OntoUML Quick Guide and Pattern Language

Ricardo de Almeida Falbo

1. Introduction

The concepts that are assumed to exist in some area of interest and the relationships that hold among them are said a conceptualization. A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose. An ontology is a formal, explicit specification of a shared conceptualization (STUDER et al., 1998).

There is a growing interest in the Conceptual Modeling community to rely on ontologies for designing Conceptual Modeling Languages, in an approach called Ontology-Driven Conceptual Modeling (GUIZZARDI, 2005). This is the case of OntoUML, an ontologically well-founded profile for UML 2.0 class diagrams (GUIZZARDI, 2005). OntoUML incorporates ontological distinctions of the Unified Foundational Ontology (UFO) in UML class diagram by means of stereotypes.

The Unified Foundational Ontology (UFO) (GUIZZARDI, 2005) (GUIZZARDI; FALBO; GUIZZARDI, 2008) (GUIZZARDI et al., 2013) is constituted by three main parts. UFO-A is an ontology of endurants (objects) (GUIZZARDI, 2005), UFO-B, an ontology of events (perdurants) (GUIZZARDI; FALBO; GUIZZARDI, 2008), and UFO-C (GUIZZARDI; FALBO; GUIZZARDI, 2008) (GUIZZARDI, 2006), an ontology of social entities built on the top of UFO-A and UFO-B. All of these three parts follow the fundamental distinction in UFO between *individuals* (as entities that exist in reality and possess a unique identity, e.g., John and my car) and *universals* (as patterns of features that can be realized in a number of different individuals, e.g., the kinds Person, and Car) (GUIZZARDI, 2005). OntoUML is based on UFO-A. Thus, before presenting OntoUML, we need to understand the basic notions put forward by UFO-A.

2. **UFO-A: An Ontology of Endurants**

UFO-A (GUIZZARDI, 2005) is an ontology of *endurants*. *Endurants* are individuals that are wholly present whenever they are present (differently of *events* that are composed of temporal parts), and can be divided into substantials and moments. *Substantials* are existentially independent endurants (e.g., a person, a car). *Moments* are individuals that can only exist in other individuals, and, thus, they are existentially dependent on their bearers (e.g., John's weight, my car's color).

Endurants patterns of features are called *Endurant Universals*. Figure 1 presents the Endurant Universals hierarchy of UFO. **Endurant Universals** are distinguished into **Substantial Universals** and **Moment Universals**. Naturally, these are kinds of universals whose instances are substantial individuals and moments, respectively.

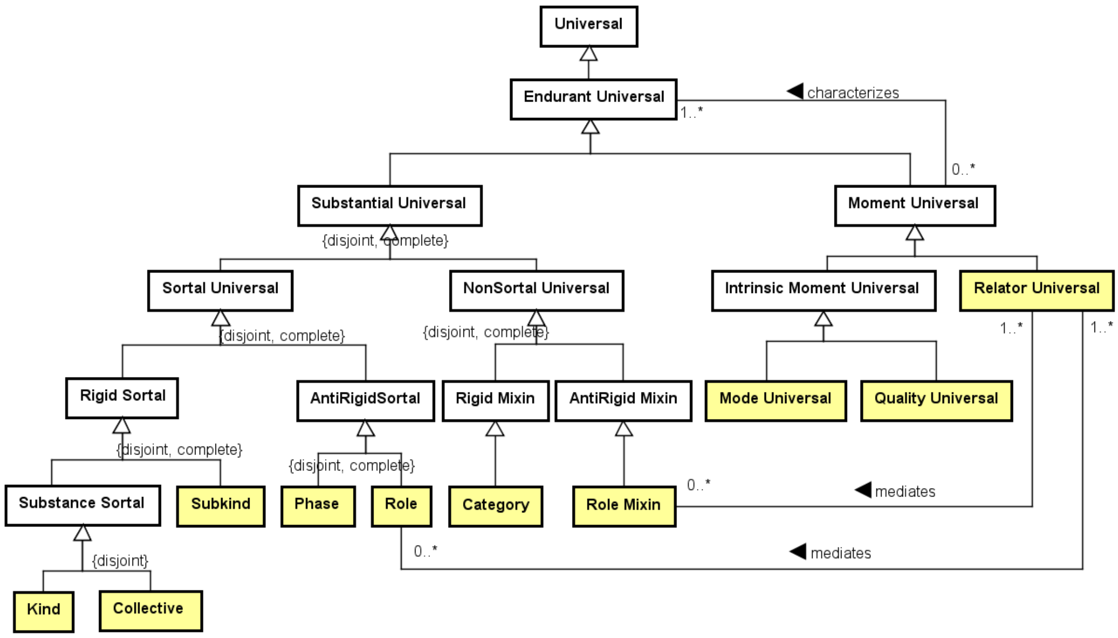


Figure 1. A UFO-A Fragment (Endurant Universals).

Concerning the Substantial Universal hierarchy, **Sortal Universals** are the ones that either provide or carry a uniform *principle of identity* for their instances. A principle of identity supports the judgment whether two individuals are the same. In particular, it also informs which changes an individual can undergo without changing its identity. **Non-Sortals Universals** are universals that aggregate properties of distinct Sortals, i.e., they can have as instances individuals obeying different principles of identity.

