Ontological Analysis and Redesign of Security Modeling in ArchiMate[⋆]

Ítalo Oliveira[1], Tiago Prince Sales[1], João Paulo A. Almeida[2], Riccardo Baratella[1],
Mattia Fumagalli[1], and Giancarlo Guizzardi[1,3]

[1] Conceptual and Cognitive Modeling Research Group (CORE),
Free University of Bozen-Bolzano, Bolzano, Italy
{idasilvaoliveira, tprincesales, rbaratella, mfumagalli,
gguizzardi}@unibz.it

[2] Ontology and Conceptual Modeling Research Group (NEMO),
Federal University of Espírito Santo, Vitória, Brazil
jpalmeida@ieee.org

[3] Services & Cybersecurity Group, University of Twente, The Netherlands

Abstract. Enterprise Risk Management and security have become a fundamental part of Enterprise Architecture, so several frameworks and modeling languages have been designed to support the activities associated with these areas. ArchiMate's Risk and Security Overlay is one of such proposals, endorsed by The Open Group. We investigate the capabilities of the proposed security-related constructs in ArchiMate with regard to the necessities of enterprise security modeling. Our analysis relies on a well-founded reference ontology of security to uncover ambiguity, missing modeling elements, and other deficiencies of the security modeling capabilities in ArchiMate. Based on this ontologically-founded analysis, we propose a redesign of security aspects of ArchiMate to overcome its original limitations.

Keywords: Security Modeling · Enterprise Architecture · ArchiMate · Ontological Analysis · Unified Foundational Ontology.

1 Introduction

Enterprise architecture refers to principles, methods, and models that are used in the design and implementation of an enterprise's organizational structure, business processes, information systems, and infrastructure [7]. Risks are pervasive throughout the activities of any enterprise, so it is important to create security mechanisms to control certain risks that are particularly threatening to an organization's objectives. Enterprise Risk Management is exactly about this process of identification, evaluation, treatment, and communication regarding these risks, as described by ISO 31000, an International Standard for Risk Management [6]. The TOGAF® Series Guide to "Integrating Risk and Security within a TOGAF Enterprise Architecture" [16] states that the Security Architecture is a cross-cutting matter, ubiquitous throughout the entire Enterprise Architecture. It is understood as a coherent collection of views, viewpoints, and artifacts,

[⋆] Work supported by Accenture Israel Cybersecurity Labs.

including security, privacy, and operational risk perspectives, along with related topics like security objectives and security services. The Security Architecture affects and informs the Business, Data, Application, and Technology Architectures [16]. Because of that, Enterprise Risk Management has, naturally, become a key aspect of Enterprise Architecture, as seen by the *Risk and Security Overlay* (RSO) of ArchiMate [1], an attempt to introduce risk and security concepts into ArchiMate language – the Open Group's open and independent conceptual modeling language for Enterprise Architecture [15].

Though the RSO is based on risk and security frameworks (COSO, ISO, TOGAF, and SABES) [1], it has already been shown to have some limitations concerning its conceptualization of risk concepts [12], including ambiguity and missing modeling elements that negatively impact the capabilities of the RSO to support enterprise risk and security modeling. Through an ontological analysis founded upon the *Unified Foundational Ontology* (UFO) [4] and the *Common Ontology of Value and Risk* (COVER) [11], researchers have shown, for instance, a *construct overload* on the Vulnerability construct, which collapses actual vulnerabilities with assessments about them, and a *construct deficit* to represent Threat Capabilities [12]. Based on the results of this analysis, an ontologically well-founded redesign of RSO was proposed to overcome the identified problems in the risk-related elements [12]. Here, employing a similar methodology of ontological analysis (tracing back to [4, 10]), we further investigate the modeling capabilities of the *security* elements of RSO: the notions of Control Objective, Security Requirement, Security Principle, Control Measure, and Implemented Control Measure. Our analysis is grounded in the *Reference Ontology of Security Engineering* (ROSE) [9], which is a UFO-based core ontology for safety and security; particularly, ROSE provides an elucidation of the notion of security mechanism. Based on the ontological analysis of security modeling constructs of RSO with ROSE, we propose a redesign of the concerned language fragment, taking advantage of the improved risk-related elements by the previous work [12].

The remainder of this paper is structured as follows: in Section 2, we provide an overview of the RSO focusing on security elements. In the same section, we present the redesigned version of the RSO with respect to risk elements, which will be the starting point of our own proposal. In Section 3, we briefly present our ontological foundations regarding value, risk, and security, which serves as the conceptual basis for the analysis in Section 4. The results of the analysis are used to redesign the RSO in Section 5. We conclude with a discussion on related work in Section 6 and final remarks in Section 7.

2 Security Modeling in Archimate

We present here the current security modeling constructs proposed for the ArchiMate language as part of the Risk and Security Overlay [1] along with the redesigned risk elements that resulted from the ontological analysis in [12].

2.1 The Original ArchiMate Risk and Security Overlay

The most updated version of the RSO was developed by a joint project of The Open Group ArchiMate Forum and The Open Group Security Forum [1], accommodating changes to the ArchiMate language in Version 3.1 of the standard. The RSO was designed through ArchiMate language customization mechanisms, in particular the specialization of both ArchiMate Core and Motivation and Strategy elements, and additional risk and security-specific attributes [1].

Table 1 summarizes the security elements according to the specialized ArchiMate elements. A Control Objective (or security objective) is a desired state of security, a high-level goal that should be realized by a Security Requirement (or control requirement), which is, during the risk analysis process, a specification of an action or set of actions that should be executed or that must be implemented as part of the control, treatment, and mitigation of a particular risk [1]. A Control Objective is associated with risk assessment, so that, for example, if the risk of workplace accident is assessed as unacceptable, then the organization could decide to reduce it as its Control Objective. To achieve this goal, the organization could define that its employees should wear personal protective equipment as Security Requirement. RSO proposes the representation of Control Objective and Security Requirement as specializations of the ArchiMate constructs of Goal and Requirement, respectively.

