were then compared to detect if there are any statistical significant results.

6.8 Scoring and Measurement

The speed of each subject's responses was measured using stopwatches held by each subject. We also measured the number of errors made by each subject by scoring the answered questionnaires. Scoring the questionnaires is not a subjective process as all the questions had definitive correct answers. Therefore, there was no need to elicit independent researchers to perform unbiased evaluations. Scoring of the questionnaires was performed by the research team. The research team recorded qualitative data on a spreadsheet to determine any threats to validity and to gain any useful insights by the subjects. In particular, the subjects were asked which notation they found clearer and more suggestive of its underlying semantics. The qualitative data was also elicited about the subjects' general preference between the two notations. The subjects were also asked if the new notation has sufficiently covered the major security related semantics. Although this question was previously answered by professional security analysts in the industrial survey (see Section 5), and although the subjects are not specifically professional security analysts, it is possible that they may suggest an important security semantics that was

TABLE 7
Correlation Results

Time versus Errors	r statistic	n	t statistic	95% C.I.	p
Extended Notation	−0.03	39	−0.18	−0.34 to 0.29	0.8583
Original Notation	−0.28	39	−1.79	−0.55 to 0.047	0.0816
Overall	0.03	78	0.22	−0.20 o 0.25	0.8252

otherwise missed (by professional security analysts) in the results of the survey. Other qualitative data elicited pertained to timing pressures. Timing pressures may have affected the results of the experiment.

6.9 Analysis and Interpretation

In this section we present analyses of the data collected followed by a discussion of the findings. For each analysis performed we present a descriptive summary for each nonparametric variable. The correlations between time and errors were calculated, accumulated according to the groups, statecharts and notation types. Detecting linear correlations between errors and response time was necessary to determine whether there was any relationship between the two dependent variables that would make it inappropriate to analyze them separately. Statistical significance was also set at the standard 95 percent level. In case the variables are indeed independent, the Mann-Whitney U statistic was used to test for differences between the medians of related samples, which was calculated as prescribed in [67]. The Hodges-Lehman method [42] was used to compute the confidence intervals, given at the standard 95 percent level, around the difference between medians.

6.9.1 Performed Analysis

The analyses performed investigate the effects of the treatment variables and experimental artifacts in isolation. An outline of the analyses performed in the following sections

is shown below. The rationale behind performing each analysis is presented in its corresponding section.

- Section 6.9.1.1: Correlation analysis (Time versus Errors)
- Section 6.9.1.2: Extended Notation versus Original Notation
- Section 6.9.1.3: Extended Notation versus Original Notation (Aggregated Results)
- Section 6.9.1.4: Statechart diagram 1 versus Statechart diagram 2
- Section 6.9.1.5: Group A versus Group B

Correlation analysis—time versus errors. In this section we investigate if there is a correlation between the time and errors variables. The correlations between time and errors were calculated, accumulated according to each notation type individually and according to all performances in general (i.e., new and old notations). Correlation was calculated using the Pearson correlation coefficient [61] ($r \neq 0$) at the 0.05 level of significance. The Pearson correlation coefficient test is used as it is more appropriate for measurements taken from interval scales. Correlation results are shown in Table 7. As shown in Table 7, no linear correlation was detected between the two variables. This indicates that the subjects were not affected by time in order to influence the correctness of their answers. Similarly, the subjects were not influenced by their goal to answer questions correctly to reduce time. Accordingly, a statistically insignificant result for one variable does not influence the hypothesis evaluation of the other variable. Fig. 8 shows the scatter plot for all subject performances.
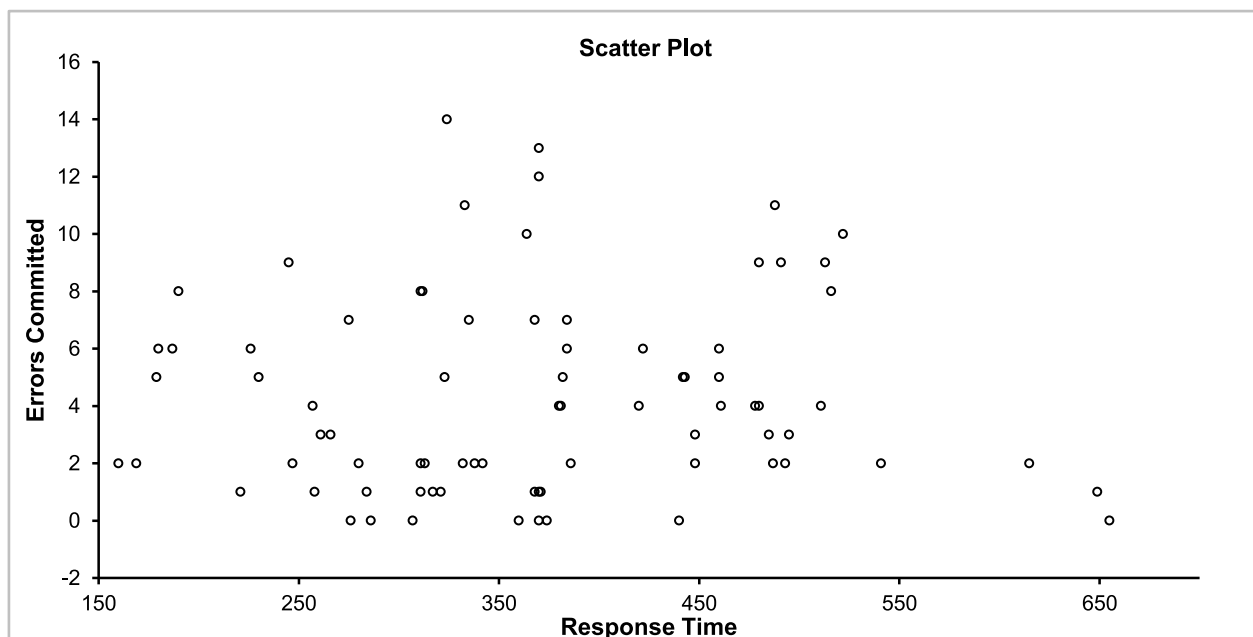

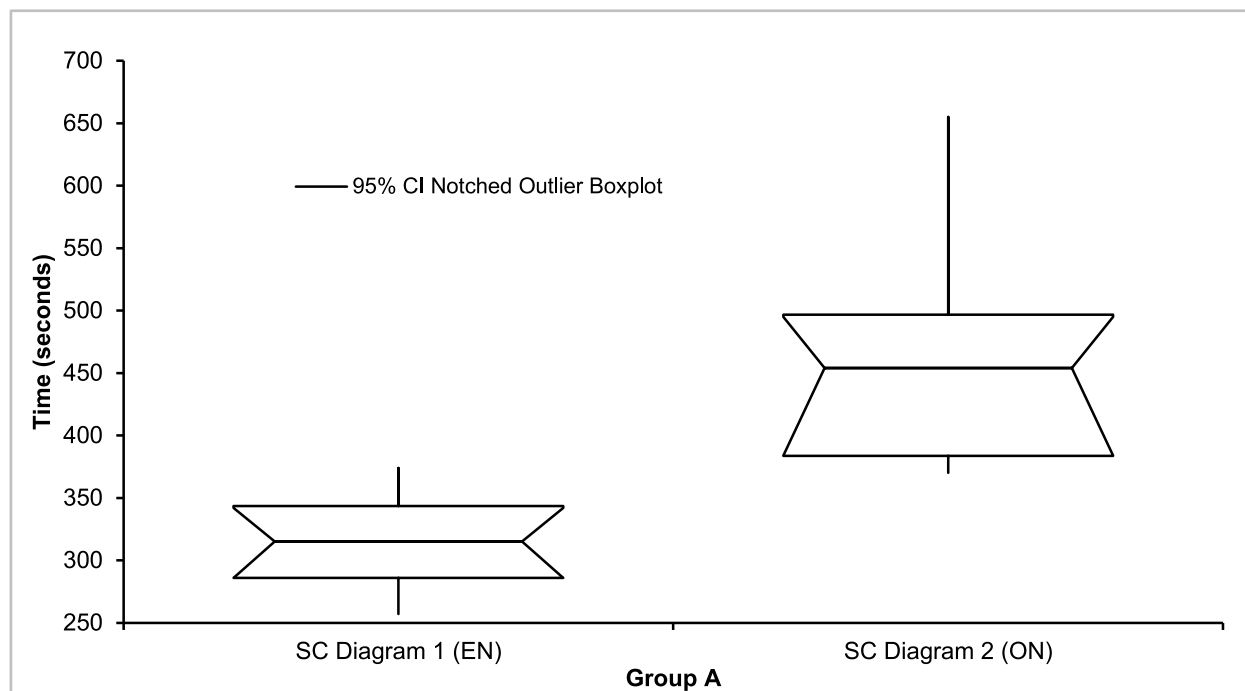Fig. 8. Scatter plot for all performances showing time versus Errors.