Within the category of **Sortal Universals**, we have the distinction between *rigid and anti-rigid* universals. A rigid universal is one that classifies its instances necessarily (in the modal sense), i.e., the instances of that universal cannot cease to be so without ceasing to exist. Anti-rigidity, in contrast, characterizes a universal whose instances can move in and out of its extension without altering their identity. For instance, contrast the rigid universal *Person* with the anti-rigid universals *Student* or *Husband*. While the same individual *John* never ceases to be instance of *Person*, he can move in and out of the extension of *Student* or *Husband*, depending on whether he *enrolls in*/*finishes college* or *marries*/*divorces*, respectively.

Concerning the distinctions between *rigid sortals* in UFO, **Kinds** are sortal rigid universals that provide a uniform principle of identity for their instances (e.g., *Person*). **Collectives** are sortal rigid universals that represent collections of individuals with uniform structure (e.g., deck of cards, a forest, a group of people, a pile of bricks). This universal provides a principle of identity for the collection (but not for every individual in the collection). **Subkinds** are sortal rigid universals that carry the principle of identity supplied by a unique **Kind** (e.g., a Kind *Person* can have the Subkinds *Man* and *Woman* that carry the principle of identity provided by *Person*).

A *member-collection relation* is one that holds between a singular entity and a collective. Member collection relations are never transitive, i.e., they are intransitive. E.g., I am member of a club (collection) and my club is a member of an International body (a collective). However, it does not follow that I am a member of this International body, since this only has clubs as members, not individuals.

Differently from collectives, *functional complexes* (complex kinds/subkinds) are composed by parts that play a multitude of functions in the context of the whole (e.g., Person has Hearth, Head, Brain, Eyes and so on). The parts of a complex have in common that they all posses a functional link with the complex. In other words, they all contribute to the functionality (or the behavior) of the complex. The *component-of relation* is the one that holds between a functional complex and its parts. Component-of relations are always transitive (e.g., Brain is part of Head, and Head is part of Person. Then, Brain is part of Person).

Concerning *anti-rigid sortals*, we have the distinction between **Roles** and **Phases**. **Phases** are relationally independent universals defined as a partition of a sortal. This partition is derived based on an intrinsic property of that universal (e.g., *Child* is a phase of *Person*, instantiated by instances of person who are less than 12 years). **Roles** are relationally dependent universals, capturing relational properties shared by instances of a given kind, i.e., putting it baldly: entities play roles when related to other entities (e.g., *Student*, *Husband*). Since the principle of identity is provided by a unique **Kind**, each sortal hierarchy has a unique **Kind** at the top. The relational dependence of **Roles** is manifested by the presence of a **Relator** in the model. **Relators** are connecting entities. For example, an *Enrollment* relator connects a *Student* role with an *Educational Institution*.

Non-Sortals Universals are universals that aggregate properties that are common to different Sortals, i.e., that ultimately classify entities that are of different **Kinds**. Non-Sortals do not provide a uniform principle of identity for their instances; instead, they just classify things that share common properties but which obey different principles of identity. *Furniture* is an example of Non-Sortal (a Category) that aggregates properties of *Table*, *Chair* and so on. Other examples include Works of Art (including paintings, music compositions, statues), insurable items (including works of art, buildings, cars, people, etc.) and social and legal objects (including people, organizations).

The meta-properties of rigidity and anti-rigidity can also be applied to distinguish different types of **Non-Sortals (Mixins)**. A **Category** represents a rigid and relationally independent mixin, i.e., a dispersive universal that aggregates essential properties that are common to different rigid sortals (e.g., *Furniture* aggregates essential properties of *Table*, *Chair*, etc.). A **RoleMixin** represents an anti-rigid and externally dependent Non-Sortal, i.e., a dispersive universal that aggregates properties that are common to different **Roles** *(*e.g., *Customer* that aggregates properties of *Person Customer* and *Corporate Customer)*.

Regarding Moment Universals, UFO distinguishes between two main types of moment universals: **Intrinsic Moment Universals** and **Relator Universals**. **Relator Universals**, as said before, are connecting entities (e.g., Employment, Enrollment, Marriage). E.g., the *Marriage* relator universal connects the *Husband* role to the *Wife* role. Every instance of a relator universal is existentially dependent on at least two distinct entities. The formal relation that takes place between a relator universal and the object classes it connects is termed *mediation*.

Intrinsic moments, in turn, are dependent on one single individual (e.g., an apple's color). UFO distinguishes between two main types of intrinsic moment universals: **Quality Universals** and **Mode Universals**. **Quality Universals** are intrinsic moment universals that are associated with a quality structure, i.e. a space of values known (e.g., Color, Weight, Name), while **Mode Universals** are intrinsic moment universals that are not associated with a quality structure (e.g., Desire, Intention, Symptom, Skill).

3. OntoUML

As said before, UFO-A was employed in the design of an ontologically well-founded version of UML 2.0 class diagrams, termed OntoUML. OntoUML has modeling constructs that reflect all the leave categories in the hierarchy of Figure 1. Moreover, its metamodel contains a number of formal constraints derived from the axiomatization of UFO that prescribe that rules that govern the allowed combination of these constructs. These rules constrain possible combination of constructs in subsumption hierarchies (e.g., an anti-rigid universal cannot be a supertype of a rigid universal; a sortal universal cannot be a subtype of a mixin universal; every sortal is either a kind or a direct or indirect subtype of a unique kind); reinforce the necessary disjointness between instances of certain modeling constructs (e.g., all kinds are mutually disjoints; all phases of a given kind must appear in a disjoint, complete generalization set specializing that kind); and reinforce the existence of relators representing the relational dependence of types such as Roles and RoleMixins.

