

A Pattern Language for Value Modeling in ArchiMate

Tiago Prince Sales^{1,2}, Ben Roelens³, Geert Poels⁴, Giancarlo Guizzardi⁵,
Nicola Guarino², and John Mylopoulos¹

¹ University of Trento, Trento, Italy

² ISTC-CNR Laboratory for Applied Ontology, Trento, Italy

³ Open University, Heerlen, The Netherlands

⁴ Ghent University, Ghent, Belgium

⁵ Free University of Bozen-Bolzano, Bolzano, Italy

{tiago.princesales,john.mylopoulos}@unitn.it, ben.roelens@ou.nl,
geert.poels@ugent.be, giancarlo.guizzardi@unibz.it, nicola.guarino@cnr.it

Abstract. In recent years, there has been a growing interest in modeling value in the context of Enterprise Architecture, which has been driven by a need to align the vision and strategic goals of an enterprise with its business architecture. Nevertheless, the current literature shows that the concept of value is conceptually complex and still causes a lot of confusion. For example, we can find in the literature the concept of value being taken as equivalent to notions as disparate as goals, events, objects and capabilities. As a result, there is still a lack of proper support for modeling all aspects of value as well as its relations to these aforementioned notions. To address this issue, we propose in this paper a pattern language for value modeling in ArchiMate, which is based on the Common Ontology of Value and Risk, a well-founded reference ontology developed following the principles of the Unified Foundation Ontology. This enables us to delineate a clear ontological foundation, which addresses the ambiguous use of the value concept. The design of the Value Pattern Language will be guided by the Design Science Research Methodology. More specifically, a first iteration of the build-and-evaluate loop is presented, which includes the development of the pattern language and its demonstration by means of a case study of a low-cost airline.

Keywords: Value Modeling · Enterprise Architecture · ArchiMate.

1 Introduction

In the last decades, several value modeling languages have been introduced, such as *e³value* [9] and VDML [19]. However, it is only recently that there is an interest in modeling value in the context of Enterprise Architecture (EA) [24]. This integration is important as the concept of value enables to bridge the gap that exists between the goals that an organization wants to achieve and the processes that are needed to achieve these goals [3]. In other words, the notion of value enables the alignment of the Architecture Vision with the Business

Architecture of an organization [24], which is needed for a company to deliver a positive end-to-end experience to their customers [15].

Despite this growing interest, it is largely recognized that value is a polisemic *term* [4,27] that might refer to several conceptually complex phenomena for which there has not been shared agreement. This issue is evinced in the current proposals to model value in ArchiMate [25]. For instance, value has been described as a goal (e.g. “Being insured” [1], “Anonymity” [8], “Security”[17]), as an object that has value (e.g. “Warehouse Space” [23]), as an event (e.g. “Payment” [1]), and as a capability (e.g. “Computer Skills” [23]).

To address this ambiguity, a value modeling approach for ArchiMate should be based on a proper ontological theory, which provides adequate real-world and formal semantics for such a language’s vocabulary [10]. In particular, we make use of the concepts and relations defined in the Common Ontology of Value and Risk (COVER) [21] (Section 2.2), a novel well-founded reference ontology that explains value and risk as two ends of the same spectrum. COVER is grounded on several theories from marketing, service science, strategy and risk management. It is specified in OntoUML [10] and thus, compliant with the meta-ontological commitments of the Unified Foundational Ontology (UFO) [10].

Based on COVER, we propose a Value Pattern Language (VPL) for ArchiMate that consists of a set of interrelated modeling patterns. ArchiMate was chosen as it is a widely used modeling standard in the EA field, which is also aligned to the TOGAF standard [26]. The advantage of a pattern language [5] 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 VPL according to a first cycle of Design Science Research [13]. As a first step in the design, a set of requirements is identified for the language (Section 3.1). These requirements ensure that the contribution of this paper is clear and verifiable, and they are needed for a formal evaluation of the language [13]. Afterwards the individual modeling patterns that compose VPL are presented (Section 3.3), as well as method for combining them (Section 3.4). We demonstrate how the VPL can be used using the case example of a low-cost airline (Section 4). The actual evaluation of the VPL is outside the scope of this paper, but it will be addressed by future research.

2 Research Baseline

2.1 ArchiMate

ArchiMate is a modeling standard to describe the architecture of enterprises [25]. The language is organized in six layers, namely Strategy, Business, Application, Technology, Physical, and Implementation & Migration [25]. For this paper, only elements of the Strategy and Business layers are particularly relevant. Each element is classified in the language according to its nature, referred to as “aspect” in ArchiMate: a Behavior Element represents *a unit of activity performed by one or more active structure elements*, 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 Motivation Element is one that *provides the context of or reason behind the architecture of an enterprise*, and a Composite Element is simply one that aggregates other elements. Table 1 lists the most relevant ArchiMate elements and relations for the VPL. The underlying logic for the relevance of each concept in this paper can be found in Section 3.2. We refer the reader to the ArchiMate specification for a detailed definition of the concepts [25], while their concrete syntax can be inferred from the patterns in Section 3.3.