Fig. 9. Group A's performance with respect to response time and each statechart in isolation.

Extended notation versus Original notation. In this section we investigate the effect of using the extended notation versus the original notation on each statechart in isolation with respect to the two dependent variables, using both groups in isolation and for each statechart separately. Figs. 9 and 10 show the results for the response times and errors committed variables for group A. Table 8 presents descriptive statistics for group A's performance. Figs. 11 and 12 show the results for the response times and errors committed variables for group B. Table 10

presents descriptive statistics for group B's performance. Tables 9 and 11 show the results of the Mann-Whitney test for both dependent variables for groups A and B, respectively.

As shown in Tables 9 and 11, statistical significance was observed in performance of both groups with respect to the time variable. This means that use of the EN needed significantly less time to read than the ON. However, statistical significance was only observed in performance of group A with respect to the errors committed variable.

Fig. 10. Group A's performance with respect to errors committed and each statechart in isolation.

TABLE 8
Descriptive Statistics for the Group A (n = 18, EN = Extended Notation, ON = Original Notation)

Variable	Notation	Min	1st Quartile	Median	3rd Quartile	Max	IQR
Response Times	EN	257	285.8	315.0	343.5	374	57.7
	ON	370	383.8	454.0	496.8	655	113.0
Errors Committed	EN	0	0.0	1.0	2.0	4	2.0
	ON	0	1.0	2.0	4.1	8	3.1

Fig. 11. Group B's performance with respect to time and each statechart in isolation.

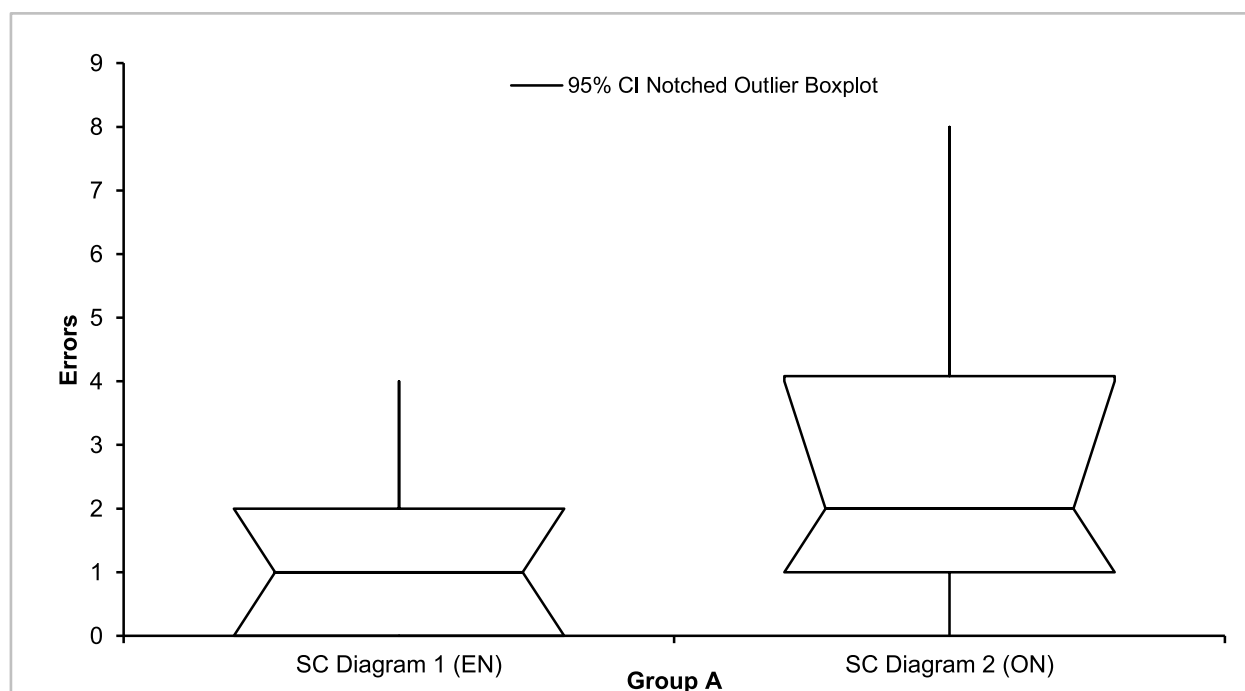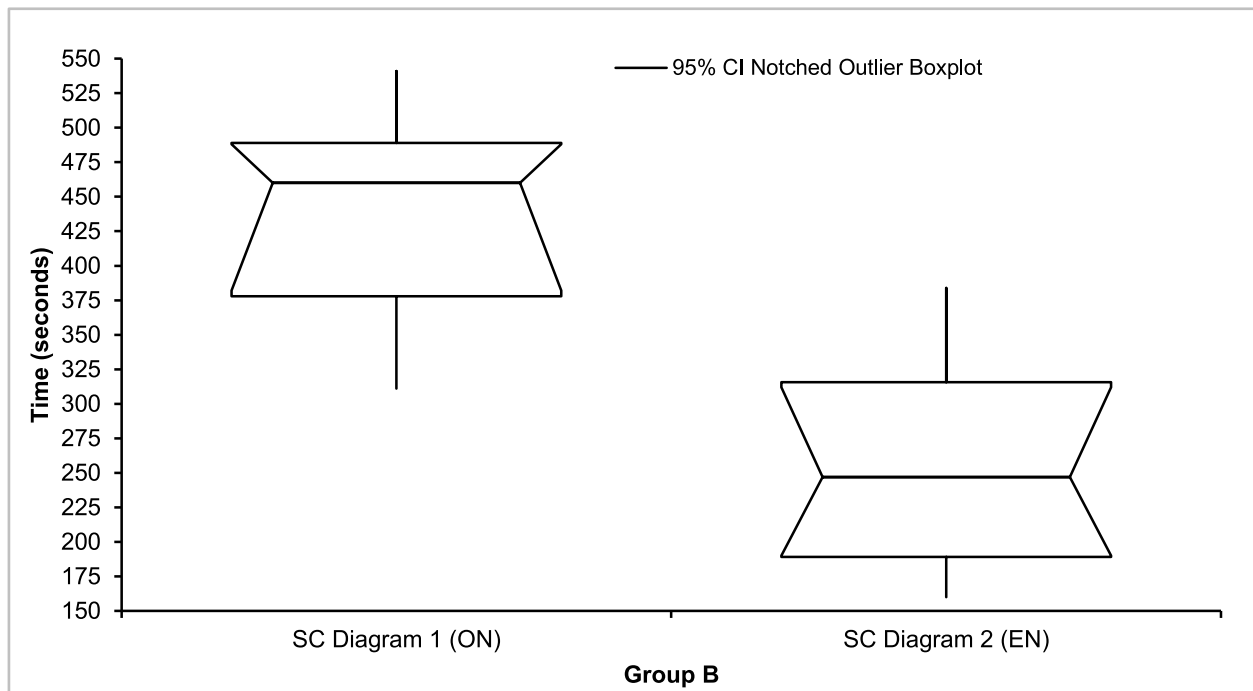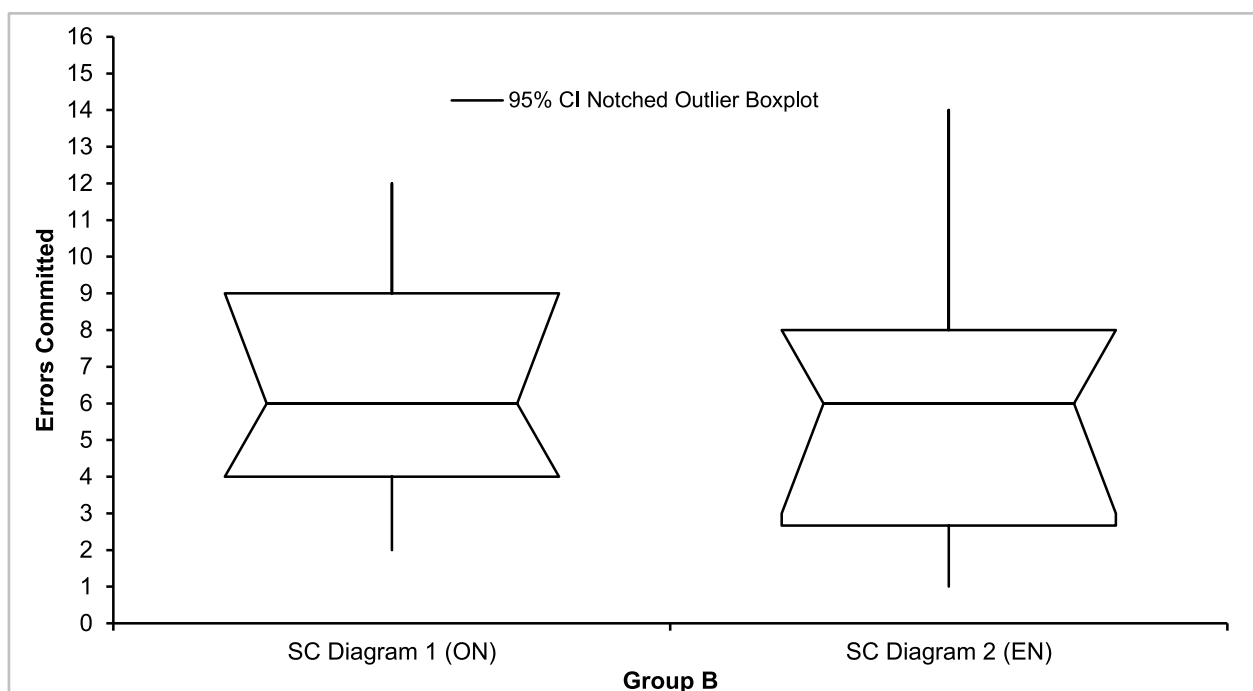Fig. 12. Group B's performance with respect to errors committed and each statechart in isolation.

