Evaluation of the Cognitive Effectiveness of the CORAS Modelling Language

Eloïse Zehnder[1], Nicolas Mayer[1], and Guillaume Gronier[1]

[1] Luxembourg Institute of Science and Technology, 5, avenue des Hauts-Fourneaux,
L-4362 Esch-sur-Alzette, Luxembourg
{eloise.zehnder, nicolas.mayer, guillaume.gronier}@list.lu

Abstract. Nowadays, Information System (IS) security and Risk Management (RM) are required for every organization that wishes to survive in this networked and open world. Thus, more and more organizations tend to implement a security strategy based on an ISSRM (IS security RM) approach. However, the difficulty of dealing efficiently with ISSRM is currently growing, because of the complexity of current IS coming with the increasing number of risks organizations need to face. To use conceptual models to deal with RM issues, especially in the information security domain, is today an active research topic, and many modelling languages have been proposed in this way. However, a current challenge remains the cognitive effectiveness of the visual syntax of these languages, i.e. the effectiveness to convey information. Security risk managers are indeed not used to use modelling languages in their daily work, making this aspect of cognitive effectiveness a must-have for these modelling languages. Instead of starting defining a new cognitive effective modelling language, our objective is rather to assess and benchmark existing ones from the literature. The aim of this paper is thus to assess the cognitive effectiveness of CORAS, a modelling language focused on ISSRM.

Keywords: Security, Risk Management, Visual Syntax, Physics of Notations.

1 Introduction

Risk management is today a steering instrument used in many domains, such as, for example finance, insurance, environment or security. From a process perspective, risk management has been standardized for years, at a generic level like in ISO 31000, as well as in specific domains such as ISO/IEC 27005 for information security. However, at the product level (i.e. the result(s) obtained as output of the different steps of the risk management process), a great variability can be observed, going from tables to detailed and specific conceptual models (e.g., in the security domain, attack-trees [1] or enterprise architecture models [2]). To propose conceptual models to deal with risk management, especially in the information security domain, is today an active research topic, and our aim is to contribute to it by improving existing languages.

In this context, this paper ties in a broader project that aims at integrating conceptual models with Information System Security Risk Management (ISSRM) [3]. This integration seems to us as a promising approach to deal with issues related to the complexity of organizations and associated risks, especially the difficulty to have a clear and manageable documentation for ISSRM activities. One of our main concerns is to take into account the target users' group of our research results that is information security risk managers. This target group is not used to use conceptual models in its daily work, and the associated modelling languages need so to be effective to this target group to convey information, i.e. to be cognitively effective.

In this frame, we have already assessed the "Risk and Security Overlay" (RSO) of the ArchiMate language [2], ArchiMate being a standardized modelling language developed by The Open Group to provide a uniform representation for diagrams that describe Enterprise Architecture (EA). The conclusion established was that the RSO can decently not be considered as an appropriate notation from a cognitive effectiveness point of view and that there is room to propose a notation better on this aspect [4]. CORAS is another well-known modelling language for ISSRM [5]. It is thus a second candidate we want to evaluate. The research question addressed in this paper is then: how cognitive effective is the CORAS language to support the users in their ISSRM activities?

The remainder of the paper is structured as follows. In the next section, the background of our work is described: it introduces cognitive effectiveness and the "Physics of Notation" (PoN), a set of nine principles we use to assess cognitive effectiveness, and then the CORAS approach itself. Section 3 is about related work. Section 4 presents the assessment of the cognitive effectiveness of CORAS: the approach followed and the results obtained. Section 4 is the discussion about the results. Finally, Section 5 concludes about our current work and presents our future work.

2 Background

2.1 Cognitive effectiveness and the "Physics of Notation"

Conceptual models are now widely used to visually communicate a great deal of information (about processes, systems, etc.) to users. Despite their interest in the communication of complex information, conceptual models can, however, give interpretation problems to their users. Indeed, several empirical studies have already shown that they can be misunderstood [6–9]. To improve the understanding of conceptual models, one of the most relevant approach comes from the cognitive sciences and is called 'cognitive effectiveness'. Cognitive sciences refer to the theories of information processing, which specifically include many concepts from cognitive psychology such as perception, memory (short and long term), attention or information processing. Thus, conceptual models are understood as an interaction between a user and a visual representation. In this context, cognitive effectiveness is embodied in the ability of conceptual models to support appropriate translations between cognitive and visual models [10].

