Evaluation of the Risk and Security Overlay of ArchiMate to model Information System Security Risks

Nicolas Mayer, Christophe Feltus
Luxembourg Institute of Science and Technology
5, avenue des Hauts-Fourneaux, L-4362 Esch-sur-Alzette, Luxembourg
{nicolas.mayer, christophe.feltus}@list.lu

Abstract—In nowadays complex and interconnected society, organizations are required to implement information system security as well as risk management. Nevertheless, in the fast moving and always more regulated environment in which we evolve, dealing with such requirements remains a challenging issue. In that regard, our previous works have consisted in considering the field of enterprise architecture to support Information System Security Risk Management (ISSRM) and more specifically the difficulty to have a clear and manageable documentation for this activity. The output of our research is currently an integrated model built on the mapping of concepts from both domains, allowing dealing with ISSRM using enterprise architecture paradigm. Our objective is now to suggest a visual syntax for this integrated model, deemed as necessary to support the practitioner to document the ISSRM steps. As a candidate for such a visual syntax, this paper analyses the "Risk and Security Overlay" of the ArchiMate language through two complementary aspects: completeness of the notation with regards to our integrated model and cognitive effectiveness with the nine related principles elaborated by Moody, also called "Physics of Notations".

Keywords—Security; Risk Management; Enterprise Architecture; Visual Syntax; Language Analysis; ArchiMate; Physics of Notations.

I. INTRODUCTION

Nowadays, Information System (IS) security and Risk Management (RM) are required for every organization that wishes to survive in this networked world. Whether for purely compliance purposes, business development opportunities, or even governance improvement, 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. In a nutshell, the main issues identified are:

- The complexity of current IS that are, in addition, continuously evolving [1], [2],
- The increasing number of risk-related regulations needed to be managed [3]–[5],
- The difficulty to have a clear and manageable documentation for ISSRM activities [6].

Enterprise Architecture Management (EAM) has shown to be a valuable and engaging instrument to face enterprise complexity and the necessary enterprise transformation [1], [7]. It offers means to govern enterprises and make informed decisions: description of an existing situation, investigation and expression of strategic direction, analysis of gaps, planning at the tactical and operational level, selection of solutions, and architecture design [8]. Our research agenda [9] aims at integrating EAM with ISSRM, to be able to deal with the preceding listed issues related to the complexity of organizations and associated risks.

The first step in that direction was the integration of ISSRM concepts with EAM concepts in a model called the "EAM-ISSRM integrated model" [10], [11]. The next step consists now in the definition of a modelling language (i.e. a graphical notation) to support this "EAM-ISSRM integrated model". Such a language will be used by practitioners to document the different steps of ISSRM and enhance decision-making all along this process [12]. It will thus contribute specifically to address the difficulty to have a clear and manageable documentation for ISSRM activities – the third issue identified above – by completing the EAM-ISSRM integrated model with a graphical notation, the latter being considered as more expressive and maintainable than the traditional table-based approach of ISSRM [6].

Instead of starting defining a new modelling language, we first want to assess existing one(s) in the literature. Amongst them, ArchiMate is a standardized modelling language developed by The Open Group, an industry consortium developing standards, to provide a uniform representation for diagrams that describe Enterprise Architecture (EA) [13]. In 2015, a White Paper aiming at providing guidelines to ArchiMate users on how to model enterprise risk and security with the ArchiMate language has been published [14]. The contribution of this White Paper has been called a "Risk and Security Overlay" (RSO) of the ArchiMate language.

The objective of our paper is to evaluate the support proposed by the RSO to represent graphically our EAM-ISSRM integrated model. This visual notation needs first to be able to express the concepts of the EAM-ISSRM integrated model. It is then

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

necessary to take into account the target group of our notation that is information security risk managers. This target group is not used to use modelling languages in its daily work, especially those related to EAM, and our language needs so to be effective to this target group to convey information, i.e. to be cognitive effective [15]. The evaluation of the RSO presented in this paper is thus performed at two different levels: completeness and cognitive effectiveness of the notation. First, we will assess the conceptual coverage of the RSO with regards to the EAM-ISSRM integrated model (does the RSO provide a complete coverage of the EAM-ISSRM integrated model?). Second, we will assess the cognitive effectiveness of the RSO (is the RSO cognitive effective 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 the EAM-ISSRM integrated model and the RSO proposed for ArchiMate. Section III is about the alignment of the EAM-ISSRM concepts with the components of ArchiMate suggested in the RSO in order to assess the completeness of the notation with regards to our conceptual model. Then, Section IV presents the assessment of the cognitive effectiveness of the RSO when modelling risk and security with ArchiMate. Section V discusses the related work. Finally, Section VI concludes about our research work and presents our future work.

II. BACKGROUND

A. The EAM-ISSRM integrated model

The EAM-ISSRM integrated model has been built by extending the ISSRM domain model [16], a conceptual model depicting the ISSRM domain, with EAM concepts. The core ISSRM domain model is composed of three groups of concepts (see Fig. 1):

Asset-related concepts (light grey concepts, i.e. the right part of Fig. 1) describe assets and the criteria which guarantee asset security. An asset is anything that has value to the organization and is necessary for achieving its objectives. A business asset consists of information, processes, capabilities, or skills inherent to the business and core mission of the organization, and that is of value for it. An IS asset is a component of the IS supporting business assets, such as for example a database where information is stored. As described in the ISSRM literature [6], an IS is a composition of hardware, software, network, people and facilities. A security criterion is a security property or constraint, such as for example confidentiality, integrity and availability. A security objective is the application of a security criterion to a business asset (for example, the confidentiality of personal information).