TABLE 9
Mann-Whitney Test for the Group A (n = 18, EN = Extended Notation, ON = Original Notation)

Variable	Notation	Rank sum	Mean rank	U	Median difference	95% CI	Mann-Whitney U statistic	p
Response	EN	175.0	9.72	320	−132.5	−∞ to −102.0	320	<0.0001
Times	ON	491.0	27.28	4.0				
Errors	EN	256.5	14.25	238.5	−1.0	−∞ to 1.0	238.5	0.0065
Committed	ON	409.5	22.75	85.5				

TABLE 10
Descriptive Statistics for the Group B (n = 21, EN = Extended Notation, ON = Original Notation)

Variable	Notation	Min	1st Quartile	Median	3rd Quartile	Max	IQR
Response	EN	311	378.0	460.0	489.0	541	111.0
Times	ON	160	189.0	247.0	315.7	384	126.7
Errors	EN	2	4.0	6.0	9.0	12	5.0
Committed	ON	1	2.7	6.0	8.0	14	5.3

TABLE 11
Mann-Whitney Test for Group B (n = 21, EN = Extended Notation, ON = Original Notation)

Variable	Notation	Rank sum	Mean rank	U	Median difference	95% CI	Mann-Whitney U statistic	p
Response	EN	247.5	11.79	424.5	−190.0	−∞ to −149.0	424.5	<0.0001
Times	ON	655.5	31.21	16.5				
Errors	EN	417.0	19.86	255.0	−1.0	−∞ to 1.0	255.0	0.1917
Committed	ON	486.0	23.14	186.0				

Extended notation versus original notation (aggregated results). In this section we investigate the effect of using the extended notation versus the original notation once again but while using aggregated results from both statecharts. This analysis explores the experimental results to provide some illustrative experiment-wide numerical statements about the study. The results provide a more general assessment of the overall impact of using the different notations in comparison to the analysis shown in Section 6.9.1.2. This analysis is especially interesting for the errors variable since previous analysis has shown that statistical significance was observed in the performance of only one group. Hence, performing analysis with the aggregated results will provide stronger confidence to reject the hypothesis
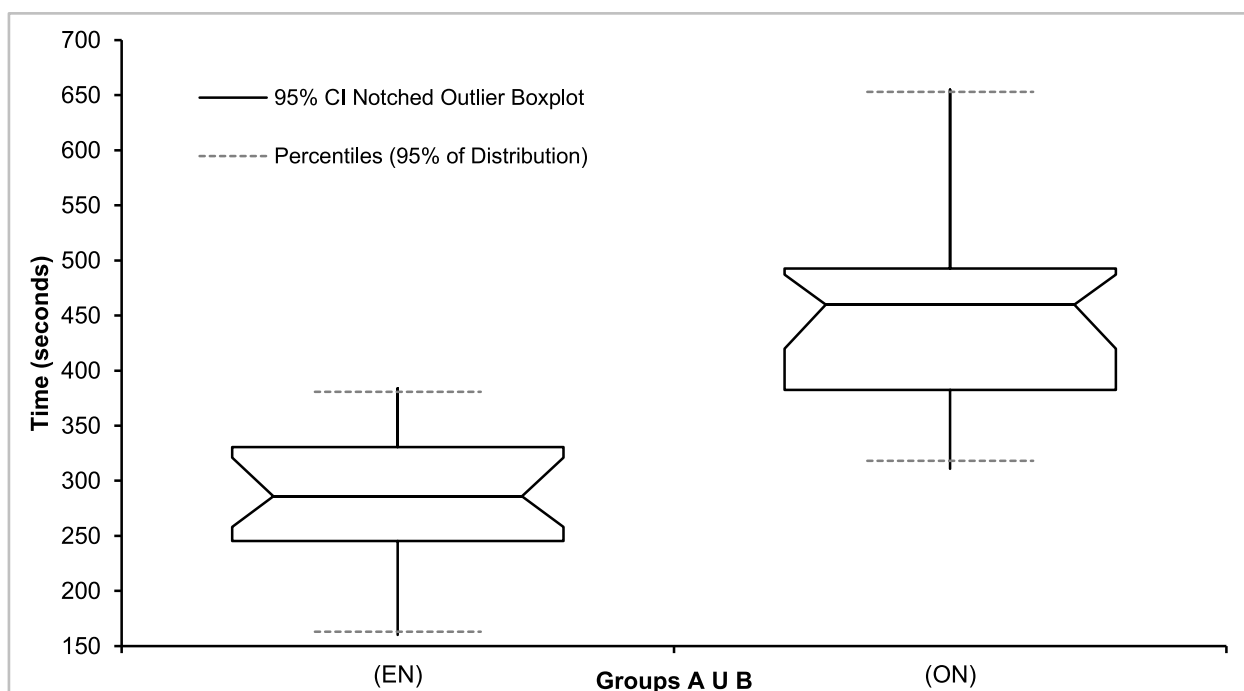

Fig. 13. The cumulative performance of both groups with respect to time for both statecharts.
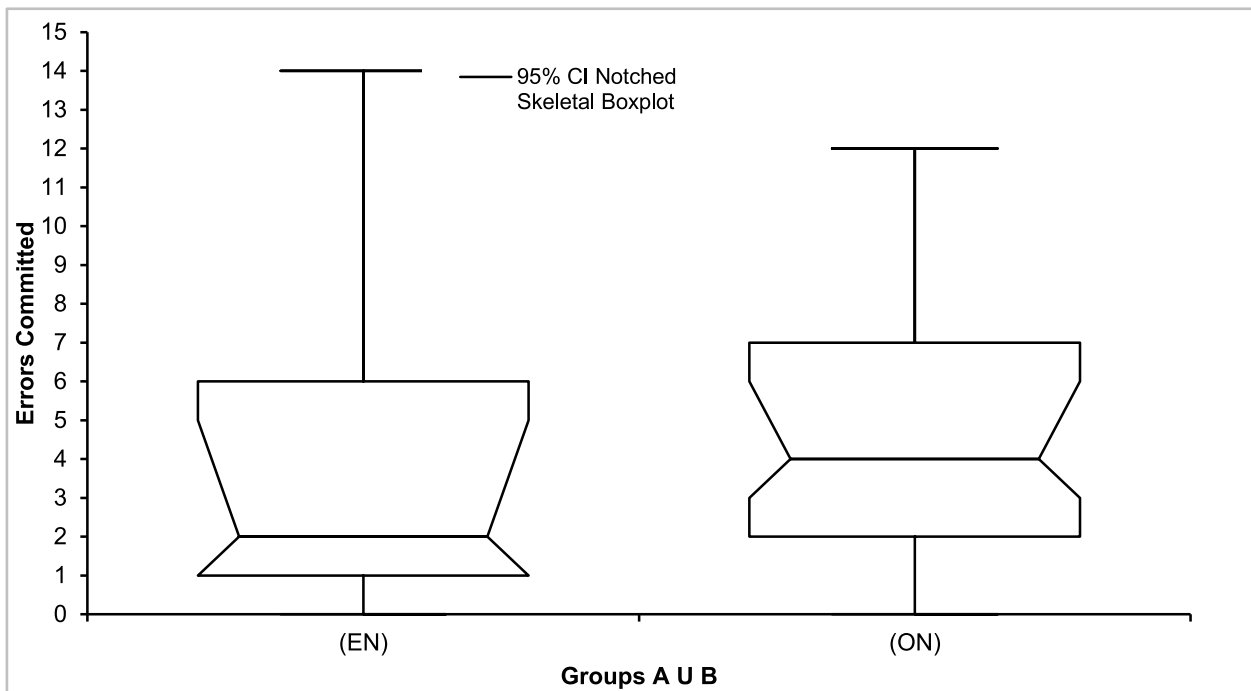
Fig. 14. The cumulative performance of both groups with respect to errors committed for both statecharts.