Cognitive effectiveness in software engineering has been addressed for years [11] but it became a more active research topic since Moody's work. In order to evaluate and obtain a better quality in the design of modelling languages, Moody has established nine principles, called the "Physics of Notation" (PoN) [12]. These principles have already been applied to assess the cognitive effectiveness of many different modelling languages [13–16] and are used in this paper to assess the one of CORAS. The nine principles, who will be detailed later, are: semiotic clarity, graphic economy, perceptual discriminability, visual expressiveness, dual coding, semantic transparency, cognitive fit, complexity management, and cognitive integration.

2.2 The CORAS Approach

CORAS is an approach for ISSRM consisting basically of a modelling language, a method and a tool [5]. A particular focus is done in this paper on the modelling language of CORAS and associated models. The CORAS language is composed of five types of (so-called) diagrams. The first one is the asset diagram, describing the focus of the analysis and coming with the context establishment. Then, the threat diagram supports the risk identification and the risk estimation steps. Threat diagram describes "scenarios which may cause harm to the assets". An example of threat diagram is proposed in Fig. 1. Next, risk diagram basically summarizes the risks presented in threat diagrams. Finally, treatment overview diagram proposes treatments to risks. The CORAS language also includes three extensions. High-Level CORAS supports abstraction and comprehensible overviews or large risk models. Dependent CORAS supports documentation of assumptions and provide tools to separate the target of study from assumptions. Finally, Legal CORAS supports documentation of legal aspects (legal risks and legal norms) and their impact.

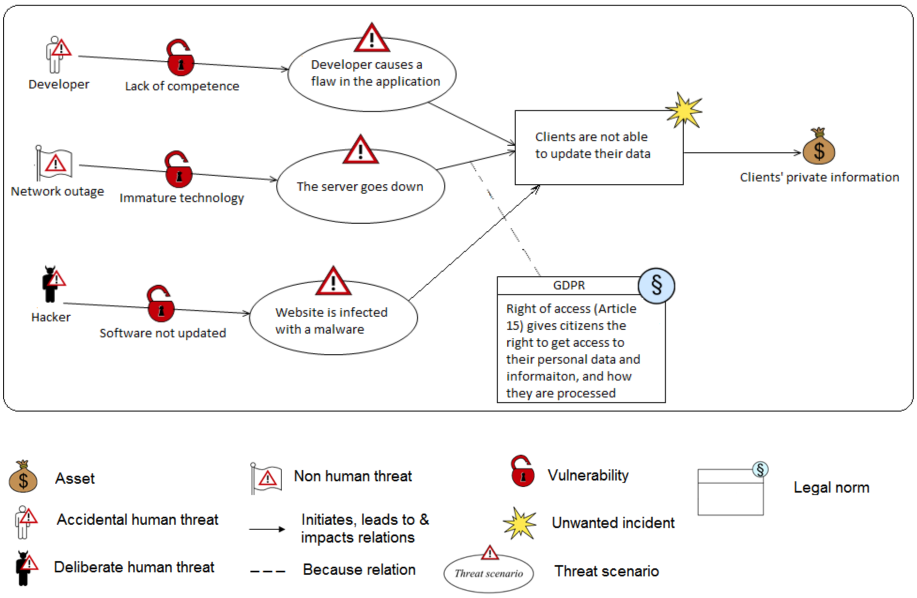
Developer
Lack of competence
Developer causes a flaw in the application
Network outage
Immature technology
The server goes down
Clients are not able to update their data
Clients' private information
Hacker
Software not updated
Website is infected with a malware
GDPR
§
Right of access (Article 15) gives citizens the right to get access to their personal data and informaiton, and how they are processed
Asset
Accidental human threat
Deliberate human threat
Non human threat
Initiates, leads to & impacts relations
Because relation
Vulnerability
Unwanted incident
Threat scenario
Threat scenario
Legal norm

Fig. 1. Example of a threat diagram, including the Legal CORAS extension

For illustration purpose, Fig. 1 is an example of threat diagram. In this example, a network outage (non-human threat) may harm the clients' private information (asset) by making the server going down (threat scenario) caused by the use of an immature technology (vulnerability). The GDPR (legal norm) actually puts a legal risk because citizens shall have access to their data and they shall be able to update them. The fact that the clients are not able to update their data (unwanted incident) can also occur because of deliberate or accidental human threats, respectively because a developer causes a flaw in the application because of lack of competence or because a hacker exploits a non-updated software to infect a website with a malware.