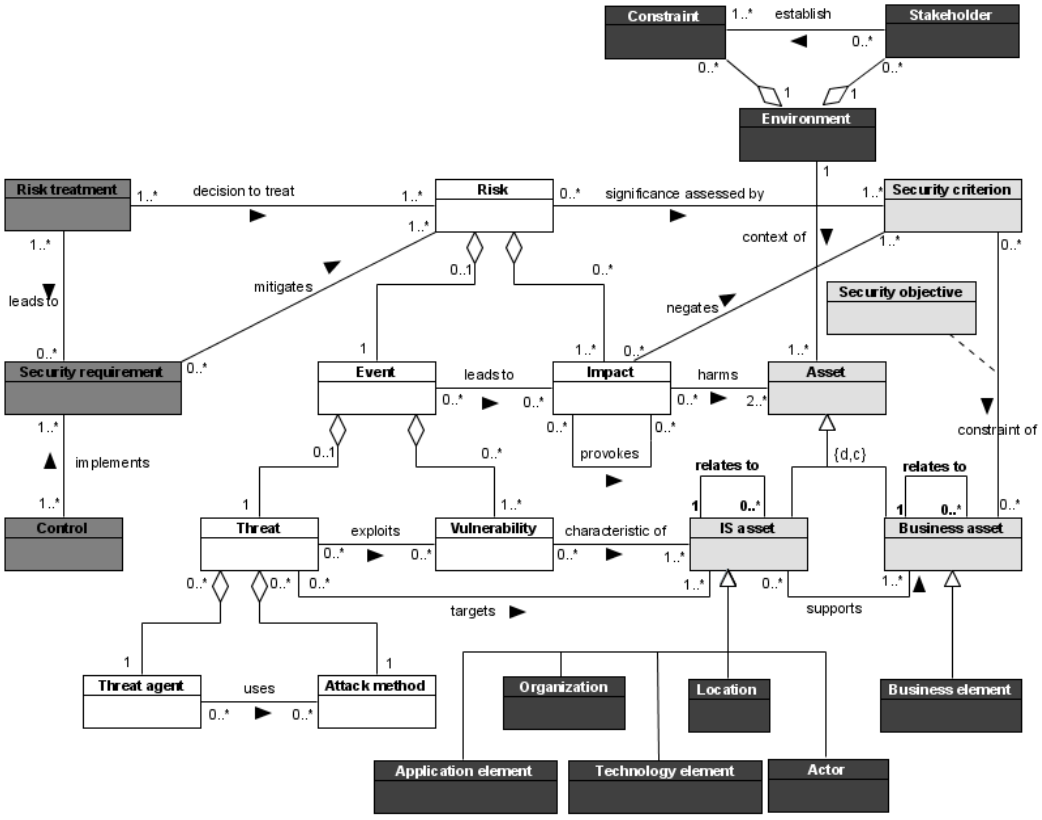

Fig. 1. The EAM-ISSRM integrated model

Risk-related concepts (white concepts, i.e. the middle part of Fig. 1) present how the risk itself is defined. A risk is the combination of an event with a negative impact harming the assets. An impact describes the potential negative consequence of an event that may harm assets of a system or organization, when an event causing this impact occurs. An event is the combination of a threat and one or more vulnerabilities. A vulnerability is a characteristic of an IS asset or group of IS assets that can constitute a weakness or a flaw that can be exploited by a threat. A threat is a potential attack or incident, which targets one or more IS assets and may lead to the assets being harmed. A threat consists of a threat agent and an attack method. A threat agent is an agent that can potentially cause harm to IS assets. An attack method is a standard means by which a threat agent carries out a threat.

Risk treatment-related concepts (dark grey concepts, i.e. the left part of Fig. 1) describe what decisions, requirements and controls should be defined and implemented in order to mitigate possible risks. A risk treatment is an intentional decision to treat identified risks, such as reducing risks through security requirements, sharing with another party the burden of loss from risks, etc. A security requirement is a desired property of an IS that contributes to a risk treatment. Controls (countermeasures or safeguards) are a designed means to improve security, specified by a security requirement, and implemented to comply with it.

The integration of EAM in the ISSRM domain model improves the latter with two extensions [10], [11]: the introduction of the environment of the assets and the refinement of business and IS assets with EAM elements. It is composed by the following concepts (black concepts with white names in Fig. 1):

First, *Environment* is defined as the set of concepts composing the ISSRM context of the assets. It is composed of *Constraint* and *Stakeholder*, a constraint being usually (but not necessarily, e.g., environmental constraints) established by a stakeholder. Then, business elements specific to EAM approaches are used to refine business assets. *Application element* and *Technology element* are specifications of IS assets, as well as *Organization*, *Location*, and *Actor*, despite being most often considered as business-related elements in EAM approaches. However, they are considered as IS assets from an ISSRM point of view, because they are part of the IS processing information, and thus potentially targeted by security threats or source of vulnerabilities, as depicted in Fig. 1

B. The Risk and Security Overlay of the ArchiMate language

The Risk and Security Overlay (RSO) of the ArchiMate language is described in a White Paper published in January 2015 and entitled "Modeling Enterprise Risk Management and Security with the ArchiMate Language" [14]. This White Paper, a joint project of The Open Group ArchiMate Forum and The Open Group Security Forum, is a guidance on how to model enterprise risks and security with ArchiMate. It is not a so-called 'extension' of ArchiMate, an extension of the language being defined as additional concepts coming with their own notation. The contribution of the white paper is called an 'overlay'. Here, no additional notation is provided to support the new concepts. However, guidelines on how to use (or extend the use of) existing notations to support the newly introduced concepts are the core of the document.

