Modeling Trust in Enterprise Architecture: A Pattern Language for ArchiMate

Glenda Amaral, Tiago Prince Sales, Giancarlo Guizzardi, João Almeida, Daniele Porello

Modeling Trust in Enterprise Architecture: A Pattern Language for ArchiMate

Glenda Amaral[1], Tiago Prince Sales[1], Giancarlo Guizzardi[1], João Paulo A. Almeida[2], and Daniele Porello[3]

[1] Conceptual and Cognitive Modeling Research Group (CORE), Free University of Bozen-Bolzano, Bolzano, Italy
[gmouraamaral,tiago.princesales,giancarlo.guizzardi]@unibz.it
[2] Ontology & Conceptual Modeling Research Group (NEMO), Federal University of Espírito Santo, Vitória, Brazil
jpalmeida@ieee.org
[3] ISTC-CNR Laboratory for Applied Ontology, Trento, Italy
daniele.porello@loa.istc.cnr.it

Summary. Trust is widely acknowledged as the cornerstone of relationships in social life. But what makes an agent trust a person, a resource or an organization? Which characteristics should a trustee have in order to be considered trustworthy? The importance of understanding trust in organizations has motivated us to investigate the representation of trust concerns in enterprise models. Based on a well-founded reference ontology of trust, we propose a pattern language for trust modeling in ArchiMate. We present a first iteration of the design cycle, which includes the development of the pattern language and its demonstration by means of a realistic case study about trust in a COVID-19 data repository.

Key words: Trust Modeling, Enterprise Architecture, ArchiMate.

1 Introduction

Trust is a vital ingredient in productive relationships. According to Castelfranchi and Falcone [5], "trust in its intrinsic nature is a dynamic phenomenon" that changes with time. In times of crisis, such as the financial crisis of 2008 and the current COVID-19 health crisis, it becomes even more evident how fragile trust is. Therefore, the understanding of the building blocks that compose the trust of agents in a given trustee (such as an organization) is of paramount importance, as they reveal the qualities and properties the trustee should have in order to be considered trustworthy and effectively promote well-placed trust. Moreover, the identification of the trust components is fundamental to the assessment of risks that can emerge from trust relations.

From the perspective of an organization trustee, the modeling of trust in the context of Enterprise Architecture (EA) enables to bridge the gap between the stakeholders' trust concerns and the processes and other elements of the architecture that are needed to achieve the organization's goal of being trustworthy. The idea of modeling social and organizational concepts in the context of Enterprise Architecture has already been proposed in the literature in the context of value [20], risk [14, 18], service contracts [8],

resources and capabilities [3], however, the problem of linking the enterprise architecture to the stakeholders' trust concerns is still an open issue.

In this paper, we address this issue by proposing a trust modeling approach for ArchiMate, which is based on a proper ontological theory that provides adequate real-world and formal semantics for the concept of trust. In particular, we leverage the concepts and relations defined in the recently proposed Reference Ontology of Trust (ROT) [1], an ontologically well-grounded reference model that formally characterizes the concept of trust and explains how risk emerges from trust. ROT is specified in OntoUML [10], and thus, compliant with the meta-ontological commitments of the Unified Foundational Ontology (UFO) [10]. Based on ROT, we propose a Trust Pattern Language (TPL) for ArchiMate–the most used modeling language in the EA field. A pattern language [4] consists of a set of interrelated modeling patterns and its main advantage is that it offers a context in which related patterns can be combined, thus, reducing the space of design choices and design constraints [7]. We designed TPL following the Design Science Research methodology [12]. In this paper, we present the first iteration of the *design cycle* (building and evaluating), which includes the development of the pattern language and its demonstration by means of a real case study of trust in a COVID-19 data repository.

The remainder of this paper is organized as follows. Section 2 introduces the reader to the Reference Ontology of Trust (ROT) that provides the ontological foundations in which the Trust Pattern Language (TPL) is grounded. Section 3 presents the set of requirements identified for the language (Section 3.1), which are needed for a formal evaluation of the language. Afterward, the individual modeling patterns that compose TPL are presented (Section 3.2), as well as a method for combining them (Section 3.3). In Section 4, we demonstrate how TPL can be used by presenting a real case example of trust in a COVID-19 data repository. We conclude in Section 5 with some final considerations.

2 Research Baseline

2.1 The Reference Ontology of Trust

The Reference Ontology of Trust[1] (ROT) is a UFO-based ontology that formally characterizes the concept of trust, clarifies the relation between trust and risk, and represents how risk emerges from trust relations [1]. ROT makes the following ontological commitments about the nature of trust:

- **Trust is relative to a goal.** An agent, the trustor, trusts someone or something, the trustee, only relative to a goal, for the achievement of which she counts upon the trustee.
- **Trust is a complex mental state of a trustor regarding a trustee and her behavior.** It is composed of: (i) a trustor's intention, whose propositional content is a goal of the trustor; (ii) the belief that the trustee has the capability to perform the desired action

[1] The complete version of ROT in OntoUML and its implementation in OWL are available at `http://purl.org/krdb-core/trust-ontology`.

or exhibit the desired behavior; and (iii) the belief that the trustee's vulnerabilities will not prevent her from performing the desired action or exhibiting the desired behavior. When the role of trustee is played by an agent, trust is also composed of the trustor's belief that the trustee has the intention to exhibit the desired behavior.