In this paper, our analysis is on the CORAS language as a whole (i.e. all provided diagrams) and is based on the reference book entitled: "Model-Driven Risk Analysis: The CORAS Approach" [5]. It has the advantage of presenting the CORAS modelling language in a self-contained, up-to-date and detailed manner. We are also aware of the existence of the more recent ISMS-CORAS method [17] which adds more detailed steps in the process and more diagrams and symbols related to the implementation of an Information Security Management System (ISMS). However, we decided to stick at the core CORAS approach and not to study this context-specific extension.

3 Related works

According to a recent study based on a Systematic Literature Review [18], Moody's framework and associated principles are one of the most cited and used approach in the assessment of visual notations. In this review, 70 papers dealing with the application of the PoN for the development and/or evaluation of a visual notation were identified, excluding overlapping versions of already included work. Languages such as UML [13], *i** [14], BPMN 2.0 [15], or ArchiMate [16] (to name just a few) were already evaluated thanks to the principles of the PoN. The PoN is globally appreciated for his scientific support and accuracy compared to other proposals in the same domain. However, the PoN approach is targeted by some criticisms coming mainly from the lack of clear guidelines provided for its practical application to assess an existing language. Operationalisations of the principles have already been attempted [19, 20], but they remain unsatisfactory because incomplete (they do not include all the principles) or impractical to use in real cases for some principles according to their authors themselves [18].

Some other frameworks exist to support visual syntax assessment. The SEQUAL framework, developed by Krogstie *et al.* [21], addresses the quality of every aspects of a modelling language through several qualities. Putting the focus on visual syntax, SEQUAL is considered as lacking some concrete guidelines to design effective visual notations, as argued by Genon [22]. Frank addresses the visual notation design in the context of the Domain-Specific Modeling Language which represents the concepts and constraints of a well-defined domain-level knowledge [23]. Regarding the design of graphical notations, the guidelines provided are built on the author's own experience and on the respective literature analysis. However, the approach developed by Frank aim at being applied during the design phase of the graphical notation. The Cognitive

Dimensions (CDs) of notations is another framework aiming at improving design practice by focussing on the usability aspects of artefacts [11]. It provides different dimensions (from 9 to 13 depending on the users) that can be used during exploratory design of a modelling language. Guizzardi *et al.* propose an ontology-based assessment and design method of domain-specific visual modelling languages [24]. This method aims at evaluating, on the one hand, the language ability to support the users in communicating and reasoning with the produced models and on the other hand, its truthfulness and appropriateness to the domain which it is supposed to represent. Kleppe introduces and explains the factors that influence an effective domain specific language design and proposes a design strategy for language creation [25]. Once again, these approaches are developed to be used during design time of a modelling language and are less suited than the PoN for the evaluation of an existing visual notation.

4 Assessment of the cognitive effectiveness of CORAS

4.1 Methodological approach

Since the PoN principles are today considered as one of the most advanced framework to evaluate an existing visual modelling language [22], our analysis will be based on these principles. However, as argued in the previous section, their operationalisation remains a complex and open issue. What they lack the most is usability, which, according to ISO 9241-11 can be defined by "the degree to which a product can be used, by means of identified users, to achieve defined goals with efficiency, effectiveness and satisfaction, in a context of specified use". Indeed, Moody's principles are neither proposing quantitative benchmarks allowing us to know the limitations of the syntax to elaborate, nor proper guidelines to use them. Therefore, we will follow recommendations established thanks to lessons learnt from previous usages of these principles: try to operationalise the principles in quantitative metrics to objectify the analysis, whenever it is possible, and apply the PoN with care by providing accountability with design rationale of the evaluated language [18, 20].

To do so, we first operationalised all the principles into tables to get a better view of them, gathering their definition, with the specific characteristics of each of them (for example, cognitive integration regroups conceptual and perceptual integration), and associated metrics established from the definition of the principle. In parallel, we gathered every symbol of CORAS in a document and we designed an illustrative example for every type of diagram included in the language in order to have a concrete example supporting our evaluation (not included in this paper because of space constraints).