The overlay consists first in the introduction of 11 new concepts represented with existing notations (see Fig. 2):

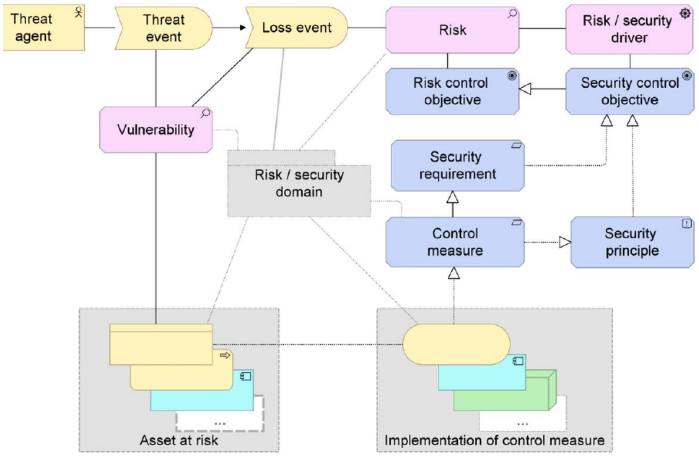

Fig. 2. The Risk and Security Overlay of the ArchiMate language (extracted from [14])

- *Threat agent* is modeled with the business actor construct, but other active structure element can be used (e.g. business role, application component, etc.)
- *Threat event* (event with the potential to adversely impact an asset) as well as *loss event* (any circumstance that causes a loss or damage to an asset) are modeled with the business event construct and are considered as a specialization of it.
- *Vulnerability* is modeled as a specialization of an assessment in the ArchiMate language, because considered as the result of analyzing the weaknesses of elements in the architecture.
- *Risk* is also modeled with the assessment construct and considered as a specialization of it.
- *Risk / security driver* are the criteria by which risks are analyzed and are represented with the driver construct.
- *Risk control objective* and *security control objective* are specialization of the objective construct and aim at mitigating risks. They are designed from risks and risk/security drivers and further refined through security requirement and principle.
- *Security requirement* and *control measure* are two specializations of the requirement construct, a control

measure being more specific (i.e. close to the implementation) than a security requirement.

- *Security principle* is considered here as a synonym of security policy and is modeled with the principle construct.

Moreover, the *Risk / security domain* concept has also been added, represented with the 'grouping' notation (considered as a relationship in ArchiMate), and defined as "consisting of entities that share one or more characteristics relevant to risk management or security". Finally, to represent 'Asset at risk' and 'Implementation of control measure', the core concepts of ArchiMate can be used. Further details about the RSO and its components can be found in the dedicated White Paper [14].

III. Alignment of the EAM-ISSRM concepts with the components of ArchiMate suggested in the RSO

The first step of our evaluation of the RSO as a notation for the EAM-ISSRM integrated model is performed at the level of completeness, i.e. the coverage that the RSO has with respect to EAM-ISSRM integrated model concepts. To do this, we first perform a mapping between the concepts of the EAM-ISSRM integrated model (first column of Table I) and the concepts of the RSO (second column of Table I). Concepts of the EAM-ISSRM integrated model are split into two classes:

- Asset-related, risk-related and risk treatment-related concepts (white cells in first column of Table I) are in the scope of the RSO and thus a mapping is proposed with concepts of the EAM-ISSRM integrated model, when relevant.
- Other concepts (grey cells in Table I) are concepts coming from the EAM extension of the ISSRM domain model and can thus directly be represented with ArchiMate core constructs. They are not in the specific scope of the RSO. ArchiMate constructs used to represent these concepts are already provided in the EAM-ISSRM integrated model (see Section II. A) and are reported in Table I (third column in Table I).

The mapping is performed based on the definitions provided in the EAM-ISSRM integrated model [10], [11], [16] and the ones provided in the RSO [14]. For most of the concepts, this mapping is obvious and direct because the definitions are very close one to the other, and, regularly, the concepts share even the same name (e.g., risk, threat, vulnerability, threat agent, etc.). However, it is worth to note that the summary of the RSO depicted in Fig. 2 is not complete, and sometimes further information is given in the White Paper [14], especially:

- Threat can be modeled as a driver instead of modeling threat agent and threat event as it sub-components
- Sometimes the same concept of the RSO is called with different names within the document. We gathered these synonyms and mentioned them as part of the RSO concepts (second column of Table I).

Second, for each concept of the RSO (second column of Table I), we report in the third column of Table I how the White Paper suggests modeling it with constructs from ArchiMate. As a consequence, we did not perform any alignment decision in this task, but we just report on the suggestions provided in the White Paper to represent RSO concepts. For some concepts, it is recommended (but not required, as also seen in the examples reported in the White Paper) to use a specialization of the traditional ArchiMate concept. In the provided examples, a specialization of a concept is represented as a <<stereotype>>-notation, as in UML [17]. In the above case, we explicitly mention it in Table I (i.e. "Specialization of *Concept*").

TABLE I. Alignment of the EAM-ISSRM concepts with the constructs of ArchiMate suggested in the RSO

EAM-ISSRM integrated model		**Risk and Security Overlay of the ArchiMate language** [14]	**Constructs from ArchiMate 2.1** [13]
Asset-related concepts	Asset	Asset at Risk	Any core concept or combination of concepts
	Business Asset		
	IS Asset		
	Security criterion	Risk / security driver	Driver
	Security objective	Security control objective	Goal
	Organization	N/A	Any core concept from the application layer, technology layer, or combination of them
	Location	N/A	Location
	Application element	N/A	Any core concept from the application layer