– **The trustor is necessarily an "intentional entity".** Briefly put, the trustor is a cognitive agent, an agent endowed with goals and beliefs [5].
– **The trustee is not necessarily a cognitive system.** The trustee is an entity capable of having a (hopefully positive) impact on a goal of the trustor by the outcome of its behavior [5]. A trustee may be a person, an animal, a car, a vaccine, etc.
– **Trust is context dependent.** The trustor may trust the trustee for a given goal in a given context, but not do so for the same goal in a different context. We assume trust relations to be highly dynamic [5].
– **Trust implies risk.** By trusting, the trustor accepts to become vulnerable to the trustee in terms of potential failure of the expected behavior and result, as the trustee may not exhibit the expected behavior or it may not have the desired result [13, p 21].

Fig. 1 depicts a ROT excerpt, which is represented in OntoUML, an ontology-driven conceptual modeling language based in UFO[11].

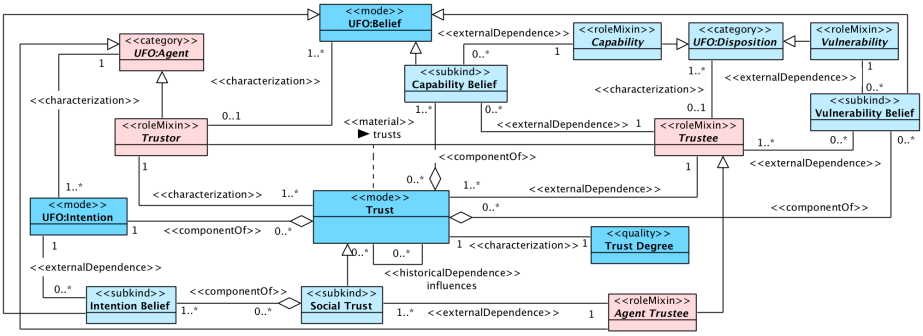


Fig. 1: A fragment of ROT depicting the mental aspects of trust

In ROT, `Trust` is modelled as a complex mode (an externally dependent entity, which can only exist by inhering in other individuals [10]) composed of an `Intention` whose propositional content is a goal of the `Trustor`, and a set of `Beliefs` that inhere in the `Trustor` and are externally dependent on the `Dispositions` [9, 3] that inhere in the `Trustee`. These beliefs include: (i) the `Belief` that the `Trustee` has the `Capability` to exhibit the desired behavior (`Capability Belief`); and (ii) the `Belief` that the `Trustee's Vulnerabilities` will not prevent her from exhibiting the desired behavior (`Vulnerability Belief`). The `Trustee's Vulnerabilities` and `Capabilities` are dispositions that inhere in the `Trustee`, which are manifested in particular situations, through the occurrence of events [9]. `Social Trust` is a specialization of `Trust` in which the `Trustee` is an `Agent`. Therefore, this form of trust is also composed of the `Trustor`'s belief that the `Agent Trustee` has the `Intention` to perform the desired action (`Intention Belief`). The relation `influences` represents that an instance of `Trust` can influence another (positively or negatively) [15].

ROT relies on the Common Ontology of Value and Risk (COVER) (Fig. 2) proposed by Sales et al. [19] to represent the relation between trust and risk. COVER proposes an ontological analysis of notions such as Risk, Risk Event and Vulnerability, among others. A central notion for characterizing risk in COVER is a chain of events that impacts an agent's goals, which the authors name Risk Experience. Risk Experiences focus on unwanted events that have the potential of causing losses and are composed of events of two types, namely threat and loss events. A `Threat Event` is the one with the potential of causing a loss, which might be intentional or unintentional. A `Threat Event` might be the manifestation of: (i) a `Vulnerability` (a special type of disposition whose manifestation constitutes a loss or can potentially cause a loss from the perspective of a stakeholder); or (ii) a `Threat Capability` (capabilities whose manifestation enables undesired events that threaten agent's abilities to achieve a goal). The second mandatory component of a Risk Experience is a `Loss Event`, which necessarily impacts intentions in a negative way [19].

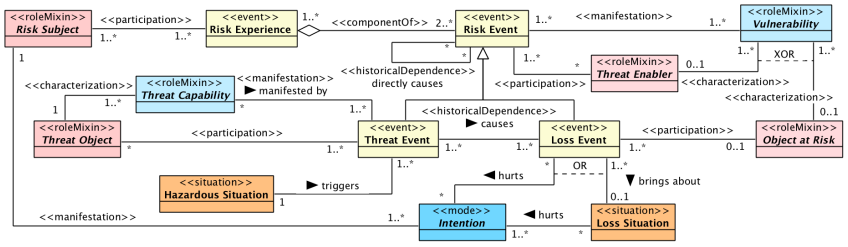


Fig. 2: A fragment of COVER depicting risk experience [19]

2.2 ArchiMate

ArchiMate is a modeling standard that defines a layered structure by means of which the architecture of enterprises can be described [21]. The language is organized in six layers, namely *Strategy*, *Business*, *Application*, *Technology*, *Physical*, and *Implementation & Migration* [21]. In this paper, we focus on the elements of the Strategy and Business layers.

A model in ArchiMate is a collection of elements and relationships. In ArchiMate, each element is classified according to its nature, referred to as "aspects": an *Active Structure Element* represents an entity that is capable of performing behavior, a *Passive Structure Element* represents a structural element that cannot perform behavior, a *Behavior Element* represents a unit of activity performed by one or more active structure elements, a *Motivation Element* is one that provides the context of or reason behind the architecture, and a *Composite Element* is simply one that aggregates other elements.