Aiming simplicity and usability in its format, our evaluation approach is intended to be used under the guise of the expert or heuristic evaluation. The heuristic evaluation is "an informal method of usability analysis where a number of evaluators are presented with an interface design and asked to comment on it" [26]. With this method, end-users are not involved in the evaluation process, the evaluation being only based on expert analysis. According to their authors, heuristic evaluation can be improved significantly by involving multiple evaluators. Thus, three evaluators (that are the authors of this paper) were involved in the review of the CORAS language: one of them is an expert in security and risk management, the two others are expert in cognitive science. Heuristic evaluation was performed by each evaluator who inspected the CORAS language alone. Then, the individual findings were discussed and aggregated. The next section summarizes the results obtained.

4.2 Results

Hereafter, we analyse the cognitive effectiveness of the RSO through the nine principles elaborated by Moody. We first remind a short definition for each principle, extracted from the PoN reference article [12]. Then, we report on how CORAS meets this principle, based on associated metrics.

Principle of semiotic clarity. According to the semiotic clarity principle, there should be a one-to-one correspondence between semantic constructs and graphical symbols. CORAS is a language composed of 17 different graphical symbols and 20 semantic constructs overall. There is a 1:1 correspondence between semantic constructs and graphical symbols for almost all constructs. The only exception is for the following constructs: *initiates*, *leads-to*, *impacts* and *harm* relations are all represented with the same arrow as graphical symbol. According to Moody, there is thus an anomaly with regards to the semiotic clarity principle called symbol overload (when two different constructs are represented by the same graphical symbol) occurring 6 times (one for each couple of two different constructs represented by the same graphical symbol). The other anomalies with regards to semiotic clarity have an occurrence of 0, as reported in Table 1.

Table 1. Anomalies with regards to semiotic clarity.

Anomaly	Definition	Occurrence
Symbol redundancy	Multiple graphical symbols can be used to represent the same semantic construct	0
Symbol excess	Graphical symbols do not correspond to any semantic construct	0
Symbol overload	Two different constructs are represented by the same graphical symbol	6
Symbol deficit	There are semantic constructs that are not represented by any graphical symbol	0

Principle of perceptual discriminability. Regarding perceptual discriminability, different symbols should be clearly distinguishable from each other. Discriminability is primarily determined by the visual distance between symbols. Visual distance between symbols occurs when these symbols differ on a sufficient number of visual variables (e.g., shape, size, colour, position, etc.) For this purpose, we compared the visual distance between every symbol two-by-two in a grid (not reported here because of space constraints). CORAS takes profit of three different visual variables in its language (colour, texture and shape; see the visual expressiveness principle for additional information), leading to a visual distance that could be between 0 and 3. As an example, the differences between the *risk* and the *stakeholder* symbols are the shape (warning sign versus human shape in a suit) and the colour (red and white versus white and brown). This gives us a visual distance of 2 for this couple. Among a total of 190 couples analysed, the summary of the visual distance between constructs is reported in Table 2.

We found that symbols are mostly separated by a visual distance of 2 and 6 interactions have a visual distance equal to 0, which are obviously the ones for which symbol overload has been found for the principle of semiotic clarity.

Table 2. Visual distance between the semantic constructs.

Visual distance	Occurrence
Visual distance equal to 0	6
Visual distance equal to 1	26
Visual distance equal to 2	125
Visual distance equal to 3	33

Principle of semantic transparency. Semantic transparency corresponds to the use of visual representations whose appearances suggest their real meaning without inducing another and/or false one. In other words, the meaning of a symbol should be understood by looking at its representation.

Determining when a symbol suggests its meaning (or not) is a quite subjective task since it can depend on a lot of individual variables such as culture or education. We thus decided to count the number of symbols (conceptual forms you can link to a concept, a referent) and the number of signs (non-representational symbol, arbitrarily assigned with a wholly learned connection to a referent) according to the work of Zender and Mejía [27]. The limit we try to define regarding semantic transparency resides in "what has to be learned or not in order to be understood".

We found 12 symbols (*unwanted incident*, *asset*, *indirect asset*, *threat scenario*, *treatment scenario*, *risk*, *vulnerability*, *stakeholder*, *deliberate human threat*, *accidental human threat*, *non-human threat*, *referring scenario*), and 5 signs (*legal norm*, *border line*, *initiates, leads-to, impacts and harm relations – having all four the same symbol, treats relation* and *because relation*). As an example, the legal norm and its '§' sign, according to the authors, does not explicitly refers to a legal symbol. None can guess the legal norm symbol refers to something at least legal unless it is learned. On the other hand, the unwanted incident is represented by an "exploding" symbol, which could be linked to a fire, an explosion, an incident, etc. Therefore it can be classified as a symbol.