Table 1. Summary of security elements in ArchiMate's Risk and Security Overlay (RSO)

RSO Element	**ArchiMate Element**
Control Objective	Goal
Security Requirement	Requirement
Security Principle	Principle
(Required) Control Measure	Requirement
Implemented Control Measure	Core Element

The notion of Security Principle is less developed in the RSO white paper [1]. A Principle in ArchiMate represents a statement of intent defining a general property that applies to any system in a certain context in the architecture [15]. Similarly to requirements, principles defines the intended properties of systems. But principles are wider in scope and more abstract than requirements. For example, the principle "Information management processes comply with all relevant laws, policies, and regulations" is realized by the requirements that are imposed by the actual laws, policies, and regulations that apply to the specific system under design [15]. A Security Principle is related to the notion of policy and ArchiMate Motivation elements, though the RSO offers neither an explicit definition of it nor its usage in an example. The white paper also notes that the ArchiMate language does not have the concept of operational policy [1].

According to the RSO, a required Control Measure, also called risk control, treatment or mitigation, specializes Security Requirement, and it is a proposed action, device, procedure, or technique that reduces a threat, a vulnerability, or an attack by eliminating or preventing it, by minimizing the harm it can cause, or by discovering and reporting it so that corrective action can be taken [1]. An Implemented Control Measure is the deployment of a Control Measure. Depending on the kind of control, almost any core concept or combination of core elements of ArchiMate can be used to model the implementation of a Control Measure. A Control Measure may also be realized by a grouping of a set of core elements as its implementation [1].

Figure 1 summarizes how RSO proposes to represent risk and security elements in ArchiMate [1]. An Implemented Control Measure is associated with an Asset at Risk, which can be a Resource or a core element of ArchiMate. An Implemented Control Measure influences negatively a Vulnerability as an Assessment, in the sense that it makes the emergence of a Threat Event and the consequent Loss Event associated with that Vulnerability less probable.

To exemplify how the RSO can be used, we present two examples extracted from [1], highlighting the assumptions that the white paper calls "common characteristics shared by entities in risk management domains". The examples refer to the case of the Coldhard Steel company, illustrating the stereotyping of ArchiMate Motivation elements as risk elements. Figure 2 represents the risk of losing production due to machine failure. A power supply assembly is an Asset at Risk that fails when the power fluctuates (a Threat Event). This power assembly failure causes the failure of other machines, characterizing a loss for the organization (a Loss Event), associated to the Risk of production loss. Then, the Control Objective is defined as an adequate peak power supply capacity, which means that the organization seeks to reduce this risk, which should be done by the Control Measure of replacing the power supply assembly. By this example, we notice some of the aforementioned characteristics: the asset is exposed to a threat or a risk due to its vulnerability, but, at the same time, the asset posses a control requirement and, indeed, participates in the realization of its own Control Measure.

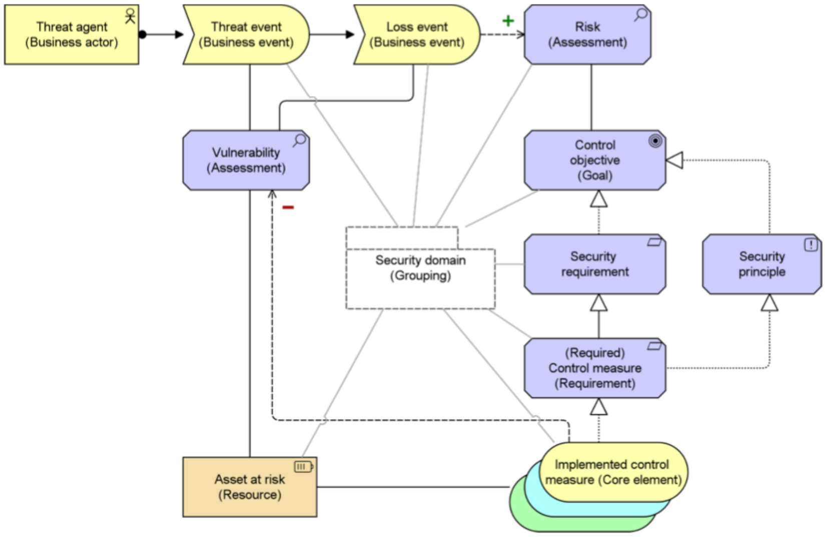
Threat agent (Business actor)
Threat event (Business event)
Loss event (Business event)
+
Risk (Assessment)
Vulnerability (Assessment)
−
Control objective (Goal)
Security domain (Grouping)
Security requirement
Security principle
(Required) Control measure (Requirement)
Asset at risk (Resource)
Implemented control measure (Core element)

Fig. 1. Mapping of Risk and Security Elements to the ArchiMate language [1]

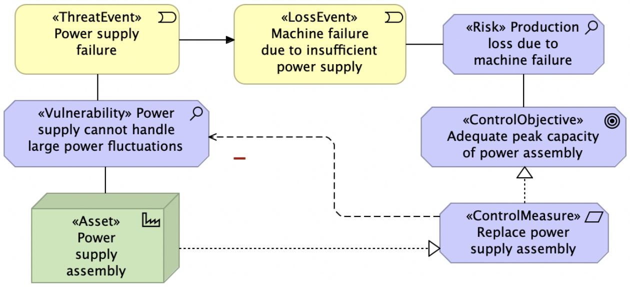
«ThreatEvent» Power supply failure
«LossEvent» Machine failure due to insufficient power supply
«Risk» Production loss due to machine failure
«Vulnerability» Power supply cannot handle large power fluctuations
«ControlObjective» Adequate peak capacity of power assembly
−
«Asset» Power supply assembly
«ControlMeasure» Replace power supply assembly

Fig. 2. Example from the case of the Coldhard Steel company [1]

The second example (Figure 3) illustrates a risk mitigation approach – continuous improvement of machine reliability – applied across the entire Coldhard Steel risk management domain. The implementation of control measures is grouped by Risk Mitigation Domain, aimed at negatively influencing the vulnerability of inadequate power supply. This implementation involves several core elements of ArchiMate, such as Contract, Outcome, Business Process, and Equipment.