The most relevant ArchiMate elements for the TPL are: (i) Stakeholder, Driver, Assessment and Goal (Motivation Elements); (ii) Resource (a Passive Structure Element); (iii) Business Actor (an Active Structure Element); (iv) Capability and Business Event (Behavior Elements); and (v) Grouping. As for relations, the most relevant ones are: (i) Composition and Realization (when applied to Structural elements); (ii) Influence (which is a sort of Dependency); (iii) Triggering (when applied to Behavior); and (iv) Association (which can be used flexibly in many contexts to relate elements when other

more specific relations are not available). A detailed definition of the concepts of the language can be found in the ArchiMate specification [21].

3 A Pattern Language for Trust Modeling

3.1 Language Requirements

According to Buschmann et al. [4], "a pattern describes a particular recurring design problem that arises in specific design contexts and presents a well-proven solution for the problem". Deutsch [6] defines a pattern language as "a set of patterns and relationships among them that can be used to systematically solve coarse-grained problems". We have established two types of requirements in the design of the TPL: (i) *analysis requirements*, which refer to what the models produced with the language should help users to achieve, either by means of automated or manual analysis; and (ii) *ontological requirements*, which refer to the concepts and relations the language should have in order to accurately represent its domain of interest and thus support its intended uses.

Below we present the list of the analysis requirements for the TPL:

R1. *Trustworthiness analysis*: an enterprise should be able to gain insight into why it trusts certain key resources, actors or partners (or event if they should do it in the first place!). In particular, for a given trust relation, the enterprise should be able to identify the capabilities and vulnerabilities of a particular trustee that are the focus of its beliefs, so that it can detect potential threats to the achievement of its goals. From the opposite perspective, the enterprise should be able to identify what makes them trustworthy (or not) from the point-of-view of their customers and partners, possibly identifying what it could change to increase trust levels, as well the key capabilities it needs to guarantee to promote well-placed trust.
R2. *Risk analysis*: By modeling the elements that compose the trust complex mental state of a trustor regarding a trustee, an enterprise should be able to identify risks that can emerge as consequence of either the manifestation of a trustee's vulnerability or the unsatisfactory manifestation of a trustee's capability.

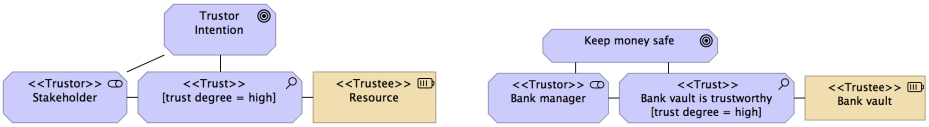
As for the ontological requirements, they consist of an isomorphic representation of the concepts and relations defined in the Reference Ontology of Trust, in which it is based. In addition to the aforementioned requirements, we assume the following constraints for the TPL:

R3. It should rely exclusively on constructs available in ArchiMate 3.0.1 [21], in an effort to retain its user base and tool support, as well as to prevent adding complexity to the language.
R4. It should map trust-related concepts into ArchiMate constructs maintaining, as much as possible, their original meaning as described in the standard. Specialized semantics should be addressed via stereotypes, constituting thus a lightweight extension of the language.

3.2 Trust Modeling Patterns

Trust Assessment. This pattern allows modelers to represent a trust relation between a trustor and a trustee, in which the former trusts the latter with respect to an intention (whose propositional content is a goal, for the achievement of which the trustor counts upon the trustee). The trustor is always a cognitive agent, endowed with goals and beliefs. As for the trustee, it is an entity able to cause an impact (hopefully positive) on a trustor's goal by the outcome of its behavior. Note that the role of trustee can be played not just by agents, but also by objects, such as rules, procedures, conventions, infrastructures, tools, artifacts in general, as well as different types of social systems. For this reason, this pattern has two variants, depending on the type of the trustee.

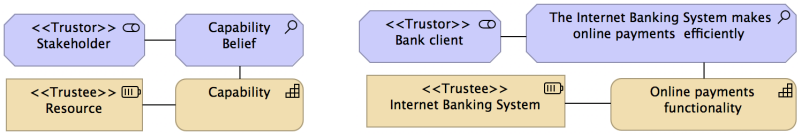
The first variant, depicted in Fig. 3a, details the trust relation when the trustee is an object. It consists of a `Structure Element`, the trustee, connected to a «`Trust`» `Assessment`, which in turn is connected both to a `Stakeholder`, the trustor, and to the `Goal` she is counting on achieving. Attached to the «`Trust`» `Assessment` is the `Trust Degree`, which is an attribute that can be described as an entry in a scale chosen by the modeler, such as a discrete scale like `<Low,Medium,High>` or a continuous scale like `<0-100>`. An example of this first variant is shown in Fig. 3b. In the second variant, the trustee is a cognitive agent and thus is modeled as a «`Trustee`» `Stakeholder`.



(a) (b)

Fig. 3: The Trust Assessment Pattern

Capability Belief. This pattern allows modelers to express which capability of the trustee is the focus of a capability belief of the trustor. Capabilities are dispositions that inhere in agents and objects, which are manifested in particular situations, through the occurrence of events. They are usually understood as positive dispositions, in the sense that they enable the manifestation of events desired by an agent. The generic structure of the Capability Belief Pattern is depicted in Fig. 4a. It connects a `Capability Belief Assessment` of a «`Trustor`» `Stakeholder` to the corresponding `Capability` of a «`Trustee`». An application of this pattern is presented in Fig. 4b.