OntoUML is an ontology modeling language that focuses on theoretical soundness and high expressiveness, instead of on computational properties (e.g., computational efficiency and tractability). Therefore, OntoUML is suitable for conceptual modeling. Table 1 presents OntoUML stereotypes. It is important to say that Quality universals are typically not represented in a conceptual model explicitly but via attributes. Thus OntoUML does not provide a stereotype for this construct.

OntoUML can also be seen as a pattern language, i.e., the modeling primitives of the language are actually higher-granularity building blocks (ontology patterns) that reflect the different ontological micro-theories in UFO. In the next section, we present a version of OntoUML in terms of an intuitive form of graph grammar that describes OntoUML as a pattern language.

Table 1. OntoUML Stereotypes

UFO-A Object Type	Stereotype
Kind	<<kind>>
Collective	<<collective>>
Subkind	<<subkind>>
Phase	<<phase>>
Role	<<role>>
Category	<<category>>
RoleMixin	<<roleMixin>>
Relator	<<relator>>
Mode	<<mode>>
UFO-A Relation Type	**Stereotype**
mediation	<<mediation>>
member-collection	<<memberOf>>
component-of	<<componentOf>>

4. OntoUML as a Pattern Language

For each of the ontological distinctions present in UFO and which are reflected as modeling constructs in OntoUML, we have a corresponding axiomatization. This axiomatization makes that OntoUML constructs can only appear in a model forming clusters of constructs with their ties and associated constraints. In other words, the actual modeling primitives of OntoUML are these structures (and their corresponding axiomatization) reflecting the underlying ontological micro-theories. Thus, OntoUML is a *pattern language* whose models are constructed via the combined instantiation of foundational patterns (RUY et al., 2017).

In this section, we present the catalog of OntoUML Patterns in a systematic manner, following a template with the following items:

- **Name**: uniquely identifies the pattern and intends to convey a brief idea of its content.
- **Acronym**: a short name to facilitate the documentation and communication about the pattern.
- **Intent**: describes the pattern purpose, providing a brief discussion of when modelers should apply the model structure identified by the pattern.
- **Rationale**: describes the foundations of the pattern (in terms of UFO).
- **Variants**: describes different ways to apply the pattern. For each variant, the following items are present:
    - **Description**: a natural language description of the structure of the pattern variant.
    - **Pattern Structure**: depicts the structure of the pattern variant.
    - **Related patterns**: indicates other patterns/variants (if any) that should be used in conjunction with this pattern variant.
    - **Example**: an example showing the application of the pattern variant.

The patterns mention expressions included in the OntoUML grammar, referring to other patterns or to more general concepts of UFO. These expressions are shown in Figure 2. Next, each pattern is presented following the template above.

Expression	Expression Structure
Endurant Universal Expression	Substantial Universal Expression or Moment Universal Expression
Substantial Universal Expression	Sortal Expression or Mixin Expression
Sortal Expression	Rigid Sortal Expression or Anti-Rigid Sortal Expression
Rigid Sortal Expression	Substance Sortal Expression or SUBKIND PATTERN
Substance Sortal Expression	<<kind>> T or COLLECTIVE PATTERN
Anti-Rigid Sortal Expression	PHASE PATTERN or ROLE PATTERN
Mixin Expression	CATEGORY PATTERN or ROLEMIXIN PATTERN
Moment Universal Expression	MODE PATTERN or RELATOR PATTERN
Relationally Dependent Universal Expression	ROLE PATTERN or ROLEMIXIN PATTERN

Figure 2. Expressions of OntoUML Grammar.

Table 2 presents the **Subkind Pattern**. As this table shows, this pattern appears in two variants. A subkind can only specialize a rigid sortal. Thus, in the **Single Subkind Variant**, we have simply a subkind specializing a **Rigid Sortal Expression**; while in the **Multiple Subkinds Variant**, we have a subkind generalization set collecting a disjoint (and optionally complete) set of subkinds that specialize the same rigid sortal. It is important to highlight that, as Table 1 shows, **Rigid Sortal Expression** refers to the **Subkind Pattern**. This recursive definition guarantees that a subkind either directly or indirectly specialize a substance sortal that provides a uniform principle of identity for its instances.