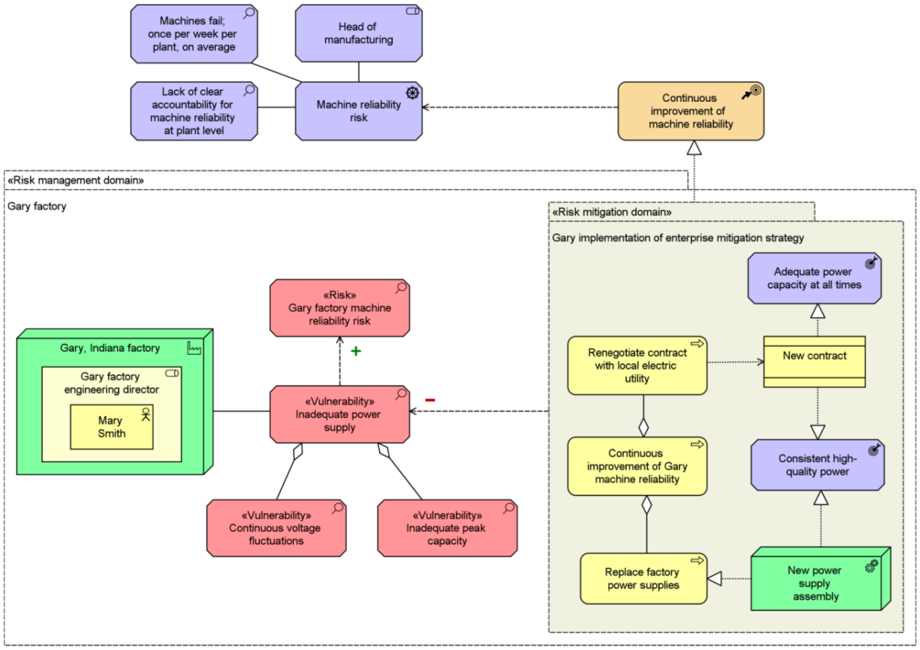
Machines fail; once per week per plant, on average
Head of manufacturing
Lack of clear accountability for machine reliability at plant level
Machine reliability risk
Continuous improvement of machine reliability
«Risk management domain»
Gary factory
«Risk mitigation domain»
Gary implementation of enterprise mitigation strategy
Adequate power capacity at all times
«Risk» Gary factory machine reliability risk
+
Gary, Indiana factory
Gary factory engineering director
Mary Smith
Renegotiate contract with local electric utility
New contract
«Vulnerability» Inadequate power supply
−
Continuous improvement of Gary machine reliability
Consistent high-quality power
«Vulnerability» Continuous voltage fluctuations
«Vulnerability» Inadequate peak capacity
Replace factory power supplies
New power supply assembly

Fig. 3. Mitigation of Machine Failure Risk at Coldhard Steel Gary Factory [1]

2.2 Redesigned Risk Elements of ArchiMate based on COVER

In [12], the authors performed an ontological analysis of the risk aspects of the RSO based on the Common Ontology of Value and Risk (COVER), proposing a redesign of part of the RSO to address the limitations identified by the analysis. Figure 4 shows the proposal of [12] for evolving the RSO, while Table 2 shows the full representation of risk concepts in ArchiMate based on COVER. This representation will be the basis of our own proposal concerning security aspects of ArchiMate.

A Hazard Assessment, proposed to represent UFO situations that activate threat capabilities, is an identified state of affairs that increases the likelihood of a Threat Event and, consequently, of a Loss Event. The occurrence of these events depends on the vulnerabilities of an Asset at Risk or of a Threat Enabler and the (threat) capabilities involving Threat Agent. All of this forms the Risk Experience of a Risk Subject, whose intention or Goal is harmed by a Loss Event. This experience may be assessed by a Risk Assessor (who may be the same subject as the Risk Subject) through a Risk Assessment (e.g., that determines that the Risk is unacceptable).

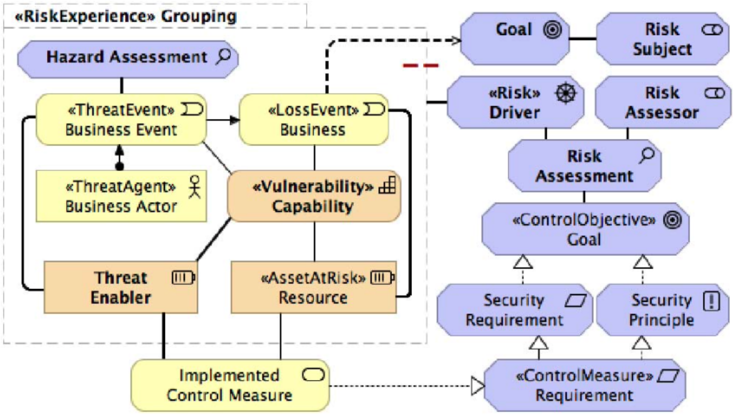

Fig. 4. Proposal of [12] for evolving the Risk and Security Overlay.

3 Ontological Foundations of Security

The Reference Ontology of Security Engineering (ROSE) [9] describes the general entities and relations of the security and safety domain, making use of an adapted version of COVER to capture value and risk domain[4]. ROSE understands the domain of security as the *intersection* between the domain of value and risk, understood under the terms of COVER [11], and the theory of prevention [2], which describes, in UFO's

[4] Files related to ROSE can be found in the following public repository: https://github.com/unibz-core/security-ontology.