Principle of complexity management. According to Moody, the notation should include explicit mechanisms for dealing with complexity between the different diagrams, basically modularization and/or hierarchy. The goal of complexity management is to reduce the cognitive overload while reading or creating diagrams. For this principle, we assessed the presence or not of modularization and hierarchy. There is actually no way in CORAS to introduce hierarchy, but modularization is admitted by the High-Level CORAS extension and its referring and referred symbols which allows to expand events happening in any kind of scenario.

Principle of cognitive integration. Explicit mechanisms to support integration of information from different diagrams should be included. This principle only applies when multiple diagrams are used to represent a system (this is the case for CORAS) and is closely related to the principle of complexity management, when modularity is used. However, it can still apply if modularity is not used in order to integrate diagrams of different types. We assessed the principle of cognitive integration through the presence or absence of related mechanisms (see Table 3). Concerning perceptual integration (perceptual cues to simplify navigation and transitions between diagrams), CORAS includes no identification (labelling of diagrams), no navigational cues, and no navigational map. The text linked to the symbols is not considered here as a label but more as a description (for example, the text in a threat scenario describes the scenario itself, not the symbol that is a threat scenario). There are also no indication or rules about the level numbering. Regarding conceptual integration (mechanisms to help the reader assemble information from separate diagrams into a coherent mental representation of the system), we can find as 'summary diagram' the treatment overview diagram, but it is only focused on risk treatment aspects. Contextualization is considered absent from the CORAS language since we have not identified any, through symbols or diagrams mechanisms.

Table 3. Mechanisms to simplify navigation and transitions between diagrams.

Perceptual integration	**Definition**	**Presence/ Absence**
Identification	Labelling of diagrams	Absence
Level numbering	Orientation information	Absence
Navigational cues	Sign-posting	Absence
Navigational map	To show all diagrams and the navigation paths between them	Absence
Conceptual integration	**Definition**	**Presence/ Absence**
Summary diagram	Provides a view of the system as a whole	Presence
Contextualization	The part of a system of current interest is displayed in the context of the system as a whole	Absence

Principle of visual expressiveness. The full range and capacities of visual variables should be used. Visual variables are shape, size, colour, brightness, orientation, and texture for retinal variables, and horizontal and vertical position for planar variables. It is worth to note that perceptual discriminability and visual expressiveness are very close to each other but however different. The first aims at measuring the visual variation between constructs in pairs, and the second aims at measuring the visual variation across the entire visual vocabulary. Among the 17 graphical symbols, 9 different shapes (human shape, flag, warning triangle, oval shape, money bag, explosion, padlock symbol, arrowhead link, and square) were identified. Two different textures (full lines and fine strokes) and seven different colours (white, black, red, brown, yellow, green, light blue) are used. However, no kind of orientation are suggested. Furthermore, we could not find any semantic meaning for the vertical or horizontal position. As a conclusion, across the whole visual vocabulary, we identified 3 visual variables, as depicted in Table 4.

Table 4. Visual variables and associated variations.

Visual variable	Total of visual variations	Variations
Shape	9	Human shape, flag, warning triangle, oval shape, bag, explosion, padlock symbol, arrowhead link, square (high level CORAS gates)
Texture	2	Full lines, fine strokes
Brightness	-	
Size	-	
Colour	7	White, Black, red, brown, yellow, green (treatment), light blue (legal norm)
Orientation	-	
Horizontal position	-	
Vertical position	-	

Principle of dual coding. Dual coding relates to the use of text to complement graphics. In the CORAS language, text is present to describe what a symbol may represent, like a precision (describing a law or an article), but it doesn't refer to the associated construct (legal norm). The symbols and the notation of CORAS are purely graphic. Therefore, CORAS is not allowing dual coding.

Principle of graphic economy. The number of different graphical symbols should be cognitively manageable. The graphic complexity is defined by the number of different graphical conventions used in a notation (i.e. number of legend entries). CORAS is composed of 17 different graphical symbols, but not all of them are used in every diagram. The reference book of CORAS [5] clearly states what are the elements that are used in the five different diagrams. For example, an asset diagram can contain only stakeholders, (direct or indirect) assets, and harm relations. The maximum number of legend entries is thus 4 for asset diagram. We used this notion of maximum since the number of legend entries can differ from one model to another: some constructs allowed to be used in a given diagram may not be used in a specific instance of this kind of diagram, reducing thus the number of legend entries in this case. For example, in a threat diagram, identified threats can be only human, when no non-human threat have been identified. Moreover, we can include CORAS extensions (High-Level CORAS, Dependent CORAS, Legal CORAS) and their symbols in the different diagrams, increasing the maximum number of legend entries. The graphic economy scores for the five different types of diagrams in CORAS are depicted in Table 5.