Table 2 – The Subkind Pattern

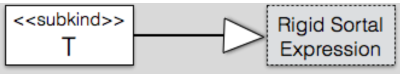
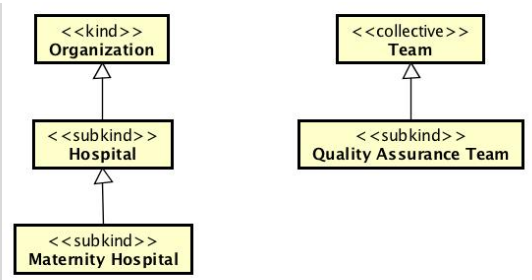
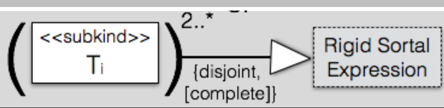
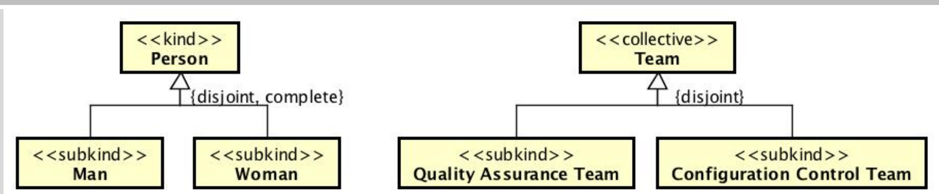
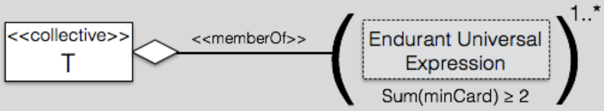
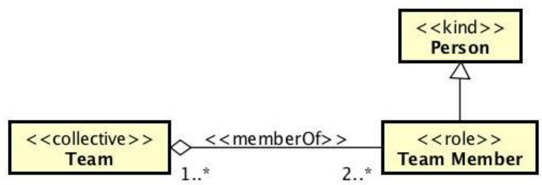
Name (Acronym)	**Intent**
Subkind Pattern (Subkind)	To specialize a kind/collective/subkind into subkinds.
Rationale	
Subkinds are sortal rigid universals that carry the principle of identity supplied by a unique **Substance Sortal**. **Substance Sortal** describes the *identity provider universals*, which can be either a **Kind** or a **Collective**. Subkinds can also specialize other subkinds, however a subkind specializing other subkinds cannot specialize more than one ultimate **Substance Sortal**.	
Variant 1	Single Subkind
Variant Description	A single subkind specializing a **Rigid Sortal Expression** (A **Rigid Sortal Expression** is either a **Substance Sortal Expression** or an occurrence of the **Subkind Pattern**).
Pattern Structure	
	
Related Patterns	-
Examples	
	
Variant 2	Multiple Subkinds
Variant Description	A subkind generalization set collecting a disjoint (and optionally complete) set of subkinds that specialize the same universal **Rigid Sortal Expression**.
Pattern Structure	
	
Related Patterns	-
Examples	
	

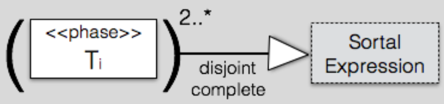
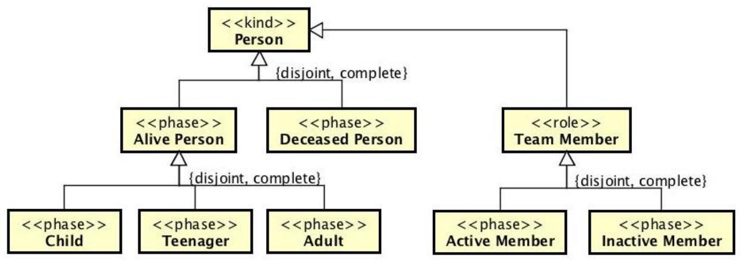
Table 3 presents the **Collective Pattern**. This pattern describes a Collective Universal and the universals whose instances are members of these collectives. The members should be of the same type, represented by the **Endurant Universal Expression**.

Table 3 – The Collective Pattern

<table>
<tr><th>Name (Acronym)</th><th>Intent</th></tr>
<tr><td>Collective Pattern (Collective)</td><td>Describes a collective universal and the universals whose instances are members of these collectives.</td></tr>
<tr><th colspan="2">Rationale</th></tr>
<tr><td colspan="2">Collectives are collections of Endurant Universals (Substantial Universals / Moment Universals) that have a uniform structure, such as a deck of cards, a forest, or a group of people. Due to the so-called Weak Supplementation Principle, required for all parthood relations, the sum of the minimum cardinality constraint on the side of the members must be equal or higher than 2.</td></tr>
<tr><th colspan="2">Pattern Structure</th></tr>
<tr><td colspan="2">
</td></tr>
<tr><th>Related Patterns</th><td>-</td></tr>
<tr><th colspan="2">Example</th></tr>
<tr><td colspan="2">
</td></tr>
</table>

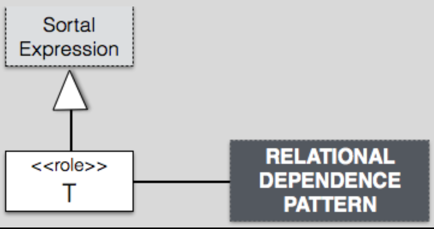
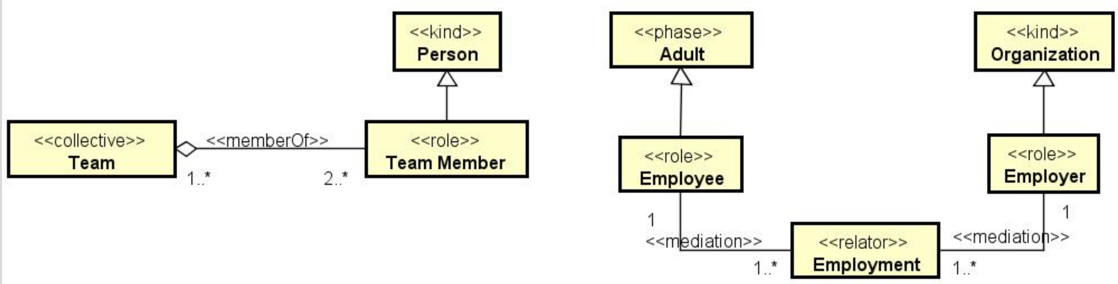
As Table 4 shows, the <u>**Phase Pattern**</u> consists of a phase partition, i.e., a disjoint and complete set of two or more complementary phases that specialize the same sortal, which is specified by a **Sortal Expression**. Notice that, once more, the recursive definition of this pattern (given by the **Sortal Expression**) guarantees that a substance sortal providing a common principle of identity for the instances of these phases is always specified in the model.