Table 1. Overview of the relevant ArchiMate concepts for the VPL.

Type	Elements
	Concepts
Motivation	Stakeholder, Driver, Assessment, Goal, Value
Structure	Resource
Behavior	Capability, Business Process, Business Interaction, Business Event
Composite	Grouping
	Relations
Structure	Composition, Realization
Dependency	Influence
Dynamic	Triggering
Other	Association

2.2 COVER: Common Ontology of ValuE and Risk

The Common Ontology of ValuE and Risk (COVER) [21] formalizes a particular sense in which the term value is used, namely that of *use value*. Briefly put, use value is the quality that summarizes a utility assessment of an object or experience from the perspective of a given subject. This is the meaning of value in sentences such as “A waterproof jacket is valuable when in a rainy city” and “A messenger app that no one uses is of no value to anyone”. The notion of use value should not be confused with those of exchange and ethical value, which are also frequently used in daily life. The former refers to the worth of something in the context of an exchange and is usually measured in monetary terms (e.g. a startup valued at €1.000.000). The latter refers to a high-level constraint that guides the behavior of individuals, as in “one of Google’s core values is that Democracy on the web works”.

COVER makes the following ontological commitments on the nature of value:

- **Value emerges from impacts on goal.** Value emerges from events that affect the degree of satisfaction of one or more goals of an agent. For example, sunscreen is valuable to a tourist in a hot summer day at the beach, as it allows to achieve the goal of protection from ultraviolet radiation—and thus premature aging.

- **Value is neither “good” or “bad”.** Even though people intuitively assume a positive connotation for the term value, use value emerges from events that impact goals either positively or negatively. For instance, consider an event in which Vittoria drops and breaks her new phone. Assuming she had the goal of keeping it intact so she could text her friends, the break event has hurt her goal, and thus has a negative value for her.
- **Value is relative.** The same object or experience may be valuable to a person and of no value to another. For instance, a cigarette has value for a smoker and virtually no value for a non-smoker.
- **Value is experiential.** Even though value can be ascribed to objects, it is ultimately grounded on experiences. For instance, in order to explain the value of a smartphone, one must refer to the experiences enabled by it. These could include sending a text message, watching a video, or paying a bill via a banking app. Then, by valuating each experience and aggregating them according to a given function, one can “compute” the smartphone’s value.
- **Value is contextual.** The value of an object can vary depending on the context in which it is used. Consider a winter jacket, for instance. If worn in a cold evening in the Italian Dolomites, it creates value by protecting one from the cold. Conversely, if worn on a warm day, it is of little use.

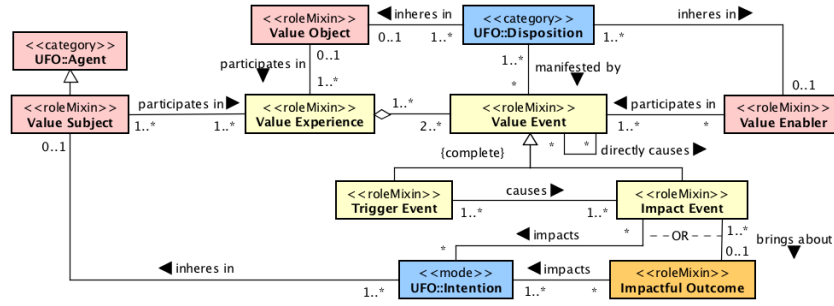


Fig. 1. A fragment of COVER depicting value experiences, their parts and participants.

The aforementioned ontological commitments are captured in the COVER diagrams presented in figures 1 and 2¹. The former is centered around the experiences that create value. It depicts the VALUE SUBJECT class as the role played by an AGENT from whose perspective a value ascription is made. If the target of such an ascription is an object, it is said to play the role of a VALUE OBJECT. Conversely, if the target is an event, it is said to play the role of a VALUE EXPERIENCE. Naturally, a VALUE EXPERIENCE involves VALUE SUBJECTS and VALUE OBJECTS as participants. Additionally, it can also involve other objects, which are labelled VALUE ENABLERS. These allow the ontology to represent participants which contribute to or are necessary for an experience, but are not the

¹ We adopt the following color coding in the OntoUML diagrams presented in this paper: events are represented in yellow, objects in pink, qualities and modes in blue, relators in green, and situations in orange.

