

Ingeniería de ontologías

Diseño de ontologías

Enfoques de reuso, patrones y antipatrones



Contenido de la presentación

Enfoques de reuso

- Top-down: ontologías fundacionales
- Bottom-up
- Patrones
- Antipatrones

Ontologías fundacionales

Describen **categorias y relaciones** en alto nivel, independientes del dominio.

- Objetos físicos vs objetos no físicos (ej. Organización)
- Procesos, estados
- Relaciones útiles en diferentes dominios temáticos

Diferentes ontologías fundacionales con diferentes categorías, algunas se solapan

- DOLCE
- BFO
- RO
- UFO
- SUMO
- UFO
- GFO

Armonización de ontologías fundacionales

Ontologías fundacionales – Categorías, relaciones, atributos

Endurant: entidad totalmente presente en un momento

vs

Perdurant: entidad que ocurre, se desenvuelve en el tiempo

Relaciones: participación, ser parte de, constitución, dependencia, constitución

Atributos (color, peso)

B. DOLCE taxonomy

- Thing
 - Particular
 - Abstract
 - Fact
 - Region
 - AbstractRegion
 - PhysicalRegion
 - SpaceRegion
 - TemporalRegion
 - TimeInterval
 - Set
 - Endurant
 - ArbitrarySum
 - NonPhysicalEndurant
 - NonPhysicalObject
 - MentalObject
 - SocialObject
 - AgentiveSocialObject
 - SocialAgent
 - Society
 - NonAgentiveSocialObject
 - PhysicalEndurant
 - AmountOfMatter
 - Feature
 - DependentPlace
 - RelevantPart
 - PhysicalObject
 - AgentivePhysicalObject
 - NonAgentivePhysicalObject
 - Perdurant
 - Event
 - Accomplishment
 - Achievement
 - Stative
 - Process
 - State
 - Quality
 - AbstractQuality
 - PhysicalQuality
 - SpatialLocation
 - TemporalQuality
 - TemporalLocation

DOLCE

Descriptive Ontology for Linguistic and Cognitive Engineering

Relación *participates*:

Dominio *Endurant*

Rango *Perdurant*

Mapeos entre elementos de diferentes ontologías fundacionales:

DOLCE

GFO

physical_object ----- *Material_object*

UFO – Unified Foundational Ontology