Table 4 – The Phase Pattern

<table>
<tr><th>Name (Acronym)</th><th>Intent</th></tr>
<tr><td>Phase Pattern (Phase)</td><td>Represents a phase partition, i.e., a disjoint and complete set of two or more complementary phases that specialize the same sortal.</td></tr>
<tr><th colspan="2">Rationale</th></tr>
<tr><td colspan="2">Phases constitute possible stages in the history of a Sortal (Rigid or Anti-rigid Sortal). They are anti-rigid and relationally independent universals defined as part of a partition of a Sortal. The phases {P, ..., Pn} that form a phase-partition of a sortal S form a disjoint and complete generalization set. This partition is derived based on an intrinsic property of that universal (e.g., Child is a phase of Person, instantiated by instances of person who are less than 12 years).</td></tr>
<tr><th colspan="2">Pattern Structure</th></tr>
<tr><td colspan="2">
</td></tr>
<tr><th>Related Patterns</th><td>-</td></tr>
<tr><th colspan="2">Examples</th></tr>
<tr><td colspan="2">
</td></tr>
</table>

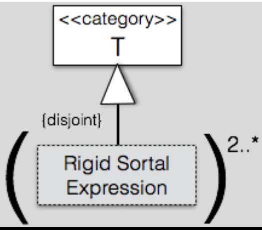
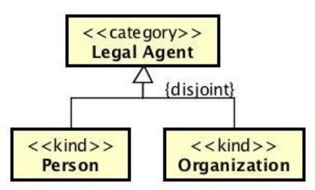
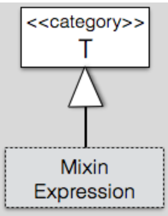
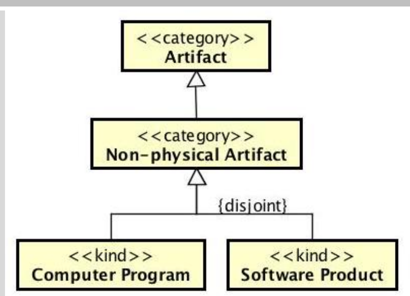
Analogously, in the **Role Pattern** (see Table 5), we have a role that specializes a sortal universal (specified by a **Sortal Expression**). However, since roles are relationally dependent universals, we have also that a role must be part of an occurrence of the **Relational Dependence Pattern**.

Table 5 – The Role Pattern

<table>
<tr><th>Name (Acronym)</th><th>Intent</th></tr>
<tr><td>Role Pattern (Role)</td><td>Represents the role an entity plays in the context of a relationship establishment.</td></tr>
<tr><th colspan="2">Rationale</th></tr>
<tr><td colspan="2">Roles are anti-rigid and relationally dependent Sortals (Rigid or Anti-rigid Sortals), capturing relational properties shared by instances of a given sortal. In other words, entities play roles when related to other entities.</td></tr>
<tr><th colspan="2">Pattern Structure</th></tr>
<tr><td colspan="2">
</td></tr>
<tr><th>Related Patterns</th><th>Justification</th></tr>
<tr><td>Relational Dependence Pattern</td><td>Since roles are relationally dependent universals, a role must be part of an occurrence of the <u>Relational Dependence Pattern</u>. Thus, when applying the <u>Role Pattern</u>, it is mandatory to apply the <u>Relational Dependence Pattern</u>.</td></tr>
<tr><th colspan="2">Examples</th></tr>
<tr><td colspan="2">
</td></tr>
</table>

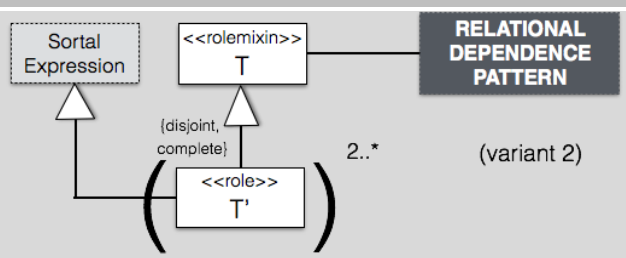
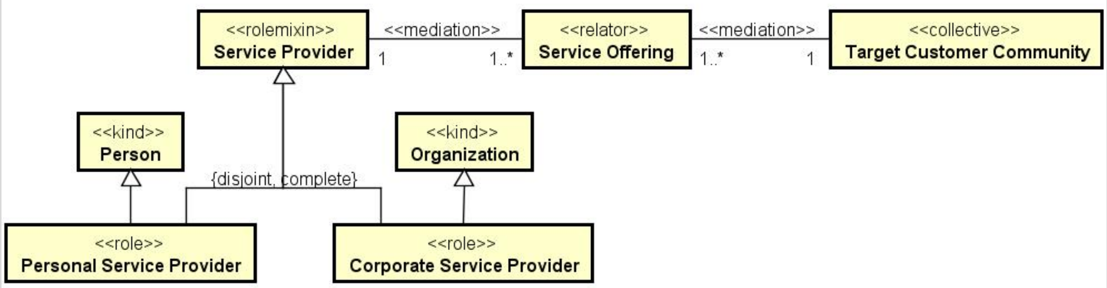
A Category ultimately captures common essential properties of entities of multiple kinds. As Table 6 shows, the **Category Pattern** appears in two variants: (i) the **Category of Rigid Sortals Variant** captures this by directly having a Category as a common abstraction of two or more disjoint **Rigid Sortal Expressions**; (ii) the **Category of Mixins Variant** represents the common essential properties of multiple kinds by indirectly having a Category as a common abstraction of another **Mixin Expression** (either another recursive occurrence of the **Category Pattern** or an occurrence of the **RoleMixin Pattern**), which will, in turn, eventually be connected to a set of **Sortal Expressions**.