regarding correctness. Figs. 13 and 14 show the aggregated results for the response times and errors for both statecharts and groups, respectively. Table 12 presents descriptive statistics while Table 13 shows the results of the Mann-Whitney test for both dependent variables for both statecharts.

Table 13 further concurs with the findings of the analysis performed in the previous section whereby statistical significance was observed in the aggregated results for the response times variable. However, statistical significance was also observed for the errors committed. We will however take a conservative approach and reject the hypothesis related to the errors committed variable. Recall that no correlation was detected between the response times and errors committed variables (see Section 6.9.1.1), therefore even if the hypothesis relating to the errors committed variable was rejected, this does not result in rejecting the hypothesis regarding the time

variable as well. However, given the rejection of the errors committed variable hypothesis in our previous analyses, subsequent analysis performed in Section 6.9.1 will no longer consider the errors committed variable.

Statechart Diagram 1 versus statechart Diagram 2. In this section we investigate differences between the two statecharts. The investigation is conducted using aggregated results from both statecharts regardless of the applied treatment. It is important to conduct this investigation to ensure that the complexity levels of the used statecharts indeed did not affect the results of the experiment. Fig. 15 shows the results of the subjects' performances for both statecharts with respect to the response times variable. A statistical significance between the two statecharts will indicate that the complexity levels of the statecharts had affected the results obtained. Table 14 presents descriptive statistics while

TABLE 12
Descriptive Statistics for the Both Groups (n = 39, EN = Extended Notation, ON = Original Notation)

Variable	Notation	Min	1st Quartile	Median	3rd Quartile	Max	IQR
Response	EN	160	245.3	286.0	330.7	384	85.3
Times	ON	311	382.3	460.0	492.7	655	110.3
Errors	EN	0	1.0	2.0	6.0	14	5.0
Committed	ON	0	2.0	4.0	7.0	12	5.0

TABLE 13
Mann-Whitney Test for the Aggregated Performance Results with Both Groups and Statecharts (n = 39, EN = Extended Notation, ON = Original Notation)

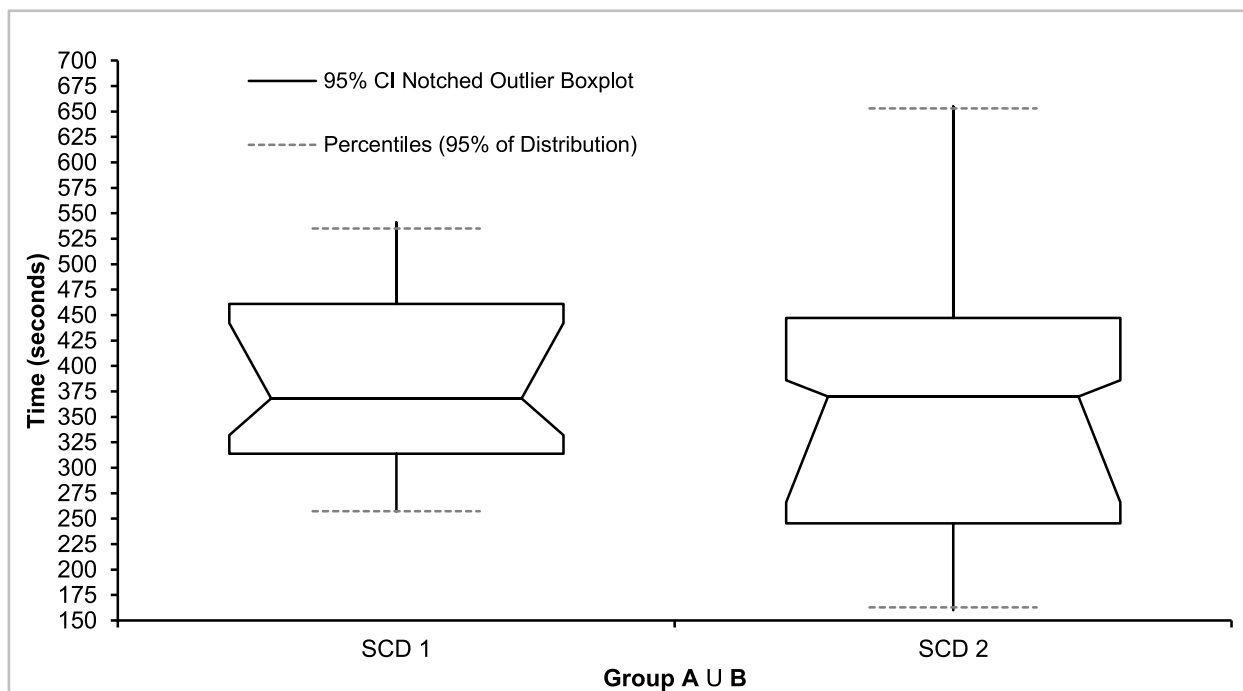| Variable	Notation	Rank sum	Mean rank	U	Median difference	95% CI	Mann-Whitney U statistic	p
Response	EN	833.5	21.37	1467.5	−164.0	−∞ to −136.0	1467.5	<0.0001
Times	ON	2247.5	57.63	53.5				
Errors	EN	1349.5	34.60	951.5	−1.0	−∞ to 0.0	951.5	0.0273
Committed	ON	1731.5	44.40	569.5				

Fig. 15. The cumulative performance of subjects with respect to errors committed for each statechart in isolation.

Table 15 shows the results of the Mann-Whitney test for the response times variables.

As shown in Table 15, no statistical significance was observed between the used statecharts thus eliminating the relative complexity of the statecharts as a factor towards the statistical significances observed in the previous sections.

Group A versus Group B. In this section we investigate the performance of Group A versus Group B with respect to each of the two dependent variables, using aggregated results from both statecharts. It is important to conduct this investigation to ensure that the relative ability levels of both groups did not affect the results of the experiment. Fig. 16 shows the results of the subjects' performances from both groups with respect to the response times variable. A statistical significance between the two groups will indicate that the ability levels of the groups had affected the results obtained. Table 16 presents descriptive statistics while Table 17 shows the results of the Mann-Whitney test for both dependent variables for Groups A and B.

Once again no statistical significance was observed between the performances of both groups. This finding eliminates the relative ability levels of the groups as a factor towards the statistical significances observed in previous analyses.

6.9.2 Discussion of the Results

The statistical results obtained from the experiment has shown that the new notation can significantly increase the speed by which statecharts rich with security aspects are read by its users. The delta of reading times between the two notations is only expected to increase as the size of the statecharts increase. The results obtained with respect to reading accuracy do not allow us to safely accept the corresponding hypothesis. With only one statistically significant result, it can be argued that if the statecharts used in the experiment were larger then statistically significant results may have been observed.