Table 2. Representation of risk concepts in ArchiMate based on COVER [12]

COVER Concept	**Representation in ArchiMate**
Vulnerability	Capability stereotyped with «Vulnerability»
Threat Object	Structure Element stereotyped with «ThreatAgent»
Threat Event	Event stereotyped with «ThreatEvent»
Hazard Assessment	Assessment stereotyped with «HazardAssessment»
Loss Event	Event stereotyped with «LossEvent»
Intention	Goal
Risk Subject	Stakeholder associated with a Goal that is negatively impacted by a «LossEvent»
Object at Risk	Structure Element stereotyped with «AssetAtRisk»
Threat Enabler	Structure Element associated with a «ThreatEvent» or a «LossEvent»
Risk Experience	Grouping stereotyped with «RiskExperience»
Risk	Driver stereotyped with «Risk»
Risk Assessment	Assessment associated with a «Risk»
Risk Assessor	Stakeholder associated with a Risk Assessment

terms, how certain types of event are prevented or interrupted due to the occurrence of other events of certain types. From this perspective, security mechanisms create value by protecting certain goals from risk events. In COVER, Value is a relational property that emerges from the relations between capacities of certain objects and the goals of an agent. The manifestations of these capacities are events that bring about a situation (a state of affairs) that impacts or satisfies the goal of a given agent – the goal is understood as the propositional content of an intention [5]. Risk is the anti-value: risk events are the manifestations of capacities, vulnerabilities, and, sometimes, intentions that inhere in an agent; these events bring about a situation that hurts the goal of a given agent. Just like value, security is also a relational property that emerges from the relations between the capabilities of objects and the goals of an agent; however, manifestations of these capabilities bring about a situation that impacts the goal of an agent in a very specific way: preventing risk events [9]. Using the prevention theory described in [2], ROSE understands that Threat Capability, Vulnerability, and, sometimes, Intention are dispositions associated with types whose instances maintain a *mutual activation partnership* to each other. This means that a Threat Object can only manifest its Threat Capability if a Vulnerability can be exploited; if the Threat Object creates an Attack (an action, a kind of event), then the Intention is also required. Analogously, a Vulnerability is only manifested in the presence of a Threat Capability. From a security point of view, the importance of this *generic dependence* relation among these entities is that it determines some ways by which security measures can work: the removal of any of them from the situation that could activate them all together implies the prevention of the associated Risk Event. Figure 5 represents the risk aspects of ROSE in OntoUML language; this part is basically an adaptation and extension of COVER, clearly showing the dependence between intentions, capabilities, and vulnerabilities.

A Security Mechanism is always designed by an Agent called the Security Designer to be a *countermeasure to* events of certain type (Risk Event Type) [2, 9]. When an object is made to be a countermeasure to certain types of events, it aggregates capabilities whose manifestations ultimately prevent these events in a systematic fashion. The Agent creating a Security Mechanism is not necessarily the one protected by its proper functioning, i.e., the Protected Subject. Nonetheless, both agents have Intentions that are positively impacted by this proper functioning. For example, the government designs policies for public safety, the functioning of such policies satisfies some goal the government had when it designed them, but also satisfies the goal of people who want to be safe. Sometimes, the Protected Subject is the same Agent as the Security Designer, such as when a person places an electric fence surrounding her own house. A Security Mechanism is an object, which may be a simple physical object like a wall, a high-tech air defense system like the Israeli Iron Dome, an Agent like a policeman, a social entity like a security standard or anti-COVID-19 rules, that bears capabilities called Control Capability. The manifestation of this kind of capability is a Protection Event, specialized in Control Chain Event and Control Event, where the former can cause the latter. The Control Event is of a type (Control Event Type) that prevents, directly or indirectly, events of certain type (Risk Event Type). This is so because the control events bring about a Controlled Situation, which is of a type that is *incompatible with* the situations of the type that trigger risk events of certain types. Since risk events are specialized in Threat Event and Loss Event, the Controlled Situation Type is incompatible with the Threatening Situation Type or with Loss Triggering Situation Type. Figure 6 shows this ontological unpacking of the notion of Security Mechanism [9].

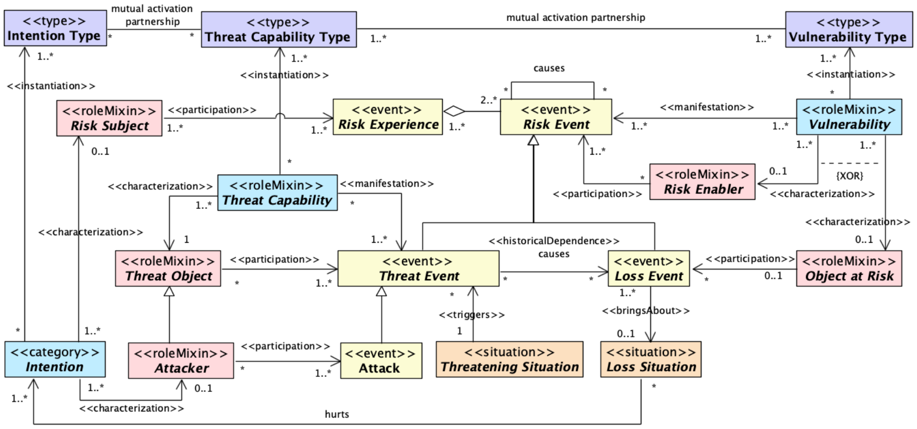

Fig. 5. Risk Aspects of ROSE [9]. The colors used signal the corresponding UFO categories: object types are represented in pink, intrinsic aspect types in light blue, situation types in orange, event types in yellow, higher-order types in a darker blue