Table 6 – The Category Pattern

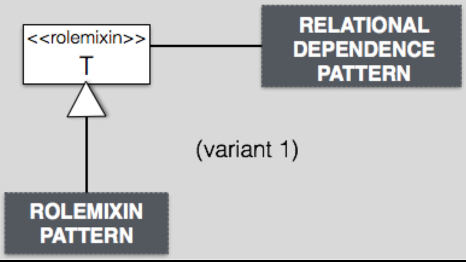
Name (Acronym)	**Intent**
Category Pattern (Category)	Represents a rigid and relationally independent mixin, aggregating essential properties that are common to different rigid sortals.
Rationale	
A **Category** represents a rigid and relationally independent mixin, i.e., a dispersive universal that ultimately aggregates essential properties that are common to different **Rigid Sortals** (**Kinds**, **Subkinds** or **Collectives**).	
Variant 1	**Variant Description**
Category of Rigid Sortals	Represents a **Category** directly aggregating essential properties that are common to different **Rigid Sortals**.
Pattern Structure	
	
Related Patterns	-
Examples	
	
Variant 2	**Variant Description**
Category of Mixins	Represents a **Category** that indirectly aggregates essential properties that are common to different **Rigid Sortals**. In this case, between a **Category** and the **Rigid Sortals** there is a **Mixin** (another **Category** or a **Rolemixin**).
Pattern Structure	
	
Related Patterns	-
Examples	
	

The **RoleMixin Pattern** appears in two variants, as Table 7 shows. The **Rolemixin and Roles Variant** defines a RoleMixin by a partition of two or more Roles, each of which is connected to a kind (directly or indirectly) via a **Sortal Expression**. The common relational dependence of these roles is captured by connecting the RoleMixin to an occurrence of the **Relational Dependence Pattern**. In the **Rolemixin of Rolemixins Variant**, a RoleMixin can appear in a model recursively applying the **RoleMixin Pattern**, i.e., specializing another RoleMixin with its associated relational dependence.

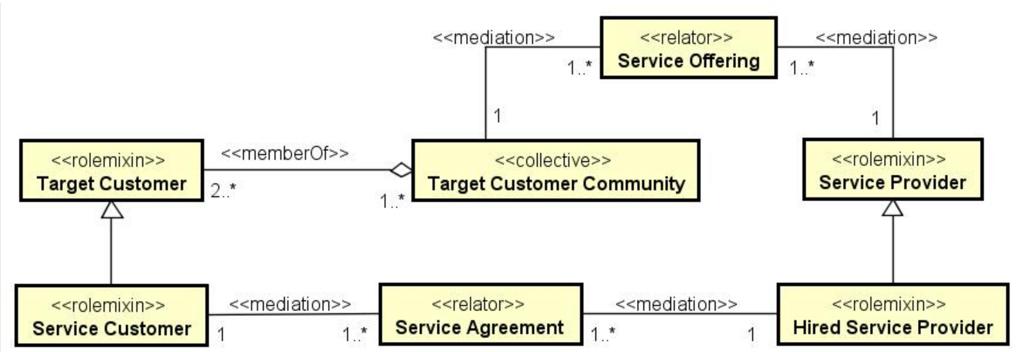
Table 7 – The Rolemixin Pattern

Name (Acronym)	**Intent**
Rolemixin Pattern (Rolemixin)	Represents an anti-rigid and externally dependent mixin, aggregating properties which are common to different roles.
Rationale	
A **RoleMixin** captures common contingent and relationally dependent properties of entities of multiple **Roles** ultimately played by different **Substance Sortals** (**Kinds** or **Collectives**). In other words, a **RoleMixin** can be seen as an abstraction capturing common characteristics of **Roles** played by instances of different **Substance Sortals**.	
Variant 1	**Variant Description**
Rolemixin and Roles	Defines a **RoleMixin** by a partition of two or more **Roles**, each of which is directly or indirectly connected to a **Substance Sortal** (**Kind** or **Collective**) via a **Sortal Expression**. In other words, when directly connected, the **Role** specializes a **Substance Sortal** (**Kind** or **Collective**) providing the identity criteria to it; when indirectly connected, the **Role** specializes a **Subkind** or an **Anti-Rigid Sortal** (**Phase** or **Role**), which, in turn, is directly or indirectly connected to a **Substance Sortal**.
Pattern Structure	
	