Table 5. Graphic economy scores

Types of diagrams	Max number of legend entries
Asset diagram	4
Threat diagram	9
Risk diagram	5
Treatment diagram	12
Treatment overview diagram	7
High-level CORAS	+1
Dependent CORAS	+1
Legal CORAS	+2

Principle of cognitive fit. For a satisfying cognitive fit, a visual language is supposed to make a good use of different visual dialects for different tasks and audience. The aim is to come up with a good ability to communicate through peers for the language and to be usable for different situations. CORAS proposes one kind of dialect, with five types of diagrams, for more than one type of audience (are cited: analysis leader, analysis secretary, representatives of the customer with decisions makers, technical expertise and users) and eight different tasks to complete a risk assessment (see Table 6).

Table 6. Cognitive fit characteristics.

Cognitive fit elements	Occurrences
Number of dialects for this language	1
Number of different audiences	1+
Number of different tasks admitted	8
Other	5 types of diagrams

5 Discussion

As already argued earlier in this paper, although the PoN is one of the most elaborated approach to evaluate cognitive effectiveness of a modelling language, it is difficult, if not impossible, to establish clear-cut conclusions with this framework. Indeed, because of the lack of reference scales or explicit outcome to achieve associated to the principles of the PoN, it is not possible to firmly claim a language is considered as satisfactory (or not). However, the conclusions we draw from the results obtained after the evaluation of the cognitive effectiveness of CORAS are the following:

- The notation of CORAS seems not very difficult to apprehend by security risk managers as a larger number of symbols compared to signs are used, making the notation pretty transparent and, therefore, understandable. The semiotic clarity principle is largely fulfilled, as only four constructs are represented with the same graphical symbol, this aspect constituting the sole anomaly with regards to semiotic clarity. Furthermore, perceptual discriminability of the language is globally allowing users to visually differentiate and compare symbols, the visual distance between the different constructs of the language being 2 or more in more than 80% of the cases. The lack of dual coding and the use of additional visual variables (e.g., spatial position or size of constructs) are however ways for improvement;
- The use of the language to support a whole security risk assessment would suffer from the low support proposed by CORAS for cognitive integration, despite the presence of the High-level CORAS extension as complexity management mechanism. Considering the complexity of current IS associated to the large number of risks to manage (usually more than hundred based on our experience), it is clearly difficult to consider that CORAS would be efficient at this level;
- Regarding the graphic economy principle, CORAS seems cognitively manageable for most diagrams, that means individual models produced are usually manageable by our short-term memory. Indeed, Miller came up with the fact that our short-term memory can manage seven, plus or minus, two items [28]. As depicted in Table 5, asset diagram, threat diagram, risk diagram and treatment overview diagram have plus or minus seven legend entries in the general case (this result may however be modified when some allowed legend entries are not introduced or when some extensions are used). Nevertheless, memory span can differ depending on the hierarchical organisation of these items in chunks. Therefore, we cannot really draw a line here on what we consider as cognitively manageable or not. An experience on short-term memory management of visual items and chunk strategy would be necessary in order to put clearer limits in an appropriate context;
- The fact that there is actually only one kind of dialect (cognitive fit) could represent some sort of challenge for the various audiences addressed. Security risk managers are the target group of CORAS model designers, but CORAS model users in general are broader, ranging from security experts to people generally weakly skilled and aware on security aspects such as the top management of a company.

During our evaluation, we identified two main threats to validity of the results obtained. The first one is that the analysis performed remains subjective, because performed (only) by three evaluators who are the authors of this article. To reduce the biases coming from this aspect, we used as much as possible quantitative and verifiable metrics during our analysis. However, it is clear to us that most of the principles would benefit from an approach based on users, who would be involved in the evaluation of the language. This “user centric approach” could be performed by semi-structured interviews, focus groups composed of actual users, and, more specifically, user testing, where users are invited to do tasks, while their behaviours are observed to identify flaws of the language. The problems found with user testing are true problems in the sense that at least one user encountered each identified problem [29]. In contrast, the problems found with heuristic evaluation, as we made in this study, are potential problems: the evaluators suspect that something may be a problem to users.