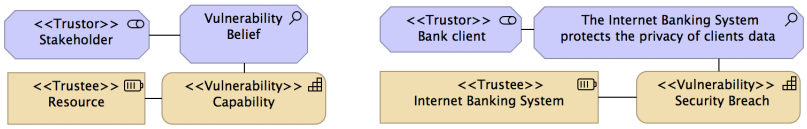


(a) (b)

Fig. 4: The Capability Belief Pattern

Vulnerability Belief. This pattern allows modelers to express which vulnerability of the trustee is the focus of a vulnerability belief of the trustor. Vulnerabilities are a special

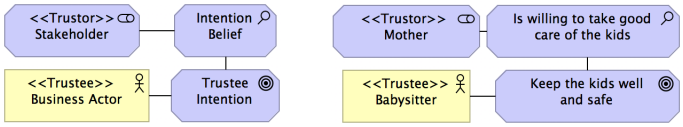
type of disposition whose manifestation constitutes a loss or can potentially cause a loss from the perspective of a stakeholder. The generic structure of the Vulnerability Belief Pattern is depicted in Fig. 5a. It connects a `Vulnerability BeliefAssessment` of a «`Trustor`» `Stakeholder` to the corresponding `Vulnerability` of a «`Trustee`». Fig. 5b presents an application example of this pattern.



(a) (b)

Fig. 5: The Vulnerability Belief Pattern

Intention Belief. This pattern allows modelers to express which intention of the trustee is the focus of an intention belief of the trustor. Its generic structure is depicted in Fig. 6a. It connects an `Intention Belief Assessment` of a «`Trustor`» `Stakeholder` to the corresponding `Goal` of a «`Trustee`». Fig. 6b presents an application example for this pattern.



(a) (b)

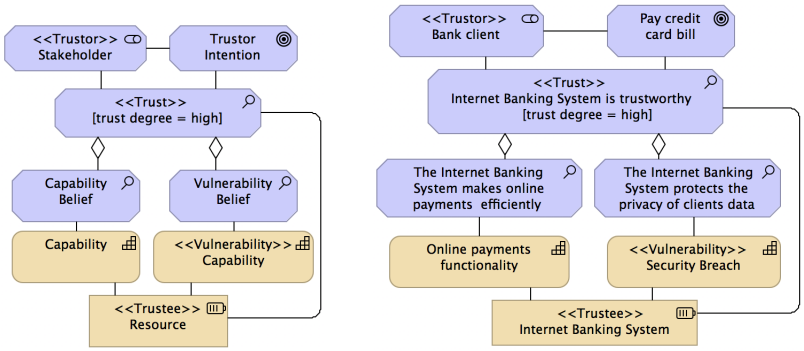
Fig. 6: The Intention Belief Pattern

Trust Composition. To account for what makes an agent trust a resource or another agent, we introduce the Trust Composition Pattern, which details the complex mental state of the trustor. The understanding of the elements that compose trust is important because they reveal the qualities and properties the trustee should have in order to be considered trustworthy and effectively promote well-placed trust. This pattern refines the Trust Assessment Pattern by detailing the decomposition of the «`Trust`» `Assessment` into the beliefs of the trustor about the trustee. It has two variants, as the beliefs of the trustor vary according to the trustee type.

The first variant, depicted in Fig. 7a, details trust when the trustee is not a cognitive agent. In this case, we make use of the Capability Pattern and the Vulnerability Pattern to represent that the «`Trust`» `Assessment` is composed of `Belief Assessments` of the trustor regarding the `Capabilities` and `Vulnerabilities` of the trustee (the trustor believes that the trustee has the capability to exhibit a desired behavior and that its vulnerabilities will not prevent it from exhibiting this behavior). Fig. 7b shows an application example of this pattern.

In the second variant the trustee is a cognitive agent endowed with goals and, therefore, her intentions are also part of the set of beliefs that compose trust. Besides believing that the trustee is capable of exhibiting a desired behavior and that her vulnerabilities

will not stop her from doing that, the trustor believes that trustee has the intention to exhibit the aforementioned behavior. Therefore, in this case, in addition to the Capability Belief and Vulnerability Belief Patterns, the Intention Belief Pattern is also used to represent the «`Trust`» `Assessment`.



(a) (b)

Fig. 7: The Trust Composition Pattern

Risk Experience. In order to account for how risk emerges from trust relations, we propose the Risk Experience Pattern, presented in Fig. 8. Once the components of trust are known (decomposed using the Trust Composition Pattern), it is possible to identify the risks related to the capabilities and vulnerabilities of the trustee, which are the focus of trustor's beliefs.