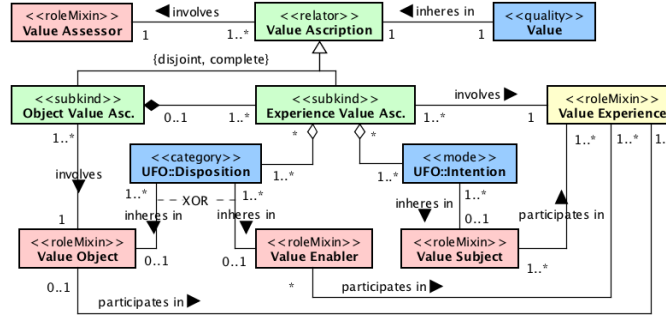


Fig. 2. A COVER fragment formalizing relationships of value ascription.

focal targets of a given valuation. Examples include a browser application, which someone needs to navigate on the internet using a computer, or a road, on which someone drives a car. COVER breaks down VALUE EXPERIENCES into “smaller” events, dubbed VALUE EVENTS. These are classified into IMPACT and TRIGGER EVENTS. The former are those that directly impact a goal or bring about a situation (named IMPACTFUL OUTCOME) that impacts a goal. On contrast, TRIGGER EVENTS are simply parts of an experience that are identified as causing IMPACT EVENTS, directly or indirectly. To formalize goals, COVER reuses the concept of INTENTION from UFO [11], as a type of mental state that describes a class of state-of-affairs that an agent, the VALUE SUBJECT in our case, is committed to bring about. Note that, since agents in UFO’s view includes both physical and social agents, we are able to represent value being ascribed from the perspective of a customer and an employee, but also from that of a business unit or even a whole enterprise.

The diagram in Figure 2 is centered around the VALUE ASCRIPTION relationship, which represents an assessment made by an AGENT, the VALUE ASSESSOR, that “attaches” a quality VALUE to a given VALUE OBJECT or EXPERIENCE from the perspective of a VALUE SUBJECT. As COVER commits to grounding value on experiences, it distinguishes between OBJECT- and EXPERIENCE VALUE ASCRIPTION relationships, with the former being composed by the latter.

2.3 On Patterns and Pattern Languages

A *modeling pattern* describes a situation-independent well-proven solution to a recurring modeling problem. Its use favors the reuse of encoded experiences and good practices. As discussed in depth in [7], a particular modeling pattern of interest is an Ontology Design Pattern (ODP). As demonstrated in [7], ODPs can be systematically extracted from so-called core ontologies, i.e., ontologies that capture phenomena that are recurrent in a number of domains.

As pointed out by Alexander et al. [2], each pattern can exist only to the extent that it is supported by other patterns. According to Schmidt et al. [22], in Software Engineering (SE), the trend is defining pattern languages, rather than stand-alone patterns. The term “pattern language” in SE refers to a network of interrelated patterns that defines a process for systematically solving

coarse-grained software development problems [22]. In [7], the authors make a case demonstrating the viability and benefits of this approach for conceptual model engineering. As shown there, from a core ontology, one can systematically extract a set of ODPs as well as their ties (comprising relations of aggregation, precedence, dependence, mutual exclusion, etc.). Languages that prescribe how ODPs extracted from the same core ontology can be used together are termed Ontology Pattern Languages (OPLs). The method proposed in [7] has been successfully employed to construct OPLs for the modeling of Enterprises, Services, Software Processes, among others. Following this method, in next section, we propose a Pattern Language for Value Modeling.

3 A Pattern Language for Value Modeling

3.1 Language Requirements

In the context of modeling language design, it is useful to identify two types of requirements. The first, named an *analysis requirement*, refers to what the models produced with the language should help users to achieve, either by means of automated or manual analysis. The second, named an *ontological requirement*, refers to the concepts and relations the language should have in order to accurately represent its domain of interest and thus support its intended uses. Let us consider a case for ArchiMate to exemplify these notions. By allowing the representation of how the various elements of an architecture are related, such as services being realized by business processes, which in turn are supported by applications (an ontological requirement), ArchiMate allows users to perform an impact-of-change analysis (an analysis requirement) [16].

For the VPL, we established the following *analysis requirements*:

- R1.** *Design-time value analysis:* An enterprise should be able to understand how it creates value for a given stakeholder, as well as identify opportunities to improve its offerings so that it can maximize value creation.
- R2.** *Run-time value analysis:* An enterprise should be able to identify which indicators it needs to monitor value creation for a given stakeholder, so that it can detect deviations from planned experiences, as well as identify opportunities for innovation.
- R3.** *Competitive analysis:* By modeling the value experiences an enterprise offers to its customers and those of its competitors, an enterprise should be able to identify its competitive advantages.

Given that we are leveraging COVER for the design of the VPL, its ontological requirements are fairly straightforward, i.e., they consist of an isomorphic representation of all concepts and relations defined in the ontology (as argued in [10]). In addition to the aforementioned requirements, we assumed the following *constraints* for the VPL:

- R4.** It should rely exclusively on constructs available in ArchiMate 3.0.1[25]. This is to avoid adding to the complexity of the language.

- R5.** It should map value-related concepts into ArchiMate constructs maintaining, as much as possible, their original meaning as described in the standard. Stereotypes should only be used if strictly necessary to refine the meaning of particular constructs.