	Technology element	N/A	Any core concept from the technology layer
	Actor	N/A	Actor
	Business element	N/A	Any core concept from the business layer
	Environment	N/A	Principle, Stakeholder, or combination of them
	Constraint	N/A	Principle
	Stakeholder	N/A	Stakeholder
Risk-related concepts	Risk	Risk	(Specialization of an) Assessment
	Impact	Loss Event	(Specialization of a) Business event
	Event	N/A	N/A
	Threat	Threat	Driver
	Vulnerability	Vulnerability	(Specialization of an) Assessment Attribute of an asset at risk or a risk domain
	Threat agent	Threat agent	Active structure elements (e.g., business actor, business role, application component, node, system software, or device)
	Attack method	Threat event	(Specialization of a) Business event
Risk treatment-related concepts	Risk treatment	Risk control objective	Goal
	Security requirements	Security requirement	Requirement
		Control measure	
	Control	Implementation of control measure (also called Risk control, treatment and mitigation)	Any core concept or combination of core concepts A grouping of a number of core concepts
		Security principle (also called Policy)	(Specialization of) Principle

As a conclusion of the alignment, the coverage of the EAM-ISSRM integrated model by the RSO is complete apart from "Event", concept not included in the RSO. However, we consider this lack as negligible because an event is defined in the EAM-ISSRM integrated model as being (only) the composition of threat and vulnerability. Thus, modeling a threat (i.e. a threat agent performing a threat event) and its associated vulnerability(ies) is strictly equivalent as modelling an event.

IV. Assessment of the Cognitive Effectiveness of the RSO when Modelling Risk and Security in ArchiMate

The second step of our evaluation of the RSO as a notation for the EAM-ISSRM integrated model is performed at the level of cognitive effectiveness [15], i.e. the effectiveness of our language to convey information to a target group composed of ISSRM practitioners, such a target group being usually not used to use EAM modelling languages like ArchiMate.

Our assessment of the cognitive effectiveness of the RSO is based on the work of Moody that has established the foundation for a science of visual notation design called "The Physics of Notations" [15]. Moody has defined a set of nine principles for designing "cognitive effective visual notations". These principles were constructed based on theory and empirical evidences about cognitive effectiveness of visual representation. They constitute what Moody calls the prescriptive theory for visual notation and allow shifting from unselfconscious into a subconscious process of visual notation design. Provided that Moody's principles have already been used for evaluating the ArchiMate visual syntax [18], we have decided to use them for evaluating the RSO of ArchiMate.

Hereafter, we analyze the cognitive effectiveness of the RSO through the nine principles elaborated by Moody. We first remind a short definition for each principle. Then, we report on the analysis of the cognitive effectiveness of ArchiMate performed by Moody in his technical report entitled "Review of ArchiMate: The Road to International Standardisation" [18]. To report on the analysis of ArchiMate is of great help to support the analysis of the RSO, the latter exploiting existing constructs of

ArchiMate for modelling risk and security. Finally, we suggest an analysis of the RSO that extends the review of ArchiMate.

A. *Principle of semiotic clarity*

Definition: There should be a 1:1 correspondence between semantic constructs and graphical symbols.

ArchiMate: ArchiMate does not achieve the semiotic clarity principle because of two issues:

Symbol redundancy: Symbol redundancy occurs when multiple graphical symbols can be used to represent the same semantic construct [15]. Moody has identified 10 instances of symbol redundancy in ArchiMate. Moreover, relationships can also be graphically represented in multiple ways: for example inheritance can be shown via nesting or explicit links [18].

Symbol overload: Symbol overload occurs when the same graphical convention can be used to represent different types of semantic constructs. In ArchiMate, the same graphical convention (spatial enclosure) can be used to show a range of different relationship types: inheritance, assignment, aggregation, composition, and grouping [15].

RSO: In the RSO, all of the new concepts are represented using existing graphical notations, i.e. already used to represent other concepts. For example, the risk concept is defined as a specialization of *Assessment*, however the symbol used to represent a risk is identical to the one used to represent an assessment, although their meaning is different. Another example is that a blue hexagon is used to represent a set of concepts from the RSO (security requirement, control objective, security principle, etc.) while it is already used to represent four concepts of the motivational extension of ArchiMate (goal, requirement, constraint, and principle). As a conclusion, the RSO adds 11 instances of symbol overload. No additional symbol redundancy (i.e. excluding those inherited from the core ArchiMate) has been noticed.

Another issue related to semiotic clarity is "Symbol deficit". It occurs when there are semantic constructs that are not represented by any graphical symbol. As concluded in Section III, the coverage of the EAM-ISSRM integrated model by the RSO is complete apart from *Event*, concept not included in the RSO. The RSO adds thus one symbol deficit if we consider that each concept of the integrated model is worth to be represented (that is an arguable statement as concluded in Section III).

B. *Principle of perceptual discriminability*

Definition: 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, color, position, etc.) [15].

ArchiMate: In ArchiMate, two issues have been highlighted with regards to perceptual discriminability. First, a large perceptual distance often exists between different symbols to represent the same construct [18]. For example, a business role may be represented by a rectangle or by a cylinder, and a business process may be represented by a rounded rectangle or by an arrow shape. Secondly, there are small perceptual distances between symbols used to represent different constructs [18]. For instance: both the business actor and the system software are represented by a rectangle, although there is an additional small symbol in the upper right corner to distinguish both of them. Nevertheless, the most perceptible shape is the rectangle (or, in general, the global shape of the construct) and not the small symbol in the upper right corner, leading to a weak discriminability.

RSO: In the RSO, four shapes and three colors are used to represent 11 concepts. More specifically, when combining shapes and colors, four different symbols are used (yellow rectangle, yellow rounded arrow, purple hexagon, and blue hexagon) to represent the 11 introduced concepts. Like in ArchiMate, small symbols are also used in the upper right corner of the symbols to distinguish them (e.g., a magnifying glass to identify the *Assessment* construct used to represent Vulnerability and Risk, a helm to identify the *Driver* construct used to represent Risk / security driver, etc.).

C. *Principle of semantic transparency*

Definition: Visual representations whose appearance suggests their meaning should be used. In other words, the meaning of a symbol should be understood by looking at its representation.