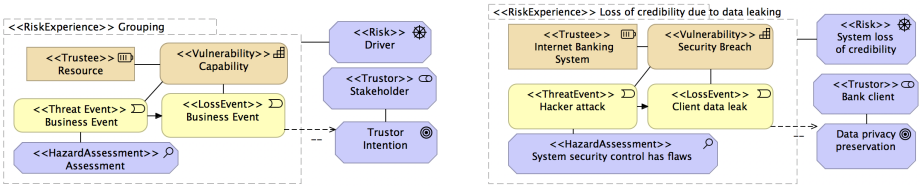
Our modeling strategy is directly inspired by the risk modeling approach proposed by Sales et al. [18]. Given the objectives of our pattern, we focus here on the perspective of risk as a chain of events that impact an agent's goals, which the authors named `Risk Experience`. Risk Experiences focus on unwanted events that have the potential of causing losses and are composed by events of two types, namely threat and loss events [18]. A `Threat Event` is the one with the potential of causing a loss. As described in [18], it might be the manifestation of: (i) a `Vulnerability`; or (ii) `Threat Capability` (as aforementioned, capabilities are usually perceived as beneficial, as they enable the manifestation of events desired by an agent. However, when the manifestation of a capability enables undesired events that threaten agent's abilities to achieve a goal, it can be seen as a `Threat Capability`). The second mandatory component of a `Risk Experience` is a `Loss Event`, which necessarily impact intentions in a negative way.

Following the strategy of Sales et al. [18], we mapped `Risk Experience` as a `Grouping` decorated with the «`RiskExperience`» stereotype. Such a grouping should aggregate the elements and the relations in an experience. Then, we associated the «`RiskExperience`» `Grouping` with risks, which are mapped as «`Risk`» `Drivers`, as drivers represent "conditions that motivate an organization to define its goals and implement the changes necessary to achieve them" [21].

The first variant, depicted in Fig. 8a, allows modelers to represent the existence of risks related to `Vulnerabilities` of the trustee that are the focus of beliefs of the

trustor. «ThreatEvent» Event might be the manifestation of a Vulnerability and may lead to a «LossEvent» Event, which impacts the Trustor Intention in a negative way, as it hurts her Intention of reaching a specific goal. «HazardAssessment» Assessment stands for situations that activate vulnerabilities and threat capabilities, which in turn will be manifested as «ThreatEvent» Events. Since ArchiMate does not provide a native construct for modeling situations in general, we followed the approach used in [18] and represent hazardous situations as assessments about them. Fig. 8b shows an application example of this pattern.

The second variant is similar to the previous one, as it also represents the existence of risks related to a disposition of the trustee, though in this case the disposition is a Threat Capability. As previously mentioned, when the manifestation of a capability enables undesired events that threatens agent's abilities to achieve a goal, it can be seen as a Threat Capability. Analogous to the former variant, a «ThreatEvent» Event might be the manifestation of a Threat Capability of the trustee if the trustee fails to perform this specific Capability that was supposed to bring about an outcome desired by the trustor. Finally, the «ThreatEvent» Event can trigger a «LossEvent» Event, which has a negative impact on the Trustor Intention.



(a) (b)

Fig. 8: The Risk Experience Pattern

Risk Assessment. This pattern, also extracted from [18], complements our approach on the modeling of risks that emerge from trust relations. It consists of a Risk Assessment made by a Stakeholder about a «Risk» Driver, which in turn is associated to a «RiskExperience» Grouping. In addition, the Risk Assessment is connected to a «ControlObjective» Goal, a sort of high level goal that defines what the organization intends to do about an identified risk. Control Goals are connected to «ControlMeasure» Requirements that represent desired properties of solutions – or means – to realize such goals. Using this pattern, depicted in Fig. 9, it is possible to model the realization of control measures by any set of core elements, such as business processes (e.g. a data quality management process), application services (e.g. a scanning service) or nodes (e.g. a document management system).

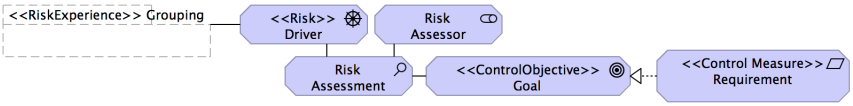


Fig. 9: The Risk Assessment Pattern

Trust Influencing Trust. This pattern allows modelers to represent that trust can influence trust, either positively or negatively. For example, one's trust in the local police officer may increase one's trust in the "judiciary system". It can be further used to characterize the existence of "trust by delegation". The idea behind "trust by delegation" is that when, for example, Alice trusts Bob, and Bob trusts Charlie, then Alice can derive a measure of "trust by delegation" in Charlie. In this case the «`Trust`» `Assessments` "Alice trusts Bob" and "Bob trusts Charlie" positively influence the «`Trust`» `Assessment` "Alice trusts Charlie". As shown in Fig. 10, the pattern makes explicit the influence association between a «`Trust`» `Assessment` and the other one under its influence.

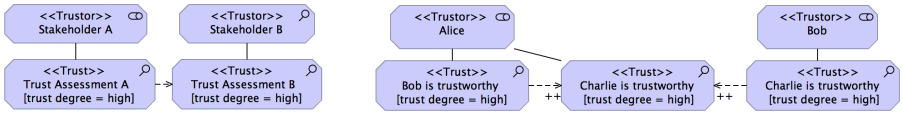


Fig. 10: The Trust Influencing Trust Pattern

The mapping between the ontological trust-related concepts and their representation in ArchiMate is listed in Table 1.

Table 1: Representation of trust and risk-related concepts in ArchiMate.