Ostensibly it may seem that the subjects have used more time to ensure that they commit the fewest possible

TABLE 14
Descriptive Statistics for Statechart Diagram 1 versus Statechart Diagram 2 ($n = 39$)

Variable	Diagram	Min	1st Quartile	Median	3rd Quartile	Max	IQR
Response	Statechart 1	160	245.3	286.0	330.7	384	85.3
Times	Statechart 2	311	382.3	460.0	492.7	655	110.3

TABLE 15
Mann-Whitney Test for Statechart Diagram 1 versus Statechart Diagram 2 ($n = 39$)

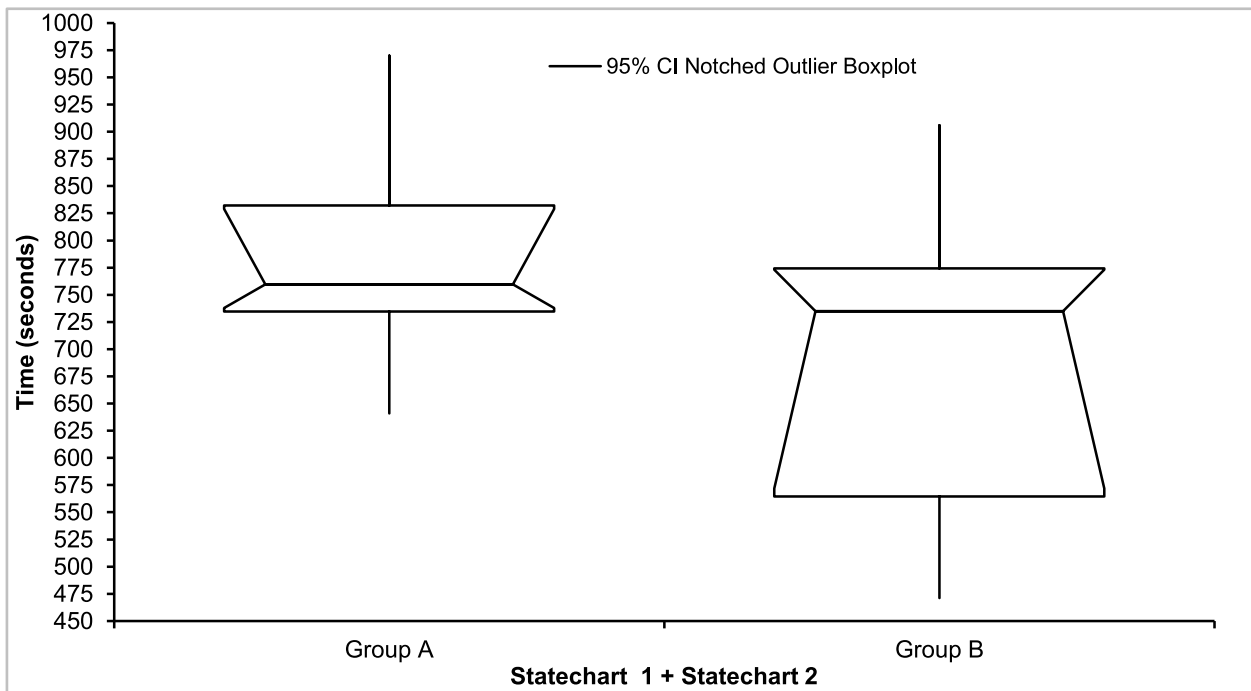| Variable	Diagram	Rank sum	Mean rank	U	Median difference	95% CI	Mann-Whitney U statistic	p
Response	Statechart 1	1665.5	42.71	635.5	40.0	−15.0 to 93.0	635.5	0.2115
Times	Statechart 2	1415.5	36.29	885.5				

Fig. 16. The cumulative performance of each group with respect to time for both statecharts.

reading mistakes. However, the linear correlation analysis performed has shown that this was indeed not the case. This means that the subjects would undertake the prescribed tasks at their normal pace. The subjects did not hurry-up or slow-down in order to ensure accuracy in reading the models.

6.10 Threats to Validity

This section presents the threats to the validity of the study in accordance with the standard classification [79].

Conclusion validity. Heterogeneity is the most obvious threat when using professionals from various backgrounds, despite most of them mentioning before the experiment that they have experience with statecharts. To mitigate against this threat the subjects were subjected to in-depth training sessions about statechart modeling and its original notation. The new notation is inherently unknown to any of the subjects before the experiment. All subjects were subjected to the same level of training with the new notation as well. The subjects were involved in hands-on exercises to develop several statecharts despite the fact that they were not required to perform any statechart modeling during the experiment. However, the hands-on exercises will further familiarize the subjects with the notations: original and new.

Internal validity. Fatigue and maturity are the forefront internal threats in any subject-based experiment. To mitigate against this threat the training sessions and experimental tasks were conducted on different days. As for the effect of fatigue and maturity during the experiment, it is not believed that these two aspects affected the results since the groups finished all exercises in approximately 14 min (on average). Group A's slowest subject needed 16 min and 10 sec while group B's slowest subject required 15 min and 6 sec. Such task durations are well within the physical abilities of the subjects as per their daily job requirements.

The experiment was conducted under the context of voluntary participation in exchange for the learning value. This raises the issue of self-selection. The threat of self-selection in subject-based experiment that uses professionals can only be fully eliminated if the experimental tasks were assigned by an authority figure relative to the professionals, such as their direct employers or managers. Even if such

TABLE 16
Descriptive Statistics for Group A versus Group B (n = 39)

Variable	Group	Min	1st Quartile	Median	3rd Quartile	Max	IQR
Response	A	641	734.7	759.5	832.2	970	97.5
Times	B	471	564.7	735.0	774.3	906	209.7

TABLE 17
Mann-Whitney Test for Group A versus Group B (n = 39)

Quality Attribute	Group	Rank sum	Mean rank	U	Median difference	95% CI	Mann-Whitney U statistic	p
Response	A	426.0	23.67	123.0	82.5	−3.0 to 183.0	123.0	0.0630
Times	B	354.0	16.86	255.0				

conditions were satisfied, the mandatory nature of the experiment will then raise morality threats and disinterest. The participation of subjects in the experiment on voluntarily basis eliminates the threat of morality issues. The subjects were self-motivated to learn and participate in the experiment. Self-selection would be a greater threat if the target of the experiment was to measure the effectiveness of a technique in absolute terms, whereby self-selection would result in an unrepresentative population of the average work-force. However, in this experiment the target was to compare the cognitive effectiveness of two notations. Self-selection was also applied to participating in the experiment as a whole, but not to deciding which group to join, which was determined by the researchers randomly.

Conducting the experiment over multiple sessions raises the issue of having different environment settings. To mitigate against this threat, the sessions were always conducted during the same time of the day. The sessions were also always conducted during a weekday. All experiment sessions were conducted in a three weeks span to eliminate any effects results from a season change. During each session, the subjects were requested to sit far from each other in order to ensure that they do not collaborate during the experiment.

There is also a threat of bias given the ever-presence of one of the authors during the experiments. It should be noted however that the experiment was conducted with the help of graduate students. Despite the ever-presence of the author, the author was not "proctoring" the participants. The author was sitting in one place throughout the experiment and only attending to participants who requested his assistance to answer questions. The author and other experiment conductors tried to avoid an overbearing presence to the best of our abilities. However, it can be argued that leaving the participants unattended may have reduced unintentional situational pressures. This experiment design decision would undoubtedly result in many uncontrolled in a controlled experiment. Examples can include:

- Participants collaborating with each other.
- Participants forgetting to start or end the timer on time.
- Participants using the bathroom and forgetting to pause the timer.
- Participants losing focus and drifting away whilst the timer is running.
- Therefore it can be argued that the advantage of being physically present (without overbearing the participants) would heavily outweigh the advantage of leaving them to work independently.

Training was offered by the first author, and bias in favor of the author by the subjects was prevented by not informing the subjects that the creator of the new notation is the trainer. We are not aware of any participant explicitly knowing this fact or if any participant deduced that themselves. The concepts of the original notation were revised for all the subjects despite some of them already having some experience. This revision of original notation helped increase homogeneity. Teaching of the original notation took much more time than the extended notation because the original notation was taught first and the extended notation seemed a relatively small addition to their knowledge base. However, the subjects went through the same number of training exercises for both types of notations.

Construct validity. It is almost impossible to fully eliminate the construct validity threat of the dependent variables in subject-based experiments. Their threat was minimized to the greatest possible extent by using a traditional 2×2 fractional factorial experiment design. This commonly used experiment design minimizes the effects of individual capabilities, system differences and ordering effects. Bias towards the new or original notation is minimized on two fronts: (a) First, the subjects were unaware of the hypothesis until the end of the experiment. (b) Second, the subjects were unaware that the new notation was devised by the authors of the paper.

In this experiment we assigned equal weighting to all types of defects. Perhaps some defects are more important that others, however there lacks any empirical evidence that proves a differentiation between the defect types that will allow us to safely quantify defects differently.