3.2 Mapping

Table 2 shows the mapping of COVER concepts into ArchiMate elements.

Table 2. Representation of value-related concepts in ArchiMate.

Concept	Representation in ArchiMate
Value Subject	Stakeholder
Value Object	Structure Element connected to a «ValueExperience»
Value Enabler	Structure Element connected to a Value Event
Value Experience	«ValueExperience» Grouping
Value Event	Business Process, Business Interaction, Business Event
Disposition	Capability
Quality	«Quality» Driver
Intention	«QualityGoal» Goal, «FunctionalGoal» Goal
Value	Value
Value Assessor	Stakeholder connected to a «Valuation»
Object Value Asc.	«Valuation» Assessment connected to a Value Object
Experience Value Asc.	«Valuation» Assessment connected to a «ValueExperience»
Likelihood	«Likelihood» Assessment connected to a triggering association between Value Events or to a «ValueExperience»

3.3 Value Modelling Patterns

Value Object. This pattern allows modelers to express which object will be the focus of a valuation, as well as which kind of experiences enabled by the object are being considered to deduce its value. Its generic structure is depicted on the left side of Fig. 3. It consists of a STRUCTURE ELEMENT—the Value Object—connected to a «ValueExperience» GROUPING that realizes a «FunctionalGoal» GOAL and for which there is a «Likelihood» ASSESSMENT. The likelihood element allows modelers to represent how frequent an experience involving the value object will occur. This assessment serves as a “weight” to the overall use value of an object.

We suggest that modelers represent one experience per type of goal that someone can accomplish with a given value object. For instance, if the value object under analysis is a car, the experiences could include driving it to work, to travel, or to buy groceries. If relevant, modelers can also represent multiple experiences that fulfill the same goal, but that take place in different contexts, such as travelling by car through highways or through dirt roads in the countryside.

Value Experience. This pattern refines the former w.r.t detailing value experiences. As depicted on the right side of Fig. 3, it consists of «ValueExperience» Grouping connected to a STAKEHOLDER acting as the value subject, and its decomposition into value events, which can be represented using BUSINESS PROCESSES, BUSINESS EVENTS and/or BUSINESS INTERACTIONS.

This pattern is neutral with regard to the level of detail at which an experience is modelled. For instance, let us consider the experience of a football fan watching a match at a stadium. It could simply include the events of going to the stadium, watching the match, and then going home. Alternatively, it could be further detailed to account for the ticket purchase, the movement within the stadium to find one’s designated seat, and the consumption of food and drinks. Still, note that the more an experience is detailed, the more accurate the description is of how value is created, and thereby, more insights can be obtained.

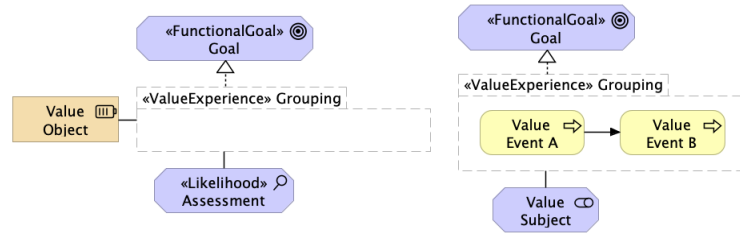


Fig. 3. Generic structures of the Value Object and Value Experience patterns.

Value Subject. In order to account for what creates value for a given stakeholder, we introduce the Value Subject Pattern, as depicted in Fig. 4. It allows one to represent every relevant GOAL of a STAKEHOLDER, as well as to specify their importance by means of a numeric REWARD attribute (represented in the figure between brackets for clarity). Adding “weights” to goals is a modeling strategy that has shown to be very useful for analyzing models, such as the optimization algorithms proposed in [18]. Moreover, to represent the various goals of a value subject in a more compact manner, we propose to represent them within a «Motivation» GROUPING associated to their owner. This modeling strategy is directly inspired by those proposed in goal modeling languages, such as i* [28].

Note that we differentiate between functional and quality goals, following the semantics proposed in [12]. Simply put, functional goals refer to what change in the state-of-affairs an agent wants to bring about, while a quality goal refers to how this change should occur. For instance, traveling to a destination is a functional goal, while doing so in less than two hours is a quality goal. One should note that this distinction is not equivalent to that of hard and soft goals.

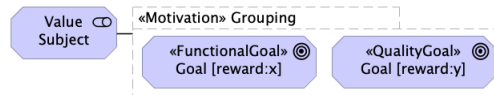


Fig. 4. Generic structure of the Value Subject Pattern.

This second classification refers to how clearly a goal is defined, and thus, is orthogonal to the former. Our previous example of traveling under two hours is considered a hard goal, whilst traveling quickly is a soft goal.