ArchiMate: In ArchiMate, only one iconic shape (i.e. symbol which perceptually resemble the object it represents) is provided: the alternative representation of a business actor that is a stickman [18]. Other constructs are represented using "neutral" shapes or even confusing ones, like, e.g., the cylinder shape that represents *Business role* but which is usually associated to a database. ArchiMate uses spatial enclosure but the latter may have different meanings (e.g., inheritance, assignment, aggregation, etc.) and, as a result, loses its value.

RSO: In the RSO, no iconic shape is introduced. Existing "neutral" shapes of ArchiMate are reused.

D. *Principle of complexity management*

Definition: Explicit mechanisms for dealing with complexity should be included, such as modularization (divide large models into smaller ones linked together) or hierarchy (use of levels of abstraction).

ArchiMate: ArchiMate provides only one way of managing complexity: layering. This is done by grouping constructs of the same type within a higher level construct or using the grouping construct. However, it does not remove the constraint of showing everything on a single diagram. Views help to manage complexity by focusing on specific aspects of an architecture model. Moreover, hierarchical structuring can be performed at the level of diagrams, when separating in a hierarchical manner

the different components of each layer (business, application and technology) when drawing a model [18].

RSO: The RSO introduces the 'risk / security domain', used to group together the new concepts. However, despite the introduction of the 'risk / security domain', it is difficult to comply with the following rule: 7 ± 2 "bubbles" per diagram (consistent with the known limits of working memory and recommended by Moody in this principle [15]) with 11 new concepts that can be used and should be integrated with 'traditional' ArchiMate models, as shown in the examples provided [14]. No guidelines are provided in the sense that modularity shall be used when modelling risk and security with the RSO. The RSO could have been enhanced with, e.g., the introduction of specific views. Hierarchy could also have been introduced, for example with a top down analysis such as Asset – Risk – Risk treatment, but no recommendations of this kind have been done and the examples provided represent the new concepts on a single diagram.

E. Principle of cognitive integration

Definition: 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 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.

ArchiMate: ArchiMate addresses the cognitive integration using the concept of view. A view is defined in ArchiMate as "*part of an architecture description that addresses a set of related concerns and is addressed to a set of stakeholders*". Moreover, "*views are specified by viewpoints. Viewpoints define abstractions on the set of models representing the enterprise architecture, each aimed at a particular type of stakeholder and addressing a particular set of concerns.*" ArchiMate proposes 16 different views (potentially completed with views designed by the user itself) which, according to Moody [18], creates enormous problems of cognitive integration. Moody observes indeed a lot of overlaps between the standard views as well as high levels of graphic complexity (see the principle of *Graphic economy*). As a conclusion, the mechanisms introduced in ArchiMate do not simplify the system effectively and sometimes make it more complex.

RSO: The RSO does not suggest any additional viewpoint. However, the user is free to define yet another view related, e.g., to risk and security management, that could contribute in strengthening cognitive integration issues.

F. Principle of visual expressiveness

Definition: The full range and capacities of visual variables should be used. Visual variables are shape, size, color, 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 symbols and the second aims at measuring the visual variation across the entire visual vocabulary. As a result, the same variables are potentially considered in the analysis of both principles.

ArchiMate: Among the eight visual variables available, ArchiMate uses a wide range of them (4 variables are used), and outperforms most IT diagramming notations on this criterion [18]. A suggestion to improve visual expressiveness of ArchiMate is to use in addition spatial location to encode information.

RSO: The RSO reuses the symbols used in ArchiMate, and the introduction of additional ones has been considered as unnecessary. The visual expressiveness of the RSO is thus equivalent to the one of ArchiMate.

G. Principle of dual coding

Definition: Text should be used to complement graphics.

ArchiMate: ArchiMate does not support annotations (i.e. textual explanations which may be included in the diagrams). Some ArchiMate tools for drawing diagrams allow making annotations, but annotations are not defined at the language level. Labels may be associated to most concepts, but this is mainly a diagram level consideration.

RSO: In the examples provided to illustrate the RSO, specific and detailed labels are used to name the diagram elements. Stereotypes are furthermore used to specify ArchiMate concepts in a risk and security context.

H. Principle of graphic economy

Definition: 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) and should thus be limited.

ArchiMate: In ArchiMate, the graphic complexity of most views exceeds the "*span of absolute judgment*" or the ability to discriminate between perceptually distinct alternative which is around six categories. Some views such as *Actor cooperation*, *Business process* or *Service realization* have a graphic complexity of more than 20 (i.e. more than 20 different symbols can be used in each of these views). Among the 16 standard ArchiMate views, only one (*Period view*) has a graphic complexity below 6.

RSO: In the RSO, no additional view has been introduced. Thus, if the additional concepts are integrated in existing views, it increases the graphic complexity. The definition of a risk and security view with a limited number of entries could respect the principle of graphic economy, but is considered as out of scope in this paper because nor introduced nor discussed in the RSO.

I. Principle of cognitive fit

Definition: Different visual dialects should be used for different tasks and audiences.

ArchiMate: The ArchiMate language exhibits a visual monolinguism in that it does not provide specific language for specific audience. Whatsoever, it is worth to say that some concepts of the framework may be expressed using two options:

- A dedicated shape for the concepts in the color of the layer it is associated to, e.g. the arrow for the process, the cylinder for the business role, etc.
- A rectangle, in the color of the layer it is associated to, with the shape corresponding to the concepts in small and in the upper right corner.

As far as we know, ArchiMate does not to provide any explanation regarding the use of these two options.

RSO: In the RSO, the new risk and security concepts that are introduced are mapped to existing visual notations. A contextualization of the language for specific audience is not provided.