A second threat to validity is that the discussion and conclusions are not based on some 'reference cognitive effectiveness scale' to which we could compare our quantitative results. Several quantitative analysis of modelling languages could lead us to establish such a calibration scale in order to better compare languages between them and get a benchmark of them for specific applications/contexts.

6 Conclusion

In this paper, we evaluated the cognitive effectiveness of the modelling language of CORAS, an approach for ISSRM. As evaluation framework, we used the PoN, a comprehensive set of principles based on a synthesis of theories from, e.g., the psychology and cognitive science fields, that can be used in order to analyse the cognitive effectiveness of existing visual notations, or aid the design of new ones. The need for such an evaluation comes from some drawbacks we observed in traditional ISSRM methods, especially the difficulty to have a clear and manageable documentation for ISSRM activities. Our insight is to introduce conceptual models as support of ISSRM activities and, in this context, the cognitive effectiveness of the produced models is considered as a must-have.

Based on the conclusion drawn during the evaluation performed, CORAS seems not very difficult to apprehend by security risk managers and individual models should generally be manageable by users. The main weaknesses identified are the difficulty to support a whole security risk assessment due to the weak inclusion of cognitive integration and complexity management mechanisms, and the absence of consideration of the different audiences that can be involved in a risk assessment. Compared to the RSO, the other ISSRM language we evaluated thanks to the PoN [4], CORAS can be considered from a general point of view as more cognitive effective, mainly because of a broader use of iconic shapes, a better semiotic transparency and a better perceptual discriminability.

Regarding future work, to complete the heuristic evaluation, we want to adopt a User Centred Design (UCD) approach, in order to take into account the capabilities of actual users, as well as their skills and cognitive limitations. To do that, we plan to apply some methods, like interviews and personas, which establish user needs and specifications. We also plan to use the experience map method, allowing us to draw and understand the walkthrough of the security risks managers when they assess the risks of an organisation. UCD should allow us to suggest recommendations and improvements aligned with actual needs of users, and to make decisions on the necessary trade-offs about our visual syntax, taking care of a specific context.

Acknowledgments. Supported by the National Research Fund, Luxembourg, and financed by the ENTRI project (C14/IS/8329158).

References

1. Kordy, B., Mauw, S., Radomirović, S., Schweitzer, P.: Foundations of Attack-defense Trees. In: Proceedings of the 7th International Conference on Formal Aspects of Security and Trust. pp. 80–95. Springer Berlin Heidelberg (2011).
2. Iver Band, Wilco Engelsman, Christophe Feltus, Sonia González Paredes, Jim Hietala, Henk Jonkers, Sebastien Massart: Modeling Enterprise Risk Management and Security with the ArchiMate® Language. The Open Group (2015).
3. Mayer, N., Grandry, E., Feltus, C., Goettelmann, E.: Towards the ENTRI Framework: Security Risk Management Enhanced by the Use of Enterprise Architectures. In: Persson, A. and Stirna, J. (eds.) Advanced Information Systems Engineering Workshops. pp. 459–469. Springer International Publishing (2015).
4. Mayer, N., Feltus, C.: Evaluation of the Risk and Security Overlay of Archimate to Model Information System Security Risks. In: 2017 IEEE 21st International Enterprise Distributed Object Computing Conference Workshops (EDOCW). pp. 106–116. IEEE (2017).
5. Lund, M.S., Solhaug, B., Stolen, K.: Model-Driven Risk Analysis: The CORAS Approach. Springer-Verlag Berlin and Heidelberg GmbH & Co. K, Berlin ; London ; New York (2010).
6. Hitchman, S.: Practitioner perceptions on the use of some semantic concepts in the entity–relationship model. European Journal of Information Systems. 4, 31–40 (1995).
7. Hitchman, S.: The Details of Conceptual Modelling Notations are Important - A Comparison of Relationship Normative Language. Communications of the Association for Information Systems. 9, 167–179 (2002).
8. Nordbotten J. C., Crosby M. E.: The effect of graphic style on data model interpretation. Information Systems Journal. 9, 139–155 (2001).
9. Shanks, G.: The challenges of strategic data planning in practice: an interpretive case study. The Journal of Strategic Information Systems. 6, 69–90 (1997).
10. Figl, K., Derntl, M., Rodriguez, M.C., Botturi, L.: Cognitive effectiveness of visual instructional design languages. Journal of Visual Languages & Computing. 21, 359–373 (2010).
11. Green, T.R.G., Petre, M.: Usability Analysis of Visual Programming Environments: A ‘Cognitive Dimensions’ Framework. Journal of Visual Languages & Computing. 7, 131–174 (1996).
12. Moody, D.: The “Physics” of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. IEEE Transactions on Software Engineering. 35, 756–779 (2009).
13. Moody, D., Hillegersberg, J. van: Evaluating the Visual Syntax of UML: An Analysis of the Cognitive Effectiveness of the UML Family of Diagrams. In: Gašević, D., Lämmel, R., and Wyk, E.V. (eds.) Software Language Engineering. pp. 16–34. Springer Berlin Heidelberg (2008).
14. Moody, D.L., Heymans, P., Matulevičius, R.: Visual syntax does matter: improving the cognitive effectiveness of the i* visual notation. Requirements Eng. 15, 141–175 (2010).