Related Patterns	**Justification**
Relational Dependence Pattern	A **RoleMixin** captures common contingent and relationally dependent properties of entities of multiple **Roles**. The common relational dependence of these **Roles** is captured by connecting the **RoleMixin** to an occurrence of the **Relational Dependence Pattern**.
Examples	
	

Variant 2	**Variant Description**
Rolemixin of Rolemixin	A **RoleMixin** captures common contingent and relationally dependent properties of entities of multiple **Roles**. The common relational dependence of these **Roles** is captured by connecting the **RoleMixin** to an occurrence of the **Relational Dependence Pattern**.
Pattern Structure	



Related Patterns	**Justification**
Relational Dependence Pattern	A **RoleMixin** captures common contingent and relationally dependent properties of entities of multiple **Roles**. The common relational dependence of these **Roles** is captured by connecting the **RoleMixin** to an occurrence of the **Relational Dependence Pattern**.
Examples	



The **Relational Dependence Pattern** is a complex pattern that describes the relational dependence condition of a relationally dependent universal (i.e., either a Role or a RoleMixin), which, is specified by a **Relationally Dependent Universal Expression**. As Table 1 shows, a **Relationally Dependent Universal Expression** is either an occurrence of the **Role Pattern** or an occurrence of the **RoleMixin Pattern**. Thus, the relational dependence condition is captured either: (i) via a connection to the **Relator Pattern** (in the **Mediation Relation Variant**), in which the relationally dependent universal at hand appears as one of the mediated types; or (ii) via a parthood relation (in the **Parthood Relation Variant**), in which the relationally dependent universal at hand appears either as a part of or a whole universal.

Table 8 – The Relational Dependence Pattern

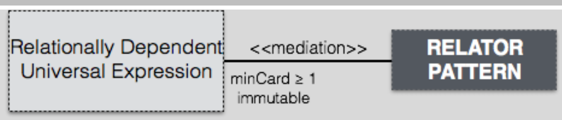
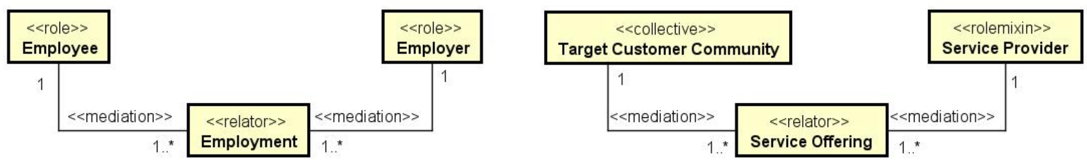
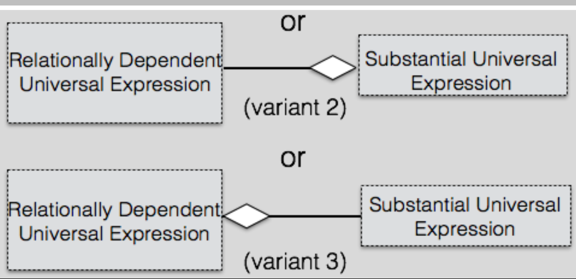
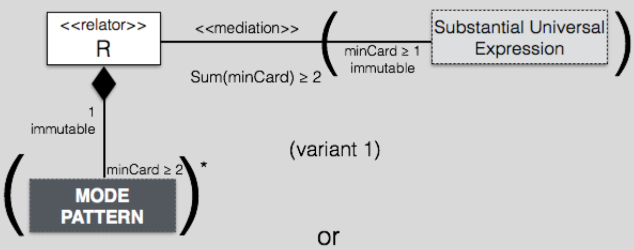
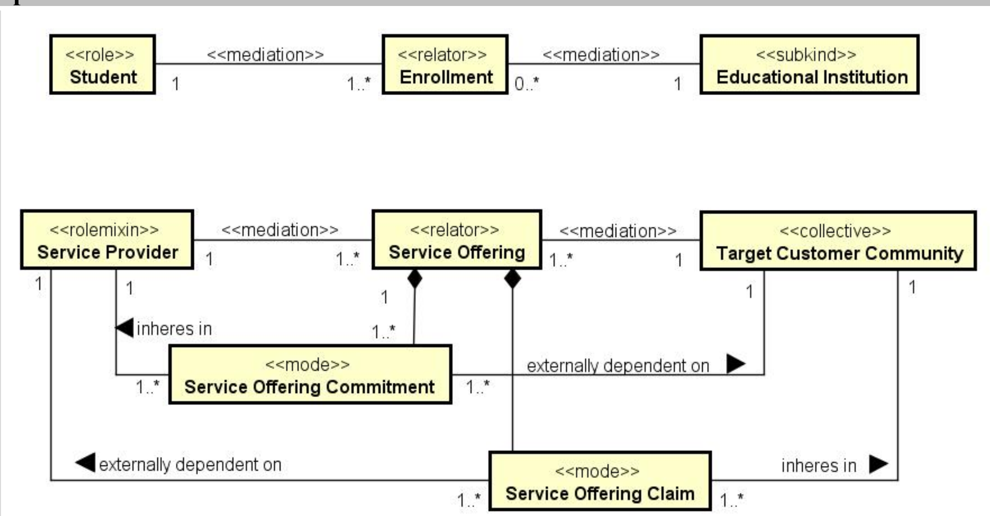
<table>
<tr><th>Name (Acronym)</th><th>Intent</th></tr>
<tr><td>Relational Dependence Pattern (RelDep)</td><td>Describes the relational dependence condition of a relationally dependent universal, i.e., either a Role or a RoleMixin.</td></tr>
<tr><th colspan="2">Rationale</th></tr>
<tr><td colspan="2">Roles and Rolemixins represent relationally dependent universals. Thus, they should be connected to other entities, via material relations, parthood relations, or, in particular, with Relators, via mediation relations. Parthood relations include componentOf, memberOf and subCollectionOf relations. A mediation relation, in turn, is a formal relation that takes place between a Relator and the relationally dependent universal it mediates.</td></tr>
<tr><th>Variant 1</th><th>Variant Description</th></tr>
<tr><td>Mediation Relation</td><td>Describes the relational dependence condition of a relationally dependent universal (a Role or a RoleMixin) by means of a mediation relation with a Relator.</td></tr>
<tr><th colspan="2">Pattern Structure</th></tr>
<tr><td colspan="2">
</td></tr>
<tr><th>Related Patterns</th><th>Justification</th></tr>
<tr><td>Relator Pattern</td><td>Since a mediation relation is a formal relation that takes place between a Relator and the relationally dependent universal it mediates, when applying the <u>Mediation Relation Pattern</u>, the modeler should also apply the <u>Relator Pattern</u>.</td></tr>
<tr><th colspan="2">Examples</th></tr>
<tr><td colspan="2">
</td></tr>
<tr><th>Variant 2</th><th>Variant Description</th></tr>
<tr><td>Parthood Relation</td><td>Describes the relational dependence condition of a relationally dependent universal (a Role or a RoleMixin) by means of a parthood relation, in which the relationally dependent universal at hand appears either as a part of or a whole universal.</td></tr>
<tr><th colspan="2">Pattern Structure</th></tr>
<tr><td colspan="2">
</td></tr>
<tr><th>Related Patterns</th><th>-</th></tr>
<tr><th colspan="2">Examples</th></tr>
<tr><td colspan="2">
</td></tr>
</table>