6.1.4 External Validity

Unlike experiments that use students as subjects, there is greater confidence to generalize the results of this experiment to industry professionals. However, the results should be interpreted with a conservative view due to two reasons: First, all the professionals involved had three years of industrial experience or less. Would the results differ had the subjects been more seasoned professionals? This remains unknown. Second, the professionals used indicated that they deal with security aspects during daily software development work; however none of them has explicitly identified themselves as expert security specialist or statechart modeling experts. The experiment does not require expert security specialists as subjects; however, would the results differ had the subjects been expert security specialists? Once again, this remains unknown.

A common reason for not generalizing the results of software engineering related experiments to industry is the size of the artifacts utilized in the experiments as they are considered smaller than artifacts used in full-scale industrial settings. However, the new proposed notation is the only notation, to the authors' best knowledge, which models security aspects in statecharts. Therefore, diagrams created using the new proposed notation by default have no precedence in industry, which disables benchmarking efforts. The only benchmarking that could be done is with other similar empirical studies that were performed to evaluate the readability and understandability of models. The models used in this experiment were made to be at least twice the size of the models used in similar empirical studies such as [58], [59], [24], [60].

Another threat to external validity is that the artifacts used in this experiment were only syntactical (abstract). Statecharts used in an industrial setting will certainly refer to domain-specific concepts. The text shown in the nodes (states) and on the edges (events) will be domain-specific. Determining the effect of the extended notation in improving domain-specific security modeling is not evaluated by this experiment.

6.11 Experiment Replication

In conformance with the commended practice to allow replication of experiments, the experimental artifacts used are available on the paper's companion website [18]. The experimental artifacts include the statecharts used (in color and black and white), the questionnaires and notation legend. Using the artifacts available on the companion website and the experiment design described in this paper, the experiment can be replicated.

7 CONCLUSION AND FUTURE WORK

The concept of state machines is crucial in software systems analysis and development. Statecharts are the ideal mechanism to model the behavior of any state-dependent system. Many real-time and embedded systems are considered to be state-dependent. The UML provides statecharts as the only design diagram that is visually tailored for modeling state-based behavior. As the importance of security in systems is constantly increasing, a notation is needed that supports the specification of security aspects in the design of state-dependent software systems. The visual design of such notation needs to be developed following principles of designing cognitively effective notations. However, there is currently no notation that satisfies this requirement. To counter this deficit, this paper presents a proposed notational set that extends the original UML statecharts notation. The new notational set allows designers to model security related aspects in state machines. This will allow state-dependent systems to have security designed within and so equipped against attacks in case external defensive mechanisms fail. The notational set was developed using a theoretical approach and not aesthetics. The approach included two phases: (a) an ontological analysis to determine the necessary semantics not supported by the current UML statecharts notations and (b) an introduction of new visual symbols based on nine evidence-based principles for developing cognitively effective notations. The design of the new notation accounted for the missing semantics and the original notation.

To validate the semantic transparency of the new symbols, an industrial survey was conducted and only security analysts were invited to participate. The responses of 58 respondents were measured. The results show that the new symbols are largely intuitive and suggestive of their underlying semantics. The responses also show that respondents find the new notation clearer and more suggestive of their underlying semantics in comparison with the original notation. Qualitative data collected from the survey also indicate that the new notation has sufficiently covered the major security related semantics.

To validate the cognitive effectiveness of the proposed notation, we presented an experiment that was conducted as a voluntary exercise using professional software engineers as subjects. The subjects applied two treatments; understanding statecharts that use the extended notation and the original notation. The results indicate a statistically significant improvement with respect to the time the subjects needed to read the models. The results pertaining to the accuracy of reading the models were not clear cut, whereby no statistical significance was observed when considering each statechart in isolation, yet statistical significance was observed when considering the aggregated results from both statecharts. We take a conservative approach by rejecting the hypothesis until new empirical evidence proves otherwise.

Qualitative data obtained from the subjects after the experiment indicates that all subjects finished the experimental tasks without facing time pressure. Therefore, the results obtained with respect to response times were considered to be purely influenced by the treatments. Qualitative data obtained has also shown that the subjects generally preferred the extended notation over the original notation. The subjects indicated that the main reason for their preference of the new notation was the use of color. Two minor improvements were suggested: (a) that the text used in defensive states would be more readable if it was white rather than black, and (b) the recovery state should be given a colorful background rather than its grey color. There were very few questions asked by the subjects during the experiment and in general there were no obvious problems observed during the experiment.

Recall that the statecharts used in the experiment presented in Section 6 are abstract. They were intentionally rid of domain-specific information to the contrary of statecharts used in industrial settings. Since the ultimate goal of this research is to improve practical security modeling, it would certainly be beneficial to perform another empirical study to evaluate the effect of the extended notation on effectiveness of security modeling with domain-specific statecharts. The experiment may also be repeated with a larger number of participants. Perhaps a larger number of participants may lead to statistical significance with respect to the errors committed variable. Future work may also be directed towards conducting empirical studies in which subjects are tasked to perform modeling activities. Such empirical studies will provide further insight on the effect of using the extended notation on security modeling with statecharts as a whole.

Motivated by the results of this research, we strongly recommend other researchers to evaluate other popular notations and suggest improvements based on the principles defined in [51]. The UML alone includes many diagrams that can be evaluated and improved. Evaluations and improvements can be extended to other types of diagrams such as misuse cases, i*, KAOS . . . etc. Their improvements should be empirically evaluated. Notation design has long been a neglected topic in software engineering. It is time now to change the view and approach to notation design. Future work will be directed towards contacting major vendors of UML modeling tools in an effort to incorporate the new notation within their tools. Tool support will significantly support the widespread use and adoption of the new notation. Tool support can also significantly aid in overcoming the issue of using a color-rich notation by the color-blind community whereby a tool can provide its users the option to use different pattern fills instead of colors.

Notation design is ultimately a design activity. Design inherently is a product that can always stand to improve as there is no maximum score. There are many aspects of proposed design that can be improved. The proposed notation is a best first-effort to adhere to Moody's principles while achieving the main objective of the paper, which is to allow

modelers the capability of modeling security related aspects in state-based systems. The main success criterion was based on achieving statistically significant improvements in comparison to the original notation. The proposed notation design is not intended to be a final untouchable design. One outstanding issue for example is the use of too many contrasting primary colors. We encourage research work to be directed towards further improving our design to account aspects such as color harmony and design layout.

The results of the empirical studies also prompt an empirical investigation of the effect of the new notation on the model construction aspect. It is interesting to investigate if the new notation helps modelers create higher quality security statecharts and whether it improves their modeling and analysis skills.

Tool support will have a significant influence on the adoption of the new notation. Tool support is currently under-development. The tool will allow modelers to create statecharts using the new notation without requiring artistic skills. Upon finalizing the metamodel of the new notation, the metamodel will be embedded in the tool which will allow it to verify the syntactical correctness of the diagrams being created. Other similar research works are underway to improve existing notations based on the PoN principles. It is planned for the tool that will be developed to support a multitude of PoN-enhanced notations.

ACKNOWLEDGMENTS

REFERENCES

[1] E. G. Amoroso, *Fundamentals of Computer Security Technology*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1994.

[2] W. A. Arbaugh, W. L. Fithen, and J. McHugh, "Windows of vulnerability: A case study analysis," *IEEE Comput.*, vol. 33, no. 12, pp. 52–59, Dec. 2000.

[3] N. Baddoo and T. Hall, "Motivators of software process improvement: An analysis of practitioners' views," *J. Syst. Softw.*, vol. 62, no. 2, pp. 85–96, 2002.

[4] D. Balzarotti, M. Cova, V. V. Felmetsger, and G. Vigna, "Multimodule vulnerability analysis of web-based applications," in *Proc. 14th ACM Conf. Comput. Commun. Security*, 2007, pp. 25–35.

[5] M. Bar and M. Neta, "Humans prefer curved visual objects," *Psychol. Sci.*, vol. 17, no. 8, pp. 645–648, 2006.

[6] D. A. Basin, J. Doser, and T. Lodderstedt, "Model driven security for process-oriented systems," in *Proc. 8th ACM Symp. Access Control Models Technol.*, 2003, pp. 100–109.

[7] J. Bertin, *Semiology of Graphics: Diagrams, Networks, Maps*. Madison, WI, USA: Univ. of Wisconsin Press, 1983.

[8] C. Britton and S. Jones, "The untrained eye: How languages for software specification support understanding in untrained users," *Human–Comput. Interact.*, vol. 14, nos. 1–2, pp. 191–244, 1999.

[9] R. J. A. Buhr, D. Amyot, M. Elammari, D. Quesnel, T. Gray, and S. Mankovski, "Feature-interaction visualisation and resolution in an agent environment," in *Proc. Feature Interactions Telecommun. Softw. Syst. V*, 1998, pp. 135–149.