Concept	Representation in ArchiMate
`Trust`	«`Trust`» `Assessment`
`Trustor`	«`Trustor`» `Stakeholder`
`Trustee`	«`Trustee`» `Stakeholder` or «`Trustee`» `Structure Element`
`Trust Degree`	Attribute of a «`Trust`» `Assessment`
`Capability`	`Capability`
`Vulnerability` [18]	«`Vulnerability`» `Capability`
`Intention`	`Goal`
`Belief`	`Assessment`
`Capability Belief`	`Assessment` connected to a `Capability`
`Vulnerability Belief`	`Assessment` connected to a «`Vulnerability`» `Capability`
`Intention Belief`	`Assessment` connected to a `Goal`
`Risk` [18]	«`Risk`» `Driver`
`Risk Assessment` [18]	`Assessment` connected to a «Risk» Driver
`Risk Assessor` [18]	`Stakeholder` connected to a `Risk Assessment`
`Risk Experience` [18]	«`RiskExperience`» `Grouping`
`Threat Event` [18]	«`ThreatEvent`» `Event`
`Loss Event` [18]	«`LossEvent`» `Event`
`Hazard Assessment` [18]	«`HazardAssessment`» `Assessment`

3.3 Combining the Patterns

To use TPL, a modeler may start with the application of the Trust Assessment Pattern to identify both the trustor and the trustee, as well as the goal of the trustor, for the achievement of which she is counting on the trustee. Then, the user should use the Trust Composition Pattern by iteratively applying the Capability Belief Pattern, the Vulnerability Pattern, and the Intention Belief Pattern (this latter only if the trustee is an agent) in order to detail the components of trust: the capabilities, vulnerabilities, and intentions of the trustee, which are the focus of the trustor's beliefs. For each vulnerability and capability, the modeler should apply the Risk Experience Pattern to identify the risks that can emerge when either the vulnerabilities are manifested or the capabilities are not manifested as expected (and in this case they play the role of threat capabilities). Finally, for each risk driver identified, the user may apply the Risk Assessment Pattern to evaluate the impact of risks and establish procedures for effective risk control, treatment, and mitigation. As previously mentioned, from this pattern it is possible to model the realization of control measures by describing how the many pieces of an enterprise's application and technology infrastructure work together to properly manage risks that emerge from trust relations. Additionally, the Trust Influencing Trust Pattern can be applied to make explicit how trust relations influence each other (for instance, Alice trusting an online store can influence her brother trusting the online store too), as well as to characterize the existence of trust by delegation (for example, Alice trusts Bob, and Bob trusts an information source, then it may be the case that Alice trusts the information source "by delegation"). The detailed diagrams presenting the complete process of combining the patterns can be found `https://purl.org/krdb-core/trust-archimate`.

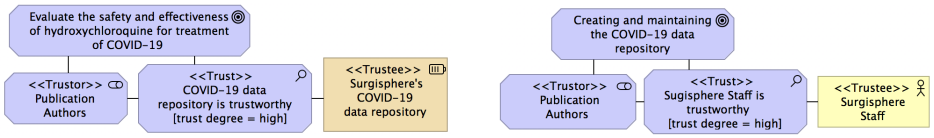
4 Case Study

In this section, we present a realistic study in which we use the TPL to model a case of "misplaced trust" in a COVID-19 data repository, which resulted in the retraction of a publication from a highly influential and prestigious medical journal. In particular, we refer to the case of a recent study published in The Lancet journal [16], which relied on data gathered by a US healthcare analytics company called Surgisphere to report issues on the efficacy and safety of hydroxychloroquine (HCQ) for treating COVID-19. When the study was first published it prompted the World Health Organisation (WHO) along with several countries to pause trials on this drug. However, this very study was retracted [17] a few days later (and the clinical trials resumed), as concerns were raised with respect to the veracity of the data, leading the authors to recognize that they could no longer vouch for the veracity of the database at the heart of the study. Examples of problems encountered include errors in the Australian data and the fact that independent reviewers could not verify the validity of the data, as Surgisphere would not give access to the full dataset, citing confidentiality and client agreements [17].

Given the limited space available, we only present relevant fragments of the resulting model. The complete case study is available at `https://purl.org/krdb-core/`

`trust-archimate`. An investigation of the characteristics a COVID-19 data repository should have in order to be held in a position of trust by the communities they intend to serve are presented in an accompanying technical report [2], available at the above-mentioned URL.

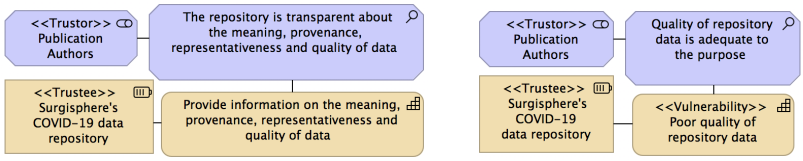
We start with the application of the Trust Assessment Pattern to identify the trustees, the trustors, and their goals. In our case study, different trust relations can be observed: (i) the Publication Authors trust the COVID-19 data repository to *evaluate the safety and effectiveness of hydroxychloroquine for treatment of COVID-19*; (ii) the Publication Authors trust the Surgisphere Staff about *creating and maintaining the COVID-19 data repository*; (iii) The Lancet trusts the Publication Authors to *accept publishing the study*; (iv) WHO trusts The Lancet to *have reliable information to make decisions w.r.t. recommendations on the treatment of diseases*; (v) WHO trusts (by delegation) the Publication Authors to *have reliable information to make decisions w.r.t. recommendations on the treatment of diseases*; and (vi) Countries trust WHO to *have reliable recommendations on the treatment of diseases*. Fig. 11a and 11b depict the modeling of trust relations (i) and (ii), respectively.