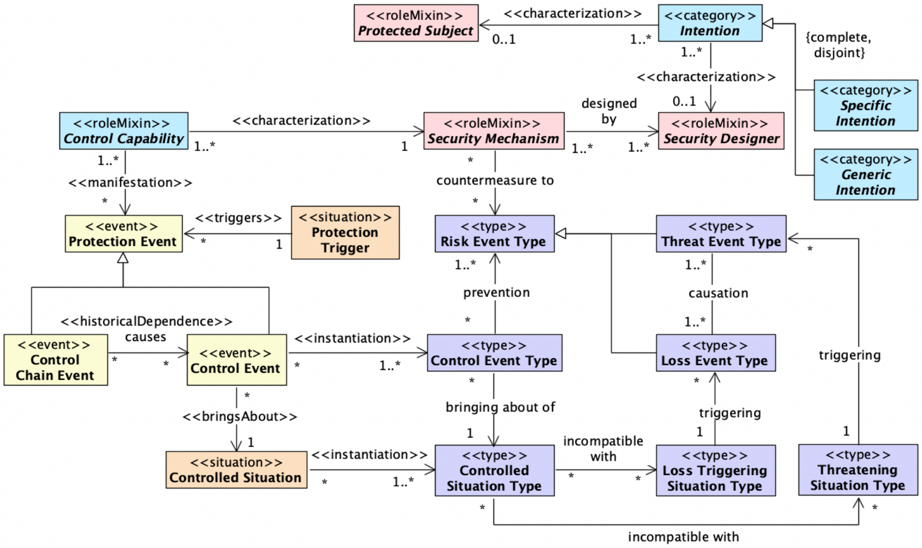

Fig. 6. Security Mechanism in ROSE [9]

4 Ontologically-founded Analysis of Security Modeling

ArchiMate is a modeling language for Enterprise Architecture. The RSO enriches ArchiMate with risk and security elements to support Enterprise Risk Management and security. It is known that one of the key success factors behind the use of a modeling language is its ability to provide its target users with a set of modeling primitives that can directly express important domain abstractions [4]. In other words, the more the grammar of a domain-specific modeling language corresponds to the ontology of the domain, the more capable the language is of modeling domain scenarios accurately. An ontological analysis is "the evaluation of a modeling grammar, from the viewpoint of a predefined and well-established ontology" [10], which is, in our case, ROSE [9] concerning the security domain. Ideally, according to Rosemann *et al.* [10], the modeling grammars should be isomorphic to their underlying ontology, that is, the interpretation from the modeling constructs to the ontology concepts should be bijective. This is a desirable characteristic because it prevents certain types of issues that affect the modeling capability of the language: (a) *ontological incompleteness* (or *construct deficit*), which is the lack of a grammatical construct for an existing ontological concept; (b) *construct overload*, which occurs when one grammatical construct represents more than one ontological construct; (c) *construct redundancy*, which happens when more than one grammatical construct represents the same ontological construct; (d) *construct excess*, when there is a grammatical construct that does not map to any ontological construct [10]. With the support of this framework, we identify shortcomings concerning the security modeling capability of the RSO.

Redundant Intentions and Lack of Clarity. The notions of Control Objective, Security Requirement, Control Measure, and Security Principle, all reflect a desired state of affairs that guides the actions of some agent. As we interpret the RSO, there are two relevant aspects among these distinctions: (1) a distinction between an end and a means to this end; that is the meaning behind, for example, the statement that a Security Requirement (a means) realizes a Control Objective (an end); and (2) the generality and abstractness of these intentions, in the sense that, for example, Control Objective is more general than Control Measure; concerning this generality and abstractness, it is not clear where Security Principle should be placed, since in Figure 1 Security Principle realizes Control Objective, though the documentation of ArchiMate suggests Principle has a higher level of generality and abstraction, which means the realization relation should be the inverse. The white paper [1] does not provide an example employing Security Principle or even Security Requirement, making use solely of Control Objective, Control Measure, and implemented control measure. Furthermore, no distinction is made regarding how Control Measure specializes Security Requirement. The means-end distinction is relational: an end targeted by a means may be a means to another end. For example, protecting the technical infrastructure from damage may be an end targeted by control measures, but it may also be a means to achieve mandatory legal requirements. Because of all that, those distinct notions of the RSO seem to be a case of construct redundancy, since different security modeling constructs represent the same ontological concept. The redundant constructs (particularly, Security Requirement and Security Principle) do not seem to play any practical role in security modeling[1]. We refer to this as *Limitation L1*.

Underspecification of Implemented Control Measures. An Implemented Control Measure can be any ArchiMate core element or multiple core elements grouped in a cluster, as seen in Figure 3. This would look like a construct overload, since a single construct collapses the object, its capability, and the event that is the manifestation of this capability. However, it is actually a strategy of representation via a supertype, so it is not an ontological problem by itself. The issue relies on the fact that this strategy offers no guidance to the modeler on what the implementation of a control measure should look like. In other words, the device of Implemented Control Measure is too generic and suffers from underspecification. In contrast, ROSE unfolds the notion of security mechanism in a general pattern that distinctively shows the difference between objects (Protected Subject, Security Designer, Security Mechanism), their modes and capabilities (Intention, Control Capability), the associated events and situations. The lack of this richness of the domain may be better classified as a construct deficit. This is aggravated by the assumption that the asset itself realizes its own control measure (see Figure 2), suggesting a confusion between the Object at Risk and elements of the pattern of Security Mechanism. We term this issue *Limitation L2*.

Lack of Distinction Between Baseline Architecture and Target Architecture. The implementation and migration concepts of ArchiMate are used to describe how an

[1] Actually, we can wonder whether the distinction of several of ArchiMate's Motivation Elements is (or not) redundant, such as goal, outcome, requirement, and principle, but this issue is outside the scope of our paper.

architecture will be realized over time through changes [7], providing the means to represent a baseline and a target architecture. The existence of these concepts in ArchiMate is justified by the importance of accounting for changes in the process of evolution of an enterprise. The introduction of a security mechanism is one of these changes. However, the RSO does not make use of this characteristic of ArchiMate, simply showing that security entities have a negative influence on Vulnerability. The redesigned RSO (see Figure 4) connects Implemented Control Measure to Threat Enabler and Asset at Risk, in order to express the impact on the threat event or the loss event. Still, no account of change is provided, as it would be expected from the capabilities of ArchiMate language by the means of constructs showing different plateaus from the baseline architecture to the target architecture. We call this lack of use of temporal aspects of ArchiMate *Limitation L3*.