Table 9 shows the **<u>Relator Pattern</u>** in its simplified representation, which represents relators connected via mediation relations to a number of substantial universals whose instances are entities mediated by this relator.

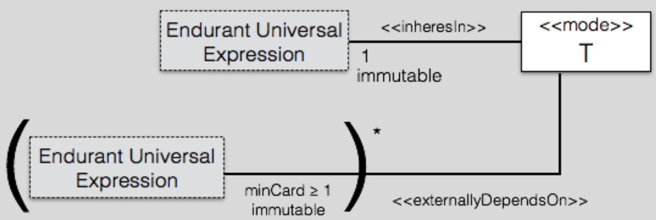
Table 9 – The Relator Pattern

Name (Acronym)	**Intent**
Relator Pattern (Relator)	Represents a material relation reified.
Rationale	
Relators are moments that existentially depend on two or more **Endurants** (e.g, marriages, service agreements, enrollments, employments). They are individuals with the power of connecting entities. Every instance of a relator universal is existentially dependent on at least two distinct entities. The formal relation that take place between a **Relator** universal and the **Substantial Universals** (**Sortal** or **Mixin Universals**) it connects is termed *mediation* (a particular type of *existential dependence* relation). **Relators** are considered to be the truthmakers of *material relations*, and thus *material relations* are said to derive from **Relators**. In the context of a *material relation*, several externally-dependent (i.e., relational) **Modes** come into existence. A **Relator** aggregates all these externally-dependent **Modes** that the **Endurants** involved in the *material relation* acquire in virtue of participating in the corresponding relation. In the UFO literature, relator names are commonly nominalizations of the verb that expresses the underlying *material relation* (e.g., married-to/ marriage).	
Variant 1	**Variant Description**
Relator Simplified Representation (SR-Relator)	In the first variant, we have a representation of **Relators** connected via *mediation relations* to possibly a number of **Substantial Universals** whose instances are entities mediated by this **Relator**. Optionally, the externally-dependent (i.e., relational) **Modes** that constitute the **Relator** can also be represented.
Pattern Structure	
	
Related Patterns	**Justification**
Mode Pattern	When modeling the externally-dependent **Modes** is required, the **<u>Mode Pattern</u>** should be applied.
Examples	
	

Finally, the **<u>Mode Pattern</u>** (see Table 10) represents a Mode Universal connected to an **Endurant Universal Expression** via an existential dependence (*inherence*) relation. This **Endurant Universal Expression** is then used to describe the universals whose instances are the bearers of the instance of this mode universal. Since a Mode can be an *externally-dependent mode*, the Mode Pattern also contains a (possibly empty) set of relationships of *external dependence* connecting the instances of the mode universal at hand with their sources of external dependence.

Table 10 – The Mode Pattern

Name (Acronym)	Intent
Mode Pattern (Mode)	Represents **Modes** of **Endurant Universals** (either **Substantial Universals** or even **Moment Universals**).
Rationale	
A **Mode** is an intrinsic moment, i.e. a moment that inheres in one single individual. **Mode Universals** are connected to an **Endurant Universal** via an existential dependence (*inherence*) relation. The **Endurant Universal** is the bearer of the **Modes**. A **Mode** can be an **Externally-Dependent Mode**, i.e. a **Mode** that inheres in an **Endurant** and is *externally dependent on* another **Endurant**.	
Pattern Structure	



Related Patterns	-
Examples