(a) (b)

Fig. 11: Application of the Trust Assessment Pattern

We proceed by iteratively applying the Capability Belief Pattern (Fig. 12a) and the Vulnerability Belief Pattern (Fig. 12b) to detail the Publication Authors' beliefs with respect to the COVID-19 data repository (trust assessment depicted in Fig. 11a). Finally, in Fig. 13 we use the Trust Composition Pattern to detail the trust complex mental state of the Publication Authors in their trust relation with the COVID-19 data repository. Note that the capabilities and vulnerabilities which are the focus of the Publication Authors' beliefs were identified based on the trust concerns for COVID-19 data presented in [2], such as *transparency*, *privacy of data*, *respect for human rights* and *data quality*.



(a) (b)

Fig. 12: Capability and Vulnerability Beliefs

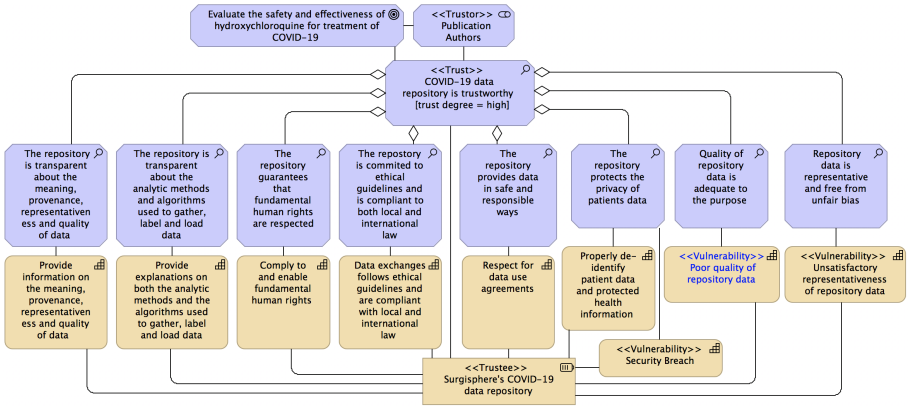


Fig. 13: Composition

Since the components of trust are known, it is possible to reason about possible manifestations of vulnerabilities and (threatening) capabilities of the COVID-19 data repository, which can enable undesired events that threaten the Publication Authors' abilities to achieve their goal.

Using the Risk Experience Pattern, we represent, in Fig. 14, the emergence of the risk of "repository loss of credibility" caused by the poor quality of data (a vulnerability), which revealed errors in the data, thus preventing the authors from attesting the validity of the study. Then we apply the Risk Assessment Pattern (Fig. 15) to represent the evaluation of the risk of "repository loss of credibility" by the Surgisphere Staff, as well as the establishment of procedures for effective risk control (improve data quality) and the definition of a control measure that describes how Surgisphere plans to realize these procedures (implement data quality management).

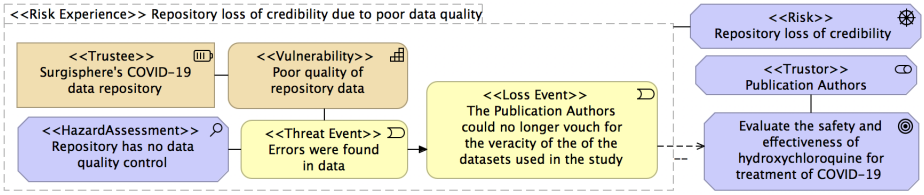


Fig. 14: Risk Experience

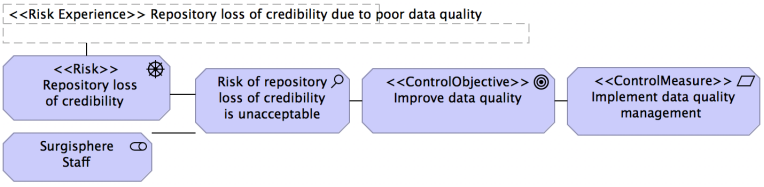


Fig. 15: Risk Assessment

Lastly, we use the Trust Influencing Trust Pattern to make explicit how some of these trust assessments influence each other. In Fig. 16a we may observe that "WHO

trusting The Lancet", positively influences "WHO's trust in the Publication Authors". Similarly, the Publication Authors' trust in the Surgisphere's Staff expertise positively influences their trust in the COVID-19 data repository (Fig. 16b). Note that as previously mentioned, this pattern can also be applied to characterize the existence of "trust by delegation". For example, considering that (1) "WHO trusts The Lancet" and (2) "The Lancet trusts the Publication Authors", there is a great chance that, (3) "WHO trusts the Publication Authors" by delegation, and in this case both (1) and (2) positively influences (3).

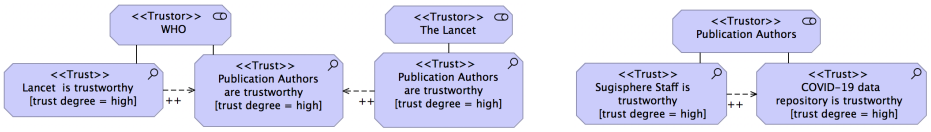


Fig. 16: Trust Influencing Trust

5 Conclusions

In this paper we presented TPL, a pattern language for modeling trust in ArchiMate that is based on ROT, a recently proposed ontology that provides clear real-world semantics for the constituting elements of trust and describes the emergence of risk from trust relations. Although trust towards agents and resources is a known concern in the literature, little has been said about what constitutes the stakeholders' trust in a given organization or resource, as well as how these trust concerns permeate the enterprise architecture. The TPL was designed aiming at addressing these issues. In particular, it allows to represent: (i) the elements that constitute the trust of an agent with respect to a resource or another agent, including organizations; (ii) the capabilities and vulnerabilities of trustees that are the focus of the trustor's beliefs, in a trust assessment; (iii) the influence that trust assessments have on each other; (iv) the risks that can emerge from trust relations; and (v) risk assessments related to these risk drivers.