Modeling the Subjects in the Security Domain. ROSE highlights there is a subject whose Intention is positively impacted by the effects of a Security Mechanism, the Protected Subject. Considering the risk domain, it is clear that this subject must be a proper subtype of the Risk Subject, which appears in the redesigned version of the RSO, as seen in Figure 4. In addition, another subject has not only his or her intentions positively impacted by the effects of a security mechanism, but is also responsible for the creation or introduction of the mechanism – often due to legal or contractual reasons, such as when someone is hired to install an electric fence. This is what ROSE calls the Security Designer. Sometimes the Protected Subject and the Security Designer are the same individual, while sometimes this is not the case. The original RSO presents none of that, whereas these subjects are not part of the scope of the redesigned version of the RSO. In summary, a case of construct deficit. We call this *Limitation L4*.

Triggering Conditions of Protection Events. The manifestation of the capability of a Security Mechanism occurs due to a Protection Trigger, a certain state of affairs that activates that capability. This represents environmental conditions that affect the manifestation of a Control Capability. For instance, a circuit breaker manifests its capability of interrupting a current flow when a fault condition is detected (heating or magnetic effects of electric current). In the redesigned RSO, there is an analogous notion for Threat Event, a threatening circumstance mapped as an assessment called Hazard Assessment [12]. They are particular configurations of the world that allow or increase the probability of the occurrence of a Threat Event. The advantage of explicitly accounting for the situations that trigger the Protection Event is that we can represent how several environmental factors increase the effectiveness of the Security Mechanism, assuming its effectiveness is directly connected to how likely it works properly, manifesting the Protection Event. This whole dimension is neglected by the RSO, a case of construct deficit – *Limitation L5*.

Interdependence Relation Among Risk Capabilities. As shown by ROSE, in its risk aspects (Figure 5), the manifestations of threat capabilities, vulnerabilities and, sometimes, intentions depend on the presence of each other. From this perspective, for example, it makes no sense to say that there is an ongoing threat without the simultaneous participation of a vulnerability. More importantly, from the security modeling point of view, recognizing this generic dependence relation among these entities allows for different strategies of protection or mitigation, since the removal of any of these capabilities or intentions would result in the prevention of the threat or loss event. Again, this dimension is not considered by the RSO, which refers to the efficacy of the control measure as simply influencing negatively a vulnerability. Doing so, the RSO says nothing about the multiple patterns of prevention uncovered by ROSE. Therefore, a case of construct deficit, *Limitation L6*.

5 Redesigning the Security Elements of ArchiMate

To address the identified shortcomings of security modeling in ArchiMate, we now propose a redesign of the security-related portion of the RSO, which also follows its original strategy of only using existing ArchiMate constructs. Since *L1* concerns a case of construct redundancy, we retain only the required constructs. So we retain Control Objective as a goal and Control Measure as a required means to achieve this goal. Considering this distinction from ROSE's perspective, we can conclude that the former is associated with a Protected Subject, while the latter is associated with a Security Designer, the one responsible for introducing the Security Mechanism. For example, a company has a Control Objective of protecting customer's data from cyberattacks. Based on an assessment, a series of Control Measures should be implemented by the company's cybersecurity team, playing the role of Security Designer; both the company and the customers may be regarded as Protected Subjects, since they have assets at risk that should be protected.

L4 is the absence of these two subjects, so we propose to introduce them, respectively, as a Stakeholder and a Business Role. The Protected Subject specializes Risk Subject, though some Risk Subjects might not be Protected Subjects due to lack of protection. Similarly, *L6* is the absence of a dependence relation among Threat Capabilities, Vulnerabilities, and Intentions (Goal in ArchiMate), a limitation that is easily solved by adding ArchiMate's associations among these entities. To address *L5*, the introduction of ROSE's concept of Protection Trigger follows the previous work [12], which uses Assessment to represent Threatening (or Hazardous) Situations. So Protection Trigger becomes Control Assessment.

Limitations L2 and *L3* are treated together: the baseline architecture reflects the state of the organization before the implementation of a security mechanism, and the target architecture shows the impact of the implementation of the security mechanism. At baseline, following a proposal for a pattern language for value modeling in ArchiMate [13], there is a Likelihood associated with the causal emergence of a Threat Event and a Loss Event. The dependence relations among risk entities are also shown, so that it should be clear that interfering in one of them would affect the likelihood of happening events like these. This is exactly what a Security Mechanism does in a systematic fashion, following the ROSE and the theory of prevention [2, 9][5]. But the implementation of a Security Mechanism is carried out by a Security Designer through the Work Package device of ArchiMate's migration layer, oriented by an

[5] Naturally, employing the theory of prevention in ArchiMate requires adaptation, considering ArchiMate does not distinguish the instance level from the type level.

identified gap in the baseline architecture. Once a Security Mechanism is implemented, the target architecture may show a different configuration of the risk entities that are interdependent, as well as a decreased likelihood concerning the emergence of a Threat Event or a Loss Event. Because of that, Risk Assessment may also be different, maybe evaluating the risk is now acceptable. Similarly, the required Control Measure might change. The pattern of Security Mechanism from ROSE is translated in ArchiMate as a Structure Element that holds a capability whose manifestation is an event that negatively influences the likelihood of Threat Event or a Loss Event. This pattern follows the value modeling pattern in ArchiMate proposed by [13], since security is a matter of specific creation of value through the prevention of risks. Figure 7 shows our proposal to evolve the security aspects of the RSO, highlighting in bold the constructs and relations we propose. Table 3 shows our proposal of the representation of security concepts in ArchiMate based on ROSE.