[10] C. C. Burt, B. R. Bryant, R. R. Raje, A. Olson, and M. Auguston, "Model driven security: Unification of authorization models for fine-grain access control," in *Proc. Enterprise Distrib. Object Comput. Conf.*, 2003, pp. 159–171.

[11] A. Blackwell. (2009). *Cognitive dimensions of notations resource site*. [Online]. Available: http://www.cl.cam.ac.uk/~afb21/CognitiveDimensions/

[12] A. Blackwell and T. Green, "Notational systems—the cognitive dimensions of notations framework," in *HCI Models Theories Framework Interdisciplinary Science*. San Mateo, CA, USA: Morgan Kaufmann, 2003.

[13] J. Dagit, J. Lawrance, C. Neumann, M. Burnett, R. Metoyer, and S. Adams, "Using cognitive dimensions: Advice from the trenches," *J. Visual Languages Comput.*, vol. 17, no. 4, pp. 302–327, 2006.

[14] A. Dardenne, A. van Lamsweerde, and S. Fickas, "Goal-directed requirements acquisition," *Sci. Comput. Program.*, vol. 20, no. 1, pp. 3–50, 1993.

[15] T. DeMarco, *Structured Analysis and System Specification*. Upper Saddle River, NJ 07458: Yourdon Press, 1979.

[16] E. Dubois and S. Wu, "A framework for dealing with and specifying security requirements in information systems," in *Proc. Inform. Syst. Security*, 1996, pp. 88–99.

[17] O. El Ariss, W. Jianfei, and X. Dianxiang, "Towards an enhanced design level security: Integrating attack trees with statecharts," in *Proc. 5th Int. Conf. Secure Softw. Integr. Rel. Improvement*, 2011, pp. 1–10.

[18] M. El-Attar (2013, May). *Companion website to security enabled statecharts research*. [Online]. Available: http://faculty.kfupm.edu.sa/ICS/melattar/ExtendedStatechartsNotationFiles.html

[19] C. Ericson, "Fault tree analysis—a history," in *Proc. 17th Int. Syst. Safety Conf.*, 1999, pp. 1–9.

[20] S. Fenz and A. Ekelhart, "Formalizing information security knowledge," in *Proc. ACM Symp. Informa., Comput. Commun. Security*, 2009, pp. 183–194.

[21] P. Fettke, "How conceptual modeling is used," *Commun. Assoc. Inf. Syst.*, vol. 25, no. 1, p. 43, 2009.

[22] C. P. Gane and T. Sarson, *Structured Systems Analysis: Tools and Techniques*. Englewood Cliffs, NJ, USA: Prentice Hall, 1979.

[23] N. Goodman, *Languages of Art: An Approach to a Theory of Symbols*. Indianapolis, IN, USA: Bobbs-Merrill, 1968.

[24] S. Gopalakrishnan, J. Krogstie, and G. Sindre "Adapting UML activity diagrams for mobile work process modelling: Experimental comparison of two notation alternatives," in *Proc. Practice Enterprise Model.*, 2010, pp. 145–161.

[25] G. J. Gorn, A. Chattopadhyay, T. Yi, and D. W. Dahl, "Effects of color as an executional cue in advertising: They're in the shade," *Manag. Sci.*, vol. 43, no. 10, pp. 1387–1400, 1997.

[26] T. R. Green, A. E. Blandford, L. Church, C. R. Roast, and S. Clarke, "Cognitive dimensions: Achievements, new directions, and open questions," *J. Vis. Lang. Comput.*, vol. 17, no. 4, pp. 328–365, 2006.

[27] T. R. G. Green and M. Petre, "Usability analysis of visual programming environments: A 'cognitive dimensions' framework," *J. Vis. Lang. Comput.*, vol. 7, no. 2, pp. 131–174, 1996.

[28] T. Hall, A. Rainer, and N. Baddoo, "Implementing software process improvement: An empirical study," *Softw. Process Improvement Practice*, vol. 7, no. 1, pp. 3–15, 2002.

[29] R. Hassan, S. Bohner, S. El-Kassas, and M. Hinchey, "Integrating formal analysis and design to preserve security properties," in *Proc. 42nd Hawaii Int. Conf. Syst. Sci.*, 2009, pp. 1–10.

[30] A. Herzog, N. Shahmehri, and C. Duma, "An ontology of information security," *Int. J. Inform. Security*, vol. 1, no. 4, pp. 1–23, 2007.

[31] S. Hitchman, "The details of conceptual modelling notations are important—a comparison of relationship normative language," *Commun. Assoc. Inf. Syst.*, vol. 9, no. 1, p. 10, 2002.

[32] P. Irani and C. Ware, "Diagramming information structures using 3D perceptual primitives," *ACM Trans. Comput.-Human Interaction*, vol. 10, no. 1, pp. 1–19, 2003.

[33] P. Irani, C. Ware, and M. Tingley, "Using perceptual syntax to enhance semantic content in diagrams," *IEEE Comput. Graph. Appl.*, vol. 21, no. 5, pp. 76–85, Sep. 2001.

[34] J. Jürjens, *Secure Systems Development with UML*. Berlin, Germany: Springer, 2005.

[35] J. Jürjens, "UMLsec: Extending UML for secure systems development," in *Proc. 5th Int. Conf. UML - Unified Model. Language*, 2002, pp. 412–425.

[36] P. Kárpáti, G. Sindre, and A. Opdahl, "Visualizing cyber attacks with misuse case maps," in *Proc. Requirements Eng.: Found. Softw. Quality*, 2010, pp. 262–275.

[37] V. Katta, P. Kárpáti, A. Opdahl, C. Raspotnig, and G. Sindre, "Comparing two techniques for intrusion visualization," in *Proc. Practice Enterprise Model.*, 2010, pp. 1–15.

[38] S. U. Khan, M. Niazi, and R. Ahmad, "Empirical investigation of success factors for offshore software development outsourcing vendors," *IET Softw.*, vol. 6, no. 1, pp. 1–15, 2012.

[39] B. Kordy, S. Mauw, S. Radomirović, and P. Schweitzer, "Foundations of attack—defense trees," in *Proc. Formal Aspects Security Trust*, 2011, pp. 80–95.

[40] I. V. Krsul, "Software vulnerability analysis," PhD thesis, Purdue Univ., West Lafayette, IN, USA, 1998.

[41] J. H. Larkin and H. A. Simon, "Why a diagram is (sometimes) worth ten thousand words," *Cogn. Sci.*, vol. 11, no. 1, pp. 65–100, 1987.

[42] E. Lehmann and H. D'Abrera, *Nonparametrics: Statistical Methods Based on Ranks*. Upper Saddle River, NJ, USA: Prentice Hall, 1998.

[43] L. Lin, B. Nuseibeh, D. Ince, M. Jackson, and J. Moffett, "Introducing abuse frames for analysing security requirements," in *Proc. 11th IEEE Int. Conf. Requirements Eng.*, 2003, pp. 371–372.

[44] (2015 Jan.). LinkedIn Corp., LinkedIn website. [Online]. Available: https://www.linkedin.com

[45] L. Liu, E. Yu, and J. Mylopoulos, "Security and privacy requirements analysis within a social setting," in *Proc. 11th IEEE Int. Conf. Requirements Eng.*, 2003, pp. 151–161.

[46] T. Lodderstedt, D. Basin, and J. Doser, "SecureUML: A UML-based modeling language for model-driven security," in *Proc. 5th Int. Conf. UML Unified Model. Language*, 2002, pp. 426–441.

[47] J. D. Mackinlay, "Automating the design of graphical presentations of relational information," *ACM Trans. Graph.*, vol. 5, no. 2, pp. 110–141, 1986.

[48] K. Masri, D. Parker, and A. Gemino, "Using iconic graphics in entity-relationship diagrams: The impact on understanding," *J. Database Manage.*, vol. 19, no. 3, pp. 22–41, 2008.

[49] G. A. Miller, "The magical number seven, plus or minus two: Some limits on our capacity for processing information," *Psychol. Rev.*, vol. 63, no. 2, p. 81, 1956.

[50] K. D. Mitnick and W. L. Simon, *The Art of Intrusion: The Real Stories Behind the Exploits of Hackers, Intruders and Deceivers*. Hoboken, NJ, USA: Wiley, 2009.

[51] D. Moody, "The 'physics' of notations: Toward a scientific basis for constructing visual notations in software engineering," *IEEE Trans. Softw. Eng.*, vol. 35, no. 6, pp. 756–779, Nov. 2009.