J. Conclusions on the Cognitive Effectiveness Assessment

In the evaluation of the ArchiMate visual notation performed by Moody [18], a set of positive aspects have been noticed in the language (e.g., the use of a wide range of visual variables regarding visual expressiveness). However, 9 negative aspects have been noticed, coming with a set of 36 recommendations to improve the cognitive effectiveness of the language. The conclusion drawn is that the visual notation of ArchiMate has some flaws and that it cannot be considered as perceptually and cognitively optimal.

After having discussed the cognitive effectiveness of the RSO through each of the nine principles elaborated by Moody, the same conclusion can be drawn for it because, although the RSO is better than ArchiMate on one principle, it is strictly equivalent or worse for the eight others. In more details:

- The RSO is better than ArchiMate on one principle: dual coding. Indeed, specific and detailed labels are used as well as stereotypes, used to specify ArchiMate concepts in a risk and security context.
- The RSO is strictly equivalent to ArchiMate on four principles: semantic transparency, cognitive integration, visual expressiveness, and cognitive fit. Modelling with the RSO does not add or modify anything in relation to these principles with regards to the proper use of ArchiMate.
- The RSO has a negative impact on four principles: semiotic clarity, perceptual discriminability, complexity management, and graphic economy. For these principles, the RSO basically inherits from the negative aspects of ArchiMate and aggravates them by adding new concepts coming with the same weaknesses.

Regarding threats to validity of this assessment, three of them have been identified:

1. The analysis performed for the RSO is subjective, because performed (only) by the authors of this article.

To reduce the biases coming from this aspect, the approach taken here is intentionally based on the conclusions drawn by Moody during his analysis of ArchiMate. Thus, instead of elaborating conclusions from scratch on the cognitive effectiveness of the RSO, we decided to derive them from the analysis performed by Moody on ArchiMate, this analysis considered as reliable, Moody having elaborated the nine reference principles. However, we acknowledge that some subjectivity remains even if we used such an approach.

2. The analysis performed for the RSO is subjective, because it relies only on qualitative statements and not on any quantitative analysis that would be best suited to provide clear-cut conclusions on the cognitive effectiveness level of the RSO.

Although some research works have been initiated to formalize and operationalize the use of Moody's principles [19], [20], it remains non-mature proposals focused on small samples of modelling languages. Thus, the use of such a qualitative analysis has been considered as the most suited approach to analyze the RSO as a whole.

3. The analysis performed is based on ArchiMate 2.1 and not on ArchiMate 3.0 that is the last published and up-to-date version of the ArchiMate standard.

The RSO being built on top of ArchiMate 2.1, it was necessary to use it as the reference for our analysis. It would be interesting to integrate in the analysis the potential improvements in terms of cognitive effectiveness adopted in ArchiMate 3.0 (e.g., the introduction of a new spatial enclosure named *Grouping* and whose objective is to "aggregate or compose concepts that belong together based on some common characteristic" should improve *semiotic clarity* and *complexity management*). However, it is worth to note than nothing specific to risk and security has been introduced in ArchiMate 3.0 and that the RSO remains thus the only dedicated proposal in our scope of interest.

V. Related work

This section reviews first the visual syntaxes that have already been evaluated on the basis of Moody's principles and second, the other approaches that can be used for such evaluations.

A. Other languages already evaluated with Moody's principles

Moody's principles have already been experienced for analyzing a set of languages and visual notations amongst which, e.g., UML, *i**, SEAM, BPMN and Misuse Cases. Although the

UML language semantics has benefit of many researches and refinements, the visual aspect and the visual notations of this language was not a central concern for the software engineering community. Acknowledging this weakness, thirteen UML diagram types were evaluated using Moody's theory [21]. The conclusion of the evaluation revealed that radical changes are required to improve and to make effective most of the UML visual notations, especially regarding the class diagram which has been considered by the authors as the worst of all. In [22], a symbol-by-symbol systematic analysis of *i** visual syntax was achieved and has revealed important shortcomings of this goal-oriented language. Weaknesses were observed for example at the level of the perceptually discriminability criterion given the use of shapes from different shape families, or at the level of the semantic transparency criterion, due to the difficulty, for a novice user, to refer to shapes and colors without the appropriate explanation. In the field of EA models, Systemic Enterprise Architecture Methodology (SEAM) is a tool aiming at analyzing and designing business and IT systems involving various stakeholders. Popescu and Wegmann [23] appraise the visual notation of this methodology using Moody's principles. The result of SEAM's visual notation assessment is that six over nine principles need to be improved, amongst which especially: the semantic transparency, the visual expressiveness, the semiotic clarity and the graphic economy. Interestingly, Popescu and Wegmann also propose an analysis of Moody's theory [15] and highlight some deficiencies like, e.g., the fact that no distinction between various contexts (business and IT models) is possible, that theory only focuses on the final models and the process of creating them is not taken into account, and finally that some principles sometimes overlap each other. The Business Decision Modeling (BDM) is a conceptual model for specifying the data structures necessary to represent a business model, as well as for designing the operational or informational systems. The seven first principles of the visual notation theory were exploited to examine and to enhance the visual representation of business rules and business decisions set forth by this approach [24]. Genon et al. analyzed BPMN 2.0 process modelling notations [25]. Therefore, the nine principles were applied to carry out a symbol-by-symbol systemic analysis of the process diagram. The analysis highlights in addition a lack of consideration for scientific principles of notation design. Likewise, Genon et al. also analyze the strengths and weaknesses of Use Case Maps (UCM) visual notation and accordingly suggest few ideas for improvements [26]. Based on their experiences, the authors propose some observations associated with the use of Moody's principles, amongst which, the fact that many principles represent conflicting goals, that the deployment of the approach is time consuming and that training is essential to correctly apprehend it. In the area of specifying functional security requirements, Saleh et al. [27] present an evaluation of the design of the original Misuse Case modeling notation and suggest a set of improvements including the usage of colors and icons which improve the readability of Misuse Case diagrams.