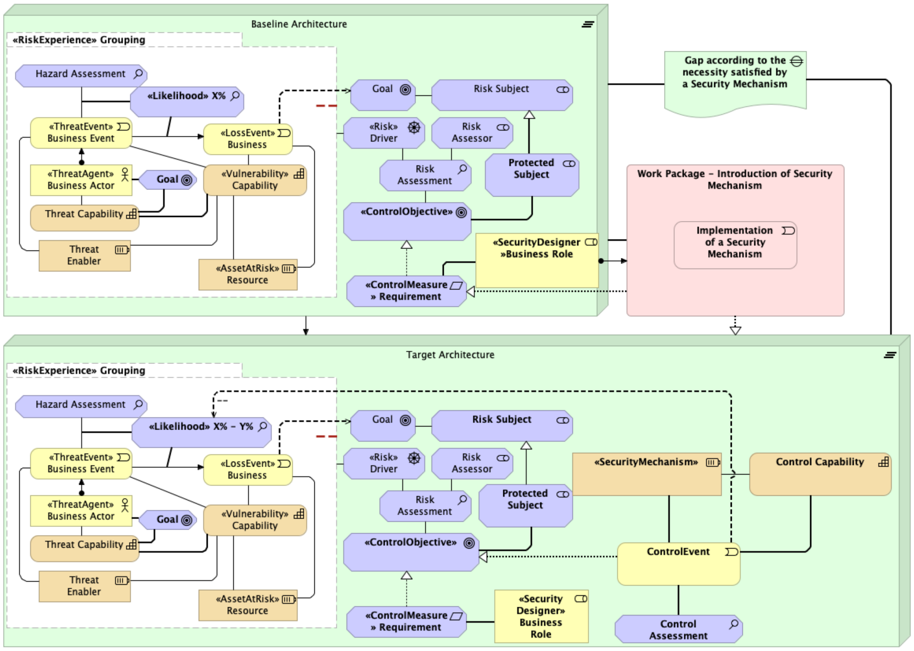

Fig. 7. Proposal for evolving the security aspects of the Risk and Security Overlay of ArchiMate

Figure 8 exemplifies our proposal using the same example from the RSO involving a loss event of production loss caused by a threat event of power fluctuation with intermediate steps in between. Notice that there is a certain likelihood associated with the causation between the power fluctuation and the power supply failure. The business owner is the risk subject, and the risk assessment is that the risk of production loss is unacceptable. Considering this risk experience in the baseline architecture, therefore

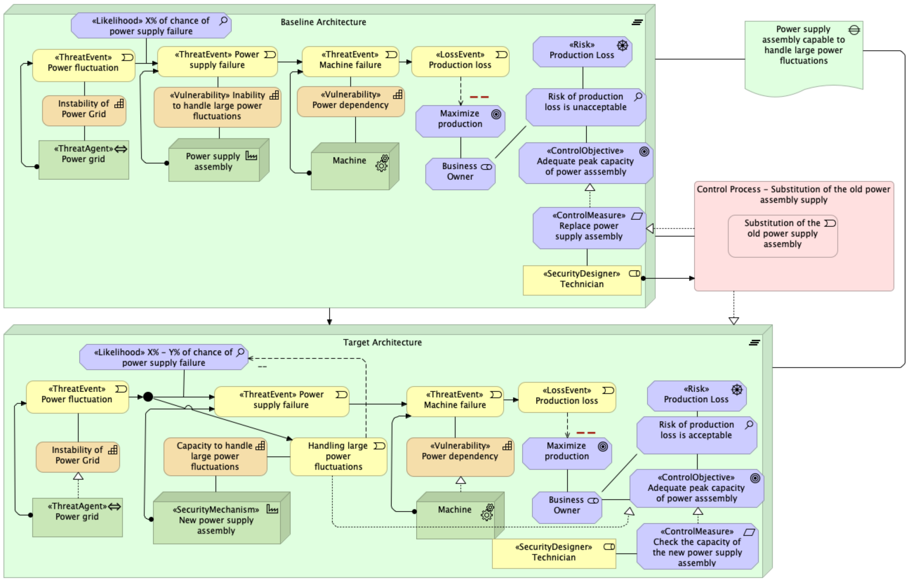
Baseline Architecture
«Likelihood» X% of chance of power supply failure
«ThreatEvent» Power fluctuation
«ThreatEvent» Power supply failure
«ThreatEvent» Machine failure
«LossEvent» Production loss
«Risk» Production Loss
Instability of Power Grid
«Vulnerability» Inability to handle large power fluctuations
«Vulnerability» Power dependency
Maximize production
Risk of production loss is unacceptable
«ThreatAgent» Power grid
Power supply assembly
Machine
Business Owner
«ControlObjective» Adequate peak capacity of power asssembly
«ControlMeasure» Replace power supply assembly
«SecurityDesigner» Technician
Power supply assembly capable to handle large power fluctuations
Control Process – Substitution of the old power assembly supply
Substitution of the old power supply assembly
Target Architecture
«Likelihood» X% – Y% of chance of power supply failure
«ThreatEvent» Power fluctuation
«ThreatEvent» Power supply failure
«ThreatEvent» Machine failure
«LossEvent» Production loss
«Risk» Production Loss
Risk of production loss is acceptable
Instability of Power Grid
Capacity to handle large power fluctuations
Handling large power fluctuations
«Vulnerability» Power dependency
Maximize production
«ControlObjective» Adequate peak capacity of power asssembly
«ThreatAgent» Power grid
«SecurityMechanism» New power supply assembly
Machine
Business Owner
«SecurityDesigner» Technician
«ControlMeasure» Check the capacity of the new power supply assembly

Fig. 8. Example of modeling the introduction of a security mechanism

before the introduction of a Security Mechanism, the Control Objective is defined to be an adequate peak capability of power assembly, realized by a Control Measure of replacing power supply assembly. This is the responsibility of a technician, the Security Designer. In the target architecture, we see some changes concerning the risk entities: the new power supply assembly is able to handle large power fluctuations, decreasing the likelihood of power supply failure; the original power supply assembly was totally removed from the scene, which means its vulnerability was also removed from scene. This is one of the ways of prevention [2]. Now, the risk of production loss is acceptable, because this interference in the risk causal chain ultimately decreased the chances of happening the production loss. Finally, Control Measure turned into checking the capability of the new power supply assembly. We provide the resulting files with related information in a public repository[6].