This work is part of a long-term research program that aims at using UFO as a semantic foundation for enterprise modeling (in particular, for ArchiMate). Next, we envision that this effort can be harmonized with previous work (on value [20], risk [18], service contracts [8], resources and capabilities [3]) to provide a comprehensive ontology-based enterprise modeling approach. We also plan to conduct empirical experiments to validate the TPL. In addition, we want to further evolve the trust ontology to allow the representation of "pieces of evidence" for trustworthiness, which comes from elements such as a history of performance and trusted third party certifications.

Acknowledgments

CAPES (PhD grant 88881.173022/2018-01) and NeXON project (UNIBZ). João Paulo A. Almeida is funded by the Brazilian National Council for Scientific and Technological Development CNPq (grant 312123/2017-5).

References

1. Amaral, G., Sales, T.P., Guizzardi, G., Porello, D.: Towards a Reference Ontology of Trust. In: International Conference on Cooperative Information Systems. pp. 3–21. Springer (2019)
2. Amaral, G., Sales, T.P., Guizzardi, G., Porello, D.: Trust Concerns for Digital Data Repositories: the COVID-19 Data Domain. Tech. rep., Free University of Bozen-Bolzano (2020)
3. Azevedo, C.L., Iacob, M.E., Almeida, J.P.A., van Sinderen, M., Pires, L.F., Guizzardi, G.: Modeling resources and capabilities in enterprise architecture: A well-founded ontology-based proposal for ArchiMate. Information systems **54**, 235–262 (2015)
4. Buschmann, F., Henney, K., Schmidt, D.C.: Pattern-oriented software architecture, on patterns and pattern languages, vol. 5. John Wiley & Sons (2007)
5. Castelfranchi, C., Falcone, R.: Trust theory: A socio-cognitive and computational model, vol. 18. John Wiley & Sons (2010)
6. Deutsch, P.: Models and patterns. In: Software factories: Assembling applications with patterns, frameworks, models and tools. John Wiley & Sons (2004)
7. Falbo, R., Barcellos, M., Ruy, F., Guizzardi, G., Guizzardi, R.: Ontology pattern languages. In: Ontology Engineering with Ontology Design Patterns: Foundations and Applications. IOS Press (2016)
8. Griffo, C., Almeida, J.P.A., Guizzardi, G., Nardi, J.C.: From an ontology of service contracts to contract modeling in enterprise architecture. In: Proc. 21st IEEE EDOC. pp. 40–49 (2017)
9. Guizzardi, G., Wagner, G., Falbo, R., Guizzardi, R., Almeida, J.: Towards Ontological Foundations for the Conceptual Modeling of Events. In: Proc. 32th Int. Conference on Conceptual Modeling (ER). Lecture Notes in Computer Science, vol. 8217, pp. 327–341. Springer (2013)
10. Guizzardi, G.: Ontological foundations for structural conceptual models. Telematica Instituut Fundamental Research Series, No. 15, ISBN 90-75176-81-3 (2005)
11. Guizzardi, G., Wagner, G., Almeida, J.P.A., Guizzardi, R.S.S.: Towards ontological foundations for conceptual modeling: the Unified Foundational Ontology (UFO) story. Applied ontology **10**(3-4), 259–271 (2015)
12. Hevner, A., Chatterjee, S.: Design science research in information systems. In: Design research in information systems, pp. 9–22. Springer (2010)
13. Luhmann, N.: Trust and power. John Wiley & Sons (2018)
14. 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. pp. 106–116. IEEE (2017)
15. McKnight, D.H., Chervany, N.L.: Trust and distrust definitions: One bite at a time. In: Trust in Cyber-societies, pp. 27–54. Springer (2001)
16. Mehra, M.R., Desai, S.S., Ruschitzka, F., Patel, A.N.: RETRACTED: Hydroxychloroquine or chloroquine with or without a macrolide for treatment of COVID-19: a multinational registry analysis. The Lancet (May 2020)
17. Mehra, M.R., Ruschitzka, F., Patel, A.N.: Retraction—hydroxychloroquine or chloroquine with or without a macrolide for treatment of COVID-19: a multinational registry analysis. The Lancet **395**(10240), 1820 (2020)
18. Sales, T.P., Almeida, J.P.A., Santini, S., Baião, F., Guizzardi, G.: Ontological analysis and redesign of risk modeling in ArchiMate. In: 2018 IEEE 22nd International Enterprise Distributed Object Computing Conference. pp. 154–163. IEEE (2018)
19. Sales, T.P., Baião, F., Guizzardi, G., Guarino, N., Mylopoulos, J.: The common ontology of value and risk. In: Proc.37th ER Conference. vol. 11157, pp. 121–135. Springer (2018)
20. Sales, T.P., Roelens, B., Poels, G., Guizzardi, G., Guarino, N., Mylopoulos, J.: A pattern language for value modeling in ArchiMate. In: International Conference on Advanced Information Systems Engineering. pp. 230–245. Springer (2019)
21. The Open Group: ArchiMate 3.0.1 Specification. Standard C179 (2017)