B. Other approaches to evaluate visual syntax

Many frameworks exist to support visual syntax assessment. The SEQUAL framework, developed by Krogstie et al. [28], 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 [29]. Frank addresses the visual notation design in the context of the Domain-Specific Modeling Language (DSML) which represents the concepts and constraints of a well-defined domain-level knowledge [30]. Seven processes compose Frank's process model, respectively: clarification of scope and purpose, analysis of generic requirements, analysis of specific requirements, language selection, design of graphical notation, development of a modeling tool, and evaluation and refinement. Regarding the design of graphical notations, the guidelines provided are built on the author's own experience and on the respective literature analysis, are: built semantic categories of concepts, create generic symbols for each category, graphical difference between symbols is a function of the semantic difference (that guideline corresponds to the visual distance advocated by Moody), prefer icons to shapes, combine shape, color and text effectively, avoid symbol overload, avoid redundant symbols, represent monotonic semantic features through compositions of symbols and finally, a graphical notation should include symbols that allow for reducing diagram complexity. However, the approach developed by Frank aim at being applied during the design of the graphical notation. Guizzardi et al. propose an ontology-based assessment and design method of domain-specific visual modeling languages [31]. 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 [32]. She addresses the problem through the lens of a clear description of the concepts in relation to the problem domain, as established through our EAM-ISSRM integrated model. In [33], Selic argues that the profile mechanisms of UML provide a powerful capability to define new DSMLs and proposes a method for defining profiles which disjoins the DSML definition from the profile definition, achieving thereby an important separation of concerns. In that regard, Giachetti et al. [34] argue that several model driven development approaches have defined DSML for representing specific semantics and therefore have tried to use UML as DSML, by using UML profiles. Provided that no UML profile generation process seems to exist in the literature, the authors proposed a process that integrates a DSML into UML through the automatic generation of a UML profile. Finally, several tools exist to support the development of DSML. Amyot et al. [35] evaluated five of them (GME, Tau G2, RSA, XMFMosaic, and Eclipse with GEF and EMF) under the perspective of how well they can be used to generate graphical editors for the Goal-oriented Requirement Language (GRL) and following well defined criteria, i.e. graphical completeness,

usability, development effort, handling of language evolution, integration with other languages, and analysis capabilities. Once again, these approaches are developed to be used during design time of a modelling language and are less suited to the evaluation of an existing visual notation.

VI. Conclusion and Future Work

In this paper, we evaluated the support proposed by the RSO to represent graphically our EAM-ISSRM integrated model. The evaluation of the RSO visual notation has been done at two different levels: completeness with regards to the EAM-ISSRM integrated model (Section III) and cognitive effectiveness, relying on the nine principles established by Moody [15], [18] (Section IV). Regarding completeness, the coverage of the EAM-ISSRM integrated model by the RSO is complete apart from "Event". As discussed in Section III, this lack is negligible and we can consider the RSO as an appropriate notation to support the EAM-ISSRM integrated model from a completeness point of view. Regarding cognitive effectiveness, many gaps have been identified with regards to the nine principle established by Moody. Although no quantitative analysis has been performed to objectify this conclusion, the RSO can decently not be considered as an appropriate notation from a cognitive effectiveness point of view and there is room to propose a notation better on this aspect.

This paper is focused on assessing the RSO without suggesting improvements based on the conclusions drawn. As a consequence, our objective for future work is to propose a more cognitive effective visual notation for the EAM-ISSRM integrated model. The approach currently considered is to operationalize Moody's principles into concrete metrics and requirements, taking into account the needs and profile of the target group of our notation (information security risk managers) through personas development [36] and user experience map [37]. With such an approach, we will be able to make decisions on the necessary trade-offs about our visual syntax, taking care of a specific context. We also aim at validating our proposal(s) with the help of tools and approaches extracted from cognitive psychology research applied to HCI domain (e.g., eye tracking, heuristic evaluation, user experience evaluation…).

References

[1] J. A. Zachman, "A framework for information systems architecture," *IBM Syst. J.*, vol. 26, no. 3, pp. 276–292, 1987.

[2] H. A. Proper, "Enterprise Architecture - Informed steering of enterprises in motion," in *Proceedings of the 15th International Conference on Enterprise Information Systems (ICEIS)*, 2013.

[3] Official Journal of the European Union, *Directive 2009/140/EC of the European Parliament and of the Council of 25 November 2009*. 2009.

[4] CSSF, "Circulaire CSSF 12/544 - Optimisation par une approche par les risques de la surveillance exercée sur les 'PSF de support,'" 2012.

[5] ISO/IEC 27001:2013, *Information technology – Security techniques – Information security management systems – Requirements*. Geneva: International Organization for Standardization, 2013.

[6] N. Mayer, "Model-based Management of Information System Security Risk," University of Namur, 2009.

[7] P. Saha, Ed., *A Systemic Perspective to Managing Complexity with Enterprise Architecture:* IGI Global, 2013.

[8] M. O. 't Land, E. Proper, M. Waage, J. Cloo, and C. Steghuis, "Positioning Enterprise Architecture," in *Enterprise Architecture*, Springer Berlin Heidelberg, 2009, pp. 25–47.

[9] N. Mayer, E. Grandry, C. Feltus, and E. Goettelmann, "Towards the ENTRI Framework: Security Risk Management enhanced by the use of Enterprise Architectures," in *Advanced Information Systems Engineering Workshops*, Springer International Publishing, 2015.

[10] N. Mayer, J. Aubert, E. Grandry, and C. Feltus, "An Integrated Conceptual Model for Information System Security Risk Management and Enterprise Architecture Management Based on TOGAF," in *The Practice of Enterprise Modeling : 9th IFIP WG 8.1. Working Conference, PoEM 2016, Skövde, Sweden*, Springer International Publishing, 2016, pp. 353–361.