6 Related Work

The closest related works to ours are proposals of modeling Enterprise Risk Management and security through ArchiMate, as seen by ArchiMate's Risk and Security Overlay. The research conducted by Mayer and his collaborators [8] is one example of these proposals. They propose a conceptual model for Information System Security Risk Management, which is then integrated with enterprise architecture through ArchiMate's RSO. Their model contains four "risk treatment-related concepts": risk treatment, security requirement, control. These concepts are mapped into RSO metamodel without

[6] See: https://github.com/unibz-core/security-archimate

Table 3. Representation of security concepts in ArchiMate based on ROSE

Ontology Concept	Representation in ArchiMate
Protected Subject	A specialization of Risk Subject associated with a «ControlObjective»
Security Designer	Business Role associated with «ControlMeasure» and assigned to the implementation of a Security Mechanism
Security Mechanism	Structure Element (Business Agent, Resource) stereotyped with «SecurityMechanism»
Control Capability	Capability associated with Control Event and *SecurityMechanism*
Protection Trigger	Assessment stereopyed with «ControlAssessment»
Protection Event	Control Event that realizes «ControlObjective» and it is associated with «SecurityMechanism»

revision, which means that the problems we have shown remain untouched, such as construct redundancy and construct deficits.

Another related proposal is the Master thesis by Sander van den Bosch [3]. Based on Zachman Framework and SABSA Model, he proposes a metamodel describing risk and security elements, which are the following: vulnerability, threat, risk, security mechanism, and security policy. Then he employs them to extend ArchiMate towards "Secure Enterprise Architecture approach". The resulting language and the metamodel are validated by interviews with experts from both the enterprise architecture and the security discipline. The Master thesis by José Miguel Lino Teixeira [14] goes in a similar direction, but it maps ISO 22301 and ISO 31000 concepts into ArchiMate concepts, then introducing risk and security concepts. For example, the concept *Risk Source* from ISO 31000 is defined as an "Element which alone or in combination has the intrinsic potential to give rise to risk". The authors understands that "Risk can come from every layer of the ArchiMate, and we can assume that all elements can be a source of risk", including Business Actor, Driver, and Resource. Although both proposals present interesting results, their analysis of security are not grounded in any well-founded ontology like ROSE, which is founded in UFO. As a consequence, their analysis suffer from a degree of informality, and certain modeling patterns and security elements are missing, such as the ones presented previously.

7 Final Remarks

We presented an ontologically-founded analysis of the security modeling fragment of ArchiMate's Risk and Security Overlay (RSO). This analysis, grounded in the *Reference Ontology of Security Engineering* (ROSE) [9], allowed us to clarify the real-world semantics underlying the security-related constructs of the overlay, as well as to unveil several deficiencies in its modeling capabilities, including both redundancy and deficit of constructs. We then addressed these issues by redesigning the security modeling aspects of the RSO, making it more precise and expressive. The proposed redesign supports the representation of several important elements of Enterprise Risk Management and security that the original RSO neglects, including patterns of security mechanism, the subjects involved in it, the interdependence relations among risk entities, and the interaction between security and ArchiMate's baseline and target architecture. In doing so, we fill the gap left by a previous work that analyzed the risk and value aspects of ArchiMate [12, 13]. Therefore, we expect to contribute to the ontology-based modeling of enterprise risk and security in a more comprehensive manner. In future work, we intend to provide more examples in ArchiMate showing different patterns described by the theory of prevention [2] and to offer support for computational simulations of scenarios in Enterprise Risk Management and security.

References

1. Band, I., et al.: How to model enterprise risk management and security with the archimate language. The Open Group white paper (W172), 9 (2019)
2. Baratella, R., Fumagalli, M., Oliveira, Í., Guizzardi, G.: Understanding and modeling prevention. In: International Conference on Research Challenges in Information Science. pp. 389–405. Springer (2022)
3. van den Bosch, S.: Designing Secure Enterprise Architectures A comprehensive approach: framework, method, and modelling language. Master's thesis (May 2014)
4. Guizzardi, G.: Ontological foundations for structural conceptual models (2005)
5. Guizzardi, G., et al.: Grounding software domain ontologies in the Unified Foundational Ontology (UFO): The case of the ODE software process ontology. In: Ibero-American Conference on Software Engineering. pp. 127–140 (2008)
6. ISO: ISO 31000:2018 - Risk management – Guidelines (2018)
7. Lankhorst, M.: Enterprise Architecture at Work: Modelling, Communication and Analysis. Springer (2017)
8. 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 Workshop (EDOCW). pp. 106–116. IEEE (2017)
9. Oliveira, Í., et al.: An ontology of security from a risk treatment perspective. In: Chakravarthy, U., Mohania, M., Ralyté, J. (eds.) Conceptual Modeling. ER 2022. Springer (2022)
10. Rosemann, M., et al.: A reference methodology for conducting ontological analyses. In: International Conference on Conceptual Modeling. pp. 110–121. Springer (2004)
11. Sales, T.P., et al.: The common ontology of value and risk. In: Conceptual Modeling. ER 2018. pp. 121–135. Springer (2018)
12. Sales, T.P., et al.: Ontological analysis and redesign of risk modeling in ArchiMate. In: Intl. Enterprise Distributed Object Computing Conference. pp. 154–163 (2018)
13. Sales, T.P., et al.: A pattern language for value modeling in archimate. In: International Conference on Advanced Information Systems Engineering. pp. 230–245. Springer (2019)
14. Teixeira, J.M.L.: Modelling Risk Management using ArchiMate. Master's thesis (2017)
15. The Open Group: Archimate® 3.1 specification, https://pubs.opengroup.org/architecture/archimate3-doc/
16. The Open Group: Integrating risk and security within a togaf® enterprise architecture. The Open Group Guide white paper (2019), www.opengroup.org/library/g152