[52] H. Mouratidis and P. Giorgini, "Secure tropos: A security-oriented extension of the tropos methodology," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 17, no. 2, pp. 285–309, 2007.

[53] M. Niazi, D. Wilson, and D. Zoghwi, "Critical success factors for software process improvement implementation: An empirical study," *Softw. Process Improvement Practice*, vol. 11, no. 2, pp. 193–211, 2006.

[54] J. C. Nordbotten and M. E. Crosby, "The effect of graphic style on data model interpretation," *Inf. Syst. J.*, vol. 9, no. 2, pp. 139–156, 1999.

[55] OMG. (2011). *Unified modeling language*. [Online]. Available: http://www.omg.org/spec/UML/2.4.1/

[56] A. Paivio, *Mental Representations: A Dual Coding Approach*. London, U.K.: Oxford Univ. Press, 1986.

[57] S. L. Pfleeger and B. A. Kitchenham, "Principles of survey research: Part 1: Turning lemons into lemonade," *ACM SIGSOFT Softw. Eng. Notes*, vol. 26, no. 6, pp. 16–18, 2001.

[58] H. C. Purchase, D. Carrington, and J.-A. Allder, "Empirical evaluation of aesthetics-based graph layout," *Empirical Softw. Eng.*, vol. 7, no. 3, pp. 233–255, 2002.

[59] H. C. Purchase, R. Welland, M. McGill, and L. Colpoys, "Comprehension of diagram syntax: An empirical study of entity relationship notations," *Int. J. Human-Comput. Stud.*, vol. 61, no. 2, pp. 187–203, 2004.

[60] H. A. Reijers and J. Mendling, "A study into the factors that influence the understandability of business process models," *IEEE Trans. Syst. Man Cybern. Part Syst. Humans*, vol. 41, no. 3, pp. 449–462, 2011.

[61] J. L. Rodgers and W. A. Nicewander, "Thirteen ways to look at the correlation coefficient," *Am. Statistician*, vol. 42, no. 1, pp. 59–66, 1988.

[62] L. Røstad, "An extended misuse case notation: Including vulnerabilities and the insider threat," in *Proc. 12th Working Conf. Requirements Eng.: Found. Softw. Quality*, 2006, pp. 33–43.

[63] H. Schmidt and J. Jürjens, "Connecting security requirements analysis and secure design using patterns and UMLsec," in *Proc. 23rd Int. Conf. Adv. Inform. Syst. Eng.*, 2011, pp. 367–382.

[64] B. Schneier (1999, Dec.). Attack trees *Dr. Dobb's J.* [Online]. *24(12)*. Available: https://www.schneier.com/paper-attacktrees-ddj-ft.html

[65] S. S. Shapiro and M. Wilk, "An analysis of variance test for the exponential distribution (complete samples)," *Technometrics*, vol. 14, no. 2, pp. 355–370, 1972.

[66] K. Siau, "Informational and computational equivalence in comparing information modeling methods," *J. Database Manage.*, vol. 15, no. 1, pp. 73–86, 2004.

[67] S. Siegel and N. Castellan, *Non Parametric Statistics for the Behavioral Sciences*, 2nd ed. New York, NY, USA: McGraw-Hill, 1988.

[68] G. Sindre and A. L. Opdahl, "Eliciting security requirements with misuse cases," *Requirements Eng.*, vol. 10, no. 1, pp. 34–44, 2005.

[69] G. Sindre, "Mal-activity diagrams for capturing attacks on business processes," in *Proc. 13th Int. Working Conf. Requirements Eng.: Found. Softw Quality*, 2007, pp. 355–366.

[70] G. Sindre, A. Opdahl, and G. F. Brevik, "Generalization/specialization as a structuring mechanism for misuse cases," in *Proc. 2nd Symp. Requirements Eng. Inform. Security*, 2002, pp. 1–16.

[71] A. Souag, C. Salinesi, and I. Comyn-Wattiau, "Ontologies for security requirements: A literature survey and classification," in *Proc. Adv. Inform. Syst. Eng. Workshops*, 2012, pp. 61–69.

[72] M. R. Stytz, "Considering defense in depth for software applications," *IEEE Security Privacy*, vol. 2, no. 1, pp. 72–75, Jan./Feb. 2004.

[73] Symantec. (2010, Aug.). *AndroidOS.FakePlayer description*. [Online]. Available: http://www.symantec.com/security_response/writeup.jsp?docid=2010-081100-1646-99

[74] A. van Lamsweerde, "Elaborating security requirements by construction of intentional anti-models," in *Proc. 26th Int. Conf. Softw. Eng.*, 2004, pp. 148–157.

[75] Y. Wand and R. Weber, "An ontological model of an information system," *IEEE Trans. Softw. Eng*, vol. 16, no. 11, pp. 1282–1292, Nov. 1990.

[76] Z. O'Connor, "Logo colour and differentiation: A new application of environmental colour mapping," *Color Res. Appl.*, vol. 36, no. 1, pp. 55–60, 2011.

[77] J. Whittle, "Extending interaction overview diagrams with activity diagram constructs," *Softw. Syst. Model.*, vol. 9, no. 2, pp. 203–224, 2010.

[78] W. Winn, "An account of how readers search for information in diagrams," *Contemporary Educ. Psychol.*, vol. 18, no. 2, pp. 162–185, 1993.

[79] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, Experimentation in software engineering, Berlin Heidelberg: Springer, Jun. 2012.

[80] J. Zhang and D. A. Norman, "Representations in distributed cognitive tasks," *Cogn. Sci.*, vol. 18, no. 1, pp. 87–122, 1994.

Mohamed El-Attar received the BEng degree from Carleton University, Canada, and the PhD degree from the University of Alberta, Canada. In 2009, he joined the Department of Information and Computer Science at the King Fahd University of Petroleum and Minerals, Saudi Arabia, and is currently an associate professor. His research interests include requirements engineering, in particular with UML and use cases, object-oriented analysis and design, model transformation, empirical studies, and visual syntax evaluation and design. For information about his research see: http://faculty.kfupm.edu.sa/ICS/melattar/index.html.

Hamzah Luqman received the BS degree from Ahgaff University, Yemen in 2006 and the MS degree in computer science from King Fahd University for Petroleum and Minerals, Saudi Arabia in 2013. He is working towards the PhD degree at the Department of Computer Science and Engineering, University of King Fahd University for Petroleum and Minerals. His research interests include software engineering, pattern recognition, document analysis and understanding, image processing, and artificial intelligence. He has publications in Arabic font recognition and distributed computing.

Peter Karpati graduated from the Eötvös Loránd University Budapest in 2001 and received the PhD degree from Klagenfurt University in 2007. He worked as a postdoc researcher at the Norwegian University of Science and Technology in Trondheim between 2007 and 2012. Since 2012 he works at the Institute of Energy Technology in Norway as a senior scientist. He has published in the areas of multimedia systems, healthcare informatics, security and safety modeling in peer-reviewed international journals and conferences. His current research interests cover combining safety and security assessments and requirements as well as challenges in justification of safety critical systems.

Guttorm Sindre received the MSc degree in computer science (information systems engineering) in 1988, and the PhD degree in the same field in 1990. He has been a full professor in information systems engineering at the Department of Computer and Information Science, Norwegian University of Science and Technology (NTNU) from 2003. His research interests are within the areas of conceptual modeling, information systems engineering, and software engineering—especially focusing on requirements engineering and techniques for elicitation and specification of security requirements. He has published more than 100 peer-reviewed papers in workshops, conferences and journals, and has been on the program committee of many international conferences, including CAiSE (where he was program co-chair in 2007), IEEE RE (where he was posters and demos co-chair in 2008), and REFSQ. He has been involved in several national and international research projects, and has made two sabbatical visits to the University of Auckland, New Zealand. He is a member of the IEEE.

Andreas L. Opdahl received the MSc and PhD degrees in computer science from the Norwegian University of Science and Technology in 1992. He is a professor of information systems development at the Department of Information Science and Media Studies, University of Bergen, Norway. He has published more than a hundred journal articles, book chapters, refereed archival conference papers, and books. His research interests include enterprise and IS modelling, requirements engineering, safety and security requirements, and semantic technologies. He is a member of IFIP WG5.8 on enterprise Interoperability and WG8.1 on Design and Evaluation of Information Systems. He serves regularly as a reviewer for the premier international journals in his field and as an organiser and on the program committees of renowned international conferences and workshops. He is a member of the IEEE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.