[11] N. Mayer, J. Aubert, E. Grandry, C. Feltus, E. Goettelmann, and R. J. Wieringa, "An Integrated Conceptual Model for Information System Security Risk Management supported by Enterprise Architecture Management," To be published.

[12] ISO/IEC 27005:2011, *Information technology – Security techniques – Information security risk management*. Geneva: International Organization for Standardization, 2011.

[13] The Open Group, "ArchiMate® 2.1 Specification," Open Group Standard (C13L), Dec. 2013.

[14] Iver Band *et al.*, "Modeling Enterprise Risk Management and Security with the ArchiMate Language," The Open Group, White Paper, Jan. 2015.

[15] 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.

[16] E. Dubois, P. Heymans, N. Mayer, and R. Matulevičius, "A Systematic Approach to Define the Domain of Information System Security Risk Management," in *Intentional Perspectives on Information Systems Engineering*, S. Nurcan, C. Salinesi, C. Souveyet, and J. Ralyté, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 289–306.

[17] M. Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2003.

[18] D. L. Moody, "Review of ArchiMate: The Road to International Standardisation," ArchiMate Foundation and BiZZDesign B.V., Technical Report, 2007.

[19] H. Störrle and A. Fish, "Towards an Operationalization of the 'Physics of Notations' for the Analysis of Visual Languages," in *Model-Driven Engineering Languages and Systems*, 2013, pp. 104–120.

[20] D. van der Linden, A. Zamansky, and I. Hadar, "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*, 2016, pp. 448–462.

[21] D. Moody and J. van Hillegersberg, "Evaluating the Visual Syntax of UML: An Analysis of the Cognitive Effectiveness of the UML Family of Diagrams," in *Software Language Engineering*, D. Gašević, R. Lämmel, and E. V. Wyk, Eds. Springer Berlin Heidelberg, 2008, pp. 16–34.

[22] D. L. Moody, P. Heymans, and R. Matulevičius, "Visual syntax does matter: improving the cognitive effectiveness of the i* visual notation," *Requir. Eng.*, vol. 15, no. 2, pp. 141–175, May 2010.

[23] G. Popescu and A. Wegmann, "Using the Physics of Notations Theory to Evaluate the Visual Notation of SEAM," in *2014 IEEE 16th Conference on Business Informatics*, 2014, vol. 2, pp. 166–173.

[24] J. C. Thomas, J. Diament, J. Martino, and R. K. E. Bellamy, "Using the 'Physics' of notations to analyze a visual representation of business decision modeling," in *2012 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2012, pp. 41–44.

[25] N. Genon, P. Heymans, and D. Amyot, "Analysing the Cognitive Effectiveness of the BPMN 2.0 Visual Notation," in *Software Language Engineering*, B. Malloy, S. Staab, and M. van den Brand, Eds. Springer Berlin Heidelberg, 2010, pp. 377–396.

[26] N. Genon, D. Amyot, and P. Heymans, "Analysing the Cognitive Effectiveness of the UCM Visual Notation," in *System Analysis and Modeling: About Models*, F. A. Kraemer and P. Herrmann, Eds. Springer Berlin Heidelberg, 2010, pp. 221–240.

[27] F. Saleh and M. El-Attar, "A Scientific Evaluation of the Misuse Case Diagrams Visual Syntax," *Inf Softw Technol*, vol. 66, no. C, pp. 73–96, Oct. 2015.

[28] J. Krogstie, "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*, IGI Global, 2001, pp. 89--106.

[29] N. Genon, "Unlocking Diagram Understanding: Empowering End-Users for Semantically Transparent Visual Symbols," University of Namur, 2016.

[30] U. Frank, "Domain-Specific Modeling Languages: Requirements Analysis and Design Guidelines," in *Domain Engineering*, I. Reinhartz-Berger, A. Sturm, T. Clark, S. Cohen, and J. Bettin, Eds. Springer Berlin Heidelberg, 2013, pp. 133–157.

[31] G. Guizzardi, L. F. Pires, and M. van Sinderen, "Ontology-Based Evaluation and Design of Domain-Specific Visual Modeling Languages," in *Advances in Information Systems Development*, A. G. Nilsson, R. Gustas, W. Wojtkowski, W. G. Wojtkowski, S. Wrycza, and J. Zupančič, Eds. Springer US, 2006, pp. 217–228.

[32] A. Kleppe, *Software Language Engineering: Creating Domain-Specific Languages Using Metamodels*, 1st ed. Addison-Wesley Professional, 2008.

[33] B. Selic, "A Systematic Approach to Domain-Specific Language Design Using UML," in *Proceedings of the 10th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing*, Washington, DC, USA, 2007, pp. 2–9.

[34] G. Giachetti, B. Marín, and O. Pastor, "Using UML as a Domain-Specific Modeling Language: A Proposal for Automatic Generation of UML Profiles," in *Advanced Information Systems Engineering*, P. van Eck, J. Gordijn, and R. Wieringa, Eds. Springer Berlin Heidelberg, 2009, pp. 110–124.

[35] D. Amyot, H. Farah, and J.-F. Roy, "Evaluation of Development Tools for Domain-Specific Modeling Languages," in *System Analysis and Modeling: Language Profiles*, R. Gotzhein and R. Reed, Eds. Springer Berlin Heidelberg, 2006, pp. 183–197.

[36] T. Miaskiewicz and K. A. Kozar, "Personas and user-centered design: How can personas benefit product design processes?," *Des. Stud.*, vol. 32, no. 5, pp. 417–430, Sep. 2011.

[37] G. Brugnoli, "Connecting the dots of user experience," *J. Inf. Archit.*, vol. 1, no. 1, pp. 6–15, 2009.