We are aware that providing concrete values for rewards given by goals is challenging. Such numbers may feel artificially chosen and contradictions may quickly arise. Nonetheless, it is fundamental to be able to articulate the motivations driving different stakeholders. To explain why, let us consider customers of low-cost and regular airlines. All of them want to reach their destination, have a comfortable trip, and minimize their financial efforts. However, customers of low-cost airlines prefer to minimize their financial efforts over having a comfortable trip, i.e. they ascribe a higher reward to the former goal than to the latter. To help modelers define these rewards, we suggest the use of prioritization techniques, such as the Analytic Hierarchy Process (AHP) [20].

Value Event. Given that we described the motivation driving a value subject, we go back to the description of value experiences. In order to account for how parts of an experience affect goals, and thus, increase or reduce its value, we propose the Value Event Pattern. Its three variants are presented in Fig. 5.

The first variant, depicted in Fig. 5a, has a very simple structure, consisting of a value event associated to a «FunctionalGoal» GOAL by means of a REALIZATION relation. One could use this variant to represent that the event of watching a movie realizes the goal of being entertained.

The second variant, depicted in Fig. 5b, has a more complex structure. It consists of a value event, a «QualityGoal» GOAL, a «Quality» DRIVER and a quality ASSESSMENT. The value event INFLUENCES the GOAL (either positively or negatively) because the magnitude of one of its qualities is directly related to the satisfaction of the goal. For instance, consider the event of waiting in line at the post office. Since most people want to minimize the time they waste doing chores, its satisfaction is directly related to the duration of the waiting event.

In the third variant, depicted in Fig. 5c, the quality related to the satisfaction of a goal does not inhere in the event, but in one of its participants. Let us consider again the post office case. One’s value perception of such an experience is also influenced by qualities like the politeness of the post office attendants, the number of seats available in the waiting room, and the number of complaints being made by other customers.

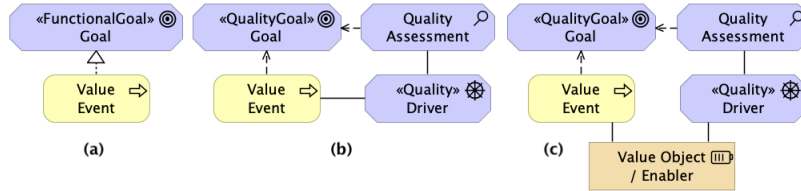


Fig. 5. The three variants of the Value Event pattern.

Disposition. This pattern further characterizes value events, in the sense of accounting for what allows them to happen (ontologically, events are always

manifestations of dispositions). As shown on the left side of Fig. 6, it consists on modelling the dispositions (i.e., CAPABILITIES in ArchiMate) whose manifestations are the value events, as well as the value objects or enablers, in which these dispositions inhere. This pattern allows one to represent that a banking app has capabilities to enable customers to check their balance and make payments. It also allows the representation of dispositions that are manifested as unwanted value events, such as a car that has a disposition to overheat.

Causality. This pattern connects value events that composed a value experience to allow its characterization as an ordered sequence of steps. Its general structure, depicted on the right side of Fig. 6, consists of two value events connected by a TRIGGERING relation, for which a «Likelihood» ASSESSMENT is made. This means that a value event has a probability to cause (or be followed by) another value event. To exemplify this pattern, let us consider the experience of using an on-demand video streaming service. A modeler could use this pattern to represent that after choosing a movie on the platform library, a viewer actually watches it, or that while watching, there is a chance that the viewer dislikes it and then proceeds to search for an alternative content to watch.

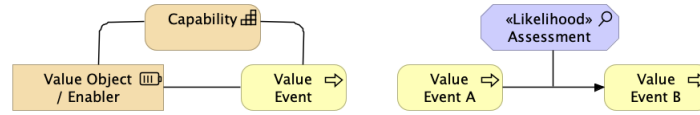


Fig. 6. Generic structures of the Disposition and the Causality patterns.

Experience Valuation. This pattern allows modelers to describe value judgments made towards experiences. As shown in Fig. 7, it consists of a «Valuation» ASSESSMENT made by a STAKEHOLDER that a «ValueExperience» GROUPING creates VALUE for another STAKEHOLDER. Note that the actual VALUE element here is not described in textual terms, but rather as an entry in a scale chosen by the modeler. Just as we are used to see in risk management methodologies, value can be described in a discrete scale like $\langle Low, Medium, High \rangle$ or in a continuous scale like $\langle \text{from } 0.00 \text{ to } 100.00 \rangle$.

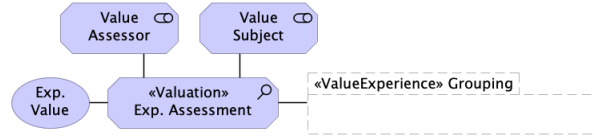


Fig. 7. Generic structure of the Experience Valuation pattern.