15. Genon, N., Heymans, P., Amyot, D.: Analysing the Cognitive Effectiveness of the BPMN 2.0 Visual Notation. In: Malloy, B., Staab, S., and Brand, M. van den (eds.) Software Language Engineering. pp. 377–396. Springer Berlin Heidelberg (2010).
16. Moody, D.L.: Review of ArchiMate: The Road to International Standardisation. ArchiMate Foundation and BiZZDesign B.V. (2007).
17. Beckers, K., Heisel, M., Solhaug, B., Stølen, K.: ISMS-CORAS: A Structured Method for Establishing an ISO 27001 Compliant Information Security Management System. In: Engineering Secure Future Internet Services and Systems. pp. 315–344. Springer, Cham (2014).
18. Linden, D. van der, Hadar, I.: A Systematic Literature Review of Applications of the Physics of Notation. IEEE Transactions on Software Engineering. PP, 1–1 (2018).
19. Störrle, H., Fish, A.: Towards an Operationalization of the "Physics of Notations" for the Analysis of Visual Languages. In: Model-Driven Engineering Languages and Systems. pp. 104–120. Springer, Berlin, Heidelberg (2013).
20. Linden, D. van der, Zamansky, A., Hadar, I.: How Cognitively Effective is a Visual Notation? On the Inherent Difficulty of Operationalizing the Physics of Notations. In: Enterprise, Business-Process and Information Systems Modeling. pp. 448–462. Springer, Cham (2016).
21. Krogstie, J.: Using a Semiotic Framework to Evaluate UML for the Development of Models of High Quality. In: Unified Modeling Language: Systems Analysis, Design and Development Issues. pp. 89–106. IGI Global (2001).
22. Genon, N.: Unlocking Diagram Understanding: Empowering End-Users for Semantically Transparent Visual Symbols, (2016).
23. Frank, U.: Domain-Specific Modeling Languages: Requirements Analysis and Design Guidelines. In: Reinhartz-Berger, I., Sturm, A., Clark, T., Cohen, S., and Bettin, J. (eds.) Domain Engineering. pp. 133–157. Springer Berlin Heidelberg (2013).
24. Guizzardi, G., Pires, L.F., Sinderen, M. van: Ontology-Based Evaluation and Design of Domain-Specific Visual Modeling Languages. In: Nilsson, A.G., Gustas, R., Wojtkowski, W., Wojtkowski, W.G., Wrycza, S., and Zupančič, J. (eds.) Advances in Information Systems Development. pp. 217–228. Springer US (2006).
25. Kleppe, A.: Software Language Engineering: Creating Domain-Specific Languages Using Metamodels. Addison-Wesley Professional (2008).
26. Nielsen, J., Molich, R.: Heuristic Evaluation of User Interfaces. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 249–256. ACM, New York, NY, USA (1990).
27. Zender, M., Mejía, G.M.: Improving Icon Design: Through Focus on the Role of Individual Symbols in the Construction of Meaning. Visible Language. 47, 66–89 (2013).
28. Miller, G.A.: The Magical Number Seven, Plus Or Minus 2: Some Limits On Our Capacity for Processing Information. Psychological review. 63, 81–97 (1956).
29. Lauesen, S., Pave Musgrove, M.: Heuristic Evaluation of User Interfaces versus Usability Testing. In: User Interface Design - A Software Engineering Perspective. pp. 443–463 (2005).