Object Valuation. The last VPL pattern is very similar to the previous one, as it also represents a value judgment. The difference is that the judgment is made towards a value object, as seen in Fig. 8. As we previously discussed, the value ascribed to objects is computed from the experiences they afford. Thus, a «Valuation» ASSESSMENT associated to a value object is composed by «Valuation» ASSESSMENTS associated to «ValueExperience» GROUPINGS that

are associated to the focal value object. We also represent a derived INFLUENCE association between the VALUE attached to each ASSESSMENT, so that we can clearly see the process of value aggregation.

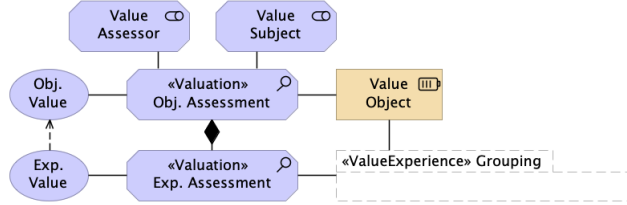


Fig. 8. Generic structure of the Object Valuation pattern.

3.4 Combining the Patterns

To use VPL, a modeler may start with the application of: (i) the Value Object pattern, if the valuation focus is an object (e.g. a product offered by the enterprise or a resource owned by it); (ii) the Value Experience pattern, if the valuation focus is an experience (e.g. a service provided by the enterprise); or (iii) the Value Subject pattern, if the focus is the value perceived by a stakeholder in multiple contexts (e.g. as a partner and as a provider).

When starting with a Value Object pattern, a user may iteratively apply it to account for the relevant ways in which an object can create value, as well as in which contexts it may do so. Each application should be followed by that of the Value Experience. Then, for every experience, the user should iteratively apply the three patterns that detail the inner structure of a value experience: the Value Event, Disposition, and Causality patterns. For each detailed experience, the modeler should apply the Experience Valuation pattern in order to represent its value for the chosen value subject. If one is valuating an object, the Object Valuation pattern should be used to group the experience valuations, and thus derive the aggregate value of the object.

The detailed diagrams presenting the complete process of combining the patterns can be found at <https://github.com/ontouml/vpl>.

4 Case Study

We now present a realistic case study in which we use the VPL to describe how a low-cost airline creates value for its customers. In particular, we model the experience of flying with such a company following a customer journey mapping approach [15], a marketing framework that proposes to map, evaluate and re-design customers' experiences when engaging with companies. Given the limited space available, we only present relevant fragments of the resulting model. The complete case study is available at <https://github.com/ontouml/vpl>.

Since our case focuses on an experience, rather than a product, we start with the application of the Value Experience pattern, as shown in Fig. 9. As the value

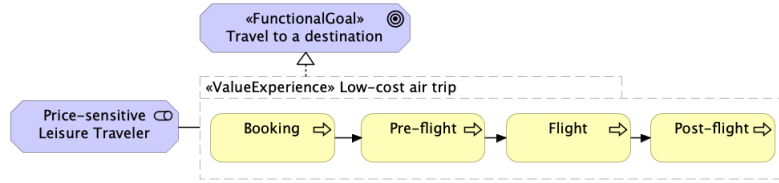


Fig. 9. Usage of the Value Experience pattern.

subject, we use a persona that exemplifies the prototypical customer of a low-cost airline, here named PRICE SENSITIVE LEISURE TRAVELER. Naturally, the main functional goal of air travelling is to TRAVEL TO A DESTINATION, and thus, we represent it as the goal the experience realizes. We also decompose the experience into 4 main steps: BOOKING, PRE-FLIGHT, FLIGHT and POST-FLIGHT.

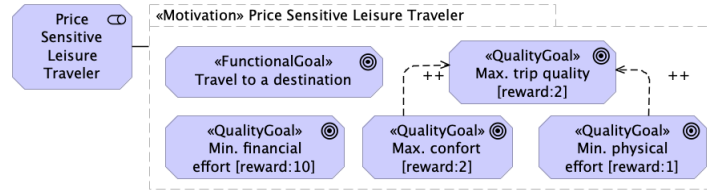


Fig. 10. Application of the Value Subject pattern.

In order to describe how our subject perceives value, we apply the Value Subject pattern, as depicted in Fig. 10. In addition to the travelling goal, we assume that the subject wants to minimize her efforts, both monetary and non-monetary. The former refers to how much she pays to book a flight, choose a seat, and dispatch her luggage, whilst the latter refers to any physical, emotional, cognitive or time effort [6] she has to endure throughout her experience. To each goal, we ascribe a reward between 1 and 10, which reflects their importance to the subject. This prioritization evinces that the subject rather prefers to minimize her financial effort than to enjoy a high quality trip.

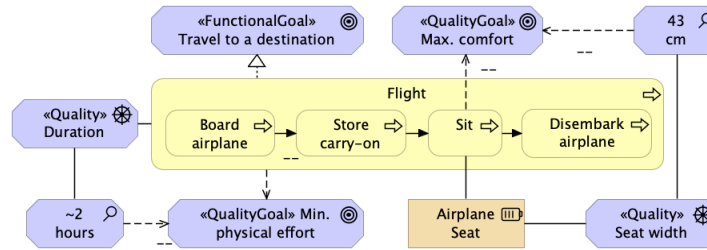


Fig. 11. Application of all three variants of the Value Event pattern.

We now further detail the travelling experience by applying the Value Event pattern three times, as depicted in Fig. 11. We first use the pattern to represent that the FLIGHT process realizes the traveling goal. Then, we apply it to express that the DURATION of the flight is inversely related to the satisfaction of the goal

of minimizing physical effort. Lastly, we apply it to represent that, the WIDTH of the AIRPLANE SEAT negatively impacts the subject’s comfort goal.

Using the Causality pattern, we represent alternative paths the air traveling experience can take. In Fig. 12, we present a refinement of the PRE-FLIGHT step that captures a choice the subject can make after checking in on the flight. She can either download her boarding pass on a smartphone or print it at a totem the airline provides at the airport. Moreover, we use the Disposition pattern to model that the DOWNLOAD BOARDING PASS process depends on the SMARTPHONE, a value enabler, having a CONNECTIVITY disposition to access the internet.

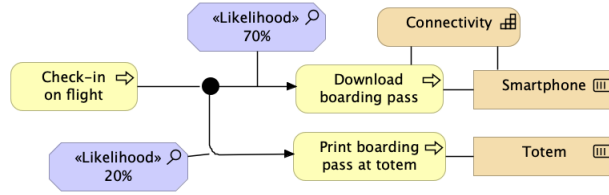


Fig. 12. Usage of the Causality and of the Disposition pattern.

Lastly, we demonstrate the application of the Experience Valuation pattern in Fig. 13. With it, we express that a CUSTOMER EXPERIENCE ANALYST, an employee of the low-cost airline, judges that our subject, the Price Sensitive Leisure Traveler ascribes a High value to the low-cost traveling experience.

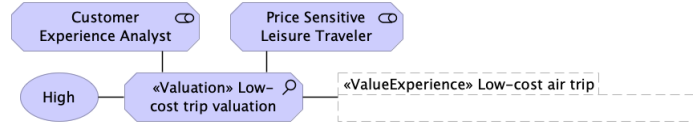


Fig. 13. Application of the Experience Valuation pattern.

5 Related Work

Iacob et al. [14] proposed to extend ArchiMate 2.0 so that it could model strategy- and value-related constructs. As we did, the authors also argue for a relative perspective on value, although it is not clear if they mean taking the perspective of the value subject or the assessor. They also explicitly relate value and risk, although they do not elaborate on the relation itself. Nonetheless, compared to this paper, an interesting aspect of their proposal is to model value from a quantitative perspective, one that is often neglected in the literature.

A second extension was proposed by Aldea et al. [1]. In their work, the authors are clearly concerned with properly grasping the nature of value, as they recognize the difference between use and exchange value. They account for the relative notion of use value by representing it as being perceived by a stakeholder, as well as the relation between value and goals. Nonetheless, the lack of clearly defined ontological foundations hinders the clarity of their proposal, as

the authors seem to contradict themselves by using the value element in the sense of a goal (“Being insured”) and in the sense of a value creating event (“Payment”).

More recently, Feltus and colleagues introduced a value co-creation extension for ArchiMate [8]. They grounded their proposal on the Value Co-creation Ontology, a reference ontology designed from a Service-dominant Logic perspective that formalizes the concept of use value. Despite their more narrow focus on co-creation, the authors also assume a relative view on value, by representing that it benefits some stakeholder. They do distinguish between value, the events that create value, and the participating objects, although their interrelations are not clearly characterized. A noteworthy caveat to their proposal, however, is the omission of goals to characterize value.

Lastly, Singh et al. [23] introduced a method based on *e³value* [9] to design a Value Creation view for ArchiMate. Their work differs significantly from what we propose here, as the conceptual foundation of their work, i.e. *e³value*, aims at modeling the exchange of value objects between economic agents. Their work focuses on *what* has value, as evinced by their use of “Warehouse Space” as a value element, whereas we focus on *what*, *why* and *how* things have value.

6 Final Remarks

In this paper, we introduced VPL, a pattern language for modeling value in ArchiMate that is based on COVER. We presented the first iteration of the build-and-evaluate loop, which consisted in the development of the language and its demonstration by means of a case describing the value perceived by customers of a low-cost airline. By deriving the proposed patterns from COVER, we provided clear real-world semantics for its constituting elements, thus reducing the ambiguity and conceptual complexity found in previous approaches. In particular, we can represent: (i) *what* we can ascribe value to—objects and events; (ii) *why* these have value—because of their impact on goals; (iii) *how* value emerges—by means of value experiences that are composed of value events; and (iv) *who* participates in a value ascription—value subjects and assessors.

As future work, we plan to evaluate VPL by means of case studies in which we will assess: (i) its suitability to model the value domain, (ii) the usability and completeness of the patterns, (iii) the adequacy of the pattern application process, and (iv) the scalability of the language. Moreover, we want to further explore the business insights one may get from using VPL.

References

1. Aldea, A. et al.: Modelling value with ArchiMate. In: International Conference on Advanced Information Systems Engineering. pp. 375–388. Springer (2015)
2. Alexander, C. et al.: A Pattern Language: Towns, Buildings, Construction. Oxford University Press (1977)
3. Andersson, B., Johannesson, P., Zdravkovic, J.: Aligning goals and services through goal and business modelling. Inf. Syst. e-Bus. Manag. **7**(2), 143–169 (2009)

4. Boztepe, S.: User value: Competing theories and models. *Int. J. Des.* **1**(2) (2007)
5. Buschmann, F., Henney, K., Schimdt, D.: *Pattern-oriented Software Architecture: on patterns and pattern language*, vol. 5. John Wiley & Sons (2007)
6. Clark, M., Bryan, A.: *Customer effort: Help or hype?* Henley Business school (2013)
7. Falbo, R.A. et al.: *Ontology pattern languages*. In: *Ontology Engineering with Ontology Design Patterns: Foundations and Applications*. IOS Press (2016)
8. Feltus, C., Proper, E.H., Haki, K.: Towards a language to support value cocreation: An extension to the ArchiMate modeling framework. In: *Federated Conference on Computer Science and Information Systems*. vol. 15, pp. 751–760. IEEE (2018)
9. Gordijn, J., Akkermans, J.: Value-based requirements engineering: exploring innovative e-commerce ideas. *Requirements engineering* **8**(2), 114–134 (2003)
10. Guizzardi, G.: *Ontological foundations for structural conceptual models*. Telematics Instituut, Enschede, The Netherlands (2005)
11. Guizzardi, G., Falbo, R., Guizzardi, R.S.S.: Grounding software domain ontologies in the Unified Foundational Ontology (UFO). In: *11th Ibero-American Conf. on Software Engineering (CibSE)*. pp. 127–140 (2008)
12. Guizzardi, R.S. et al.: An ontological interpretation of non-functional requirements. In: *Int. Conf. on Formal Ontology in Information Systems*. pp. 344–357 (2014)
13. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. *MIS Q.* **28**(1), 6 (2008)
14. Iacob, M.E., Quartel, D., Jonkers, H.: Capturing business strategy and value in enterprise architecture to support portfolio valuation. In: *16th International Enterprise Distributed Object Computing Conference*. pp. 11–20. IEEE (2012)
15. Kalbach, J.: *Mapping experiences: A complete guide to creating value through journeys, blueprints, and diagrams*. O'Reilly Media, Inc. (2016)
16. Lankhorst, M. et al.: *Enterprise Architecture at Work: Modelling, Communication and Analysis*. The Enterprise Engineering Series, Springer (2013)
17. Meertens, L. et al.: Mapping the business model canvas to ArchiMate. In: *ACM Symposium on Applied Computing*. pp. 1694–1701. ACM (2012)
18. Nguyen, C.M., Sebastiani, R., Giorgini, P., Mylopoulos, J.: Multi-objective reasoning with constrained goal models. *Requirements Engineering* **23**(2) (2018)
19. Object Management Group (OMG): *Value delivery metamodel v1.0* (2015)
20. Roelens, B., Steenacker, W., Poels, G.: Realizing strategic fit within the business architecture: the design of a process-goal alignment modeling and analysis technique. *Softw. Syst. Model.* pp. 1–32 (2017)
21. Sales, T.P., Baião, F., Guizzardi, G., Almeida, J.P.A., Guarino, N., Mylopoulos, J.: The common ontology of value and risk. In: *International Conference on Conceptual Modeling*. vol. 11157, pp. 121–135. Springer (2018)
22. Schmidt, D.C., Stal, M., Rohnert, H., Buschmann, F.: *Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects*. Wiley (2000)
23. Singh, P.M. et al.: Modeling value creation with enterprise architecture. In: *Int. Conf. on Enterprise Information Systems (ICEIS)*. vol. 3, pp. 343–351 (2014)
24. Svee, E.O., Zdravkovic, J.: Extending enterprise architectures to capture consumer values: the case of togaf. In: *International Conference on Advanced Information Systems Engineering*. pp. 221–232. Springer (2015)
25. The Open Group: *ArchiMate v3.0.1 Specification*. Standard C179 (2017)
26. The Open Group: *TOGAF v9.2* (2018)
27. Vargo, S.L., Maglio, P.P., Akaka, M.A.: On value and value co-creation: A service systems and service logic perspective. *Eur. Manag. J.* **26**(3), 145–152 (2008)
28. Yu, E., Giorgini, P., Maiden, N., Mylopoulos, J.: *Social modeling for requirements engineering*. MIT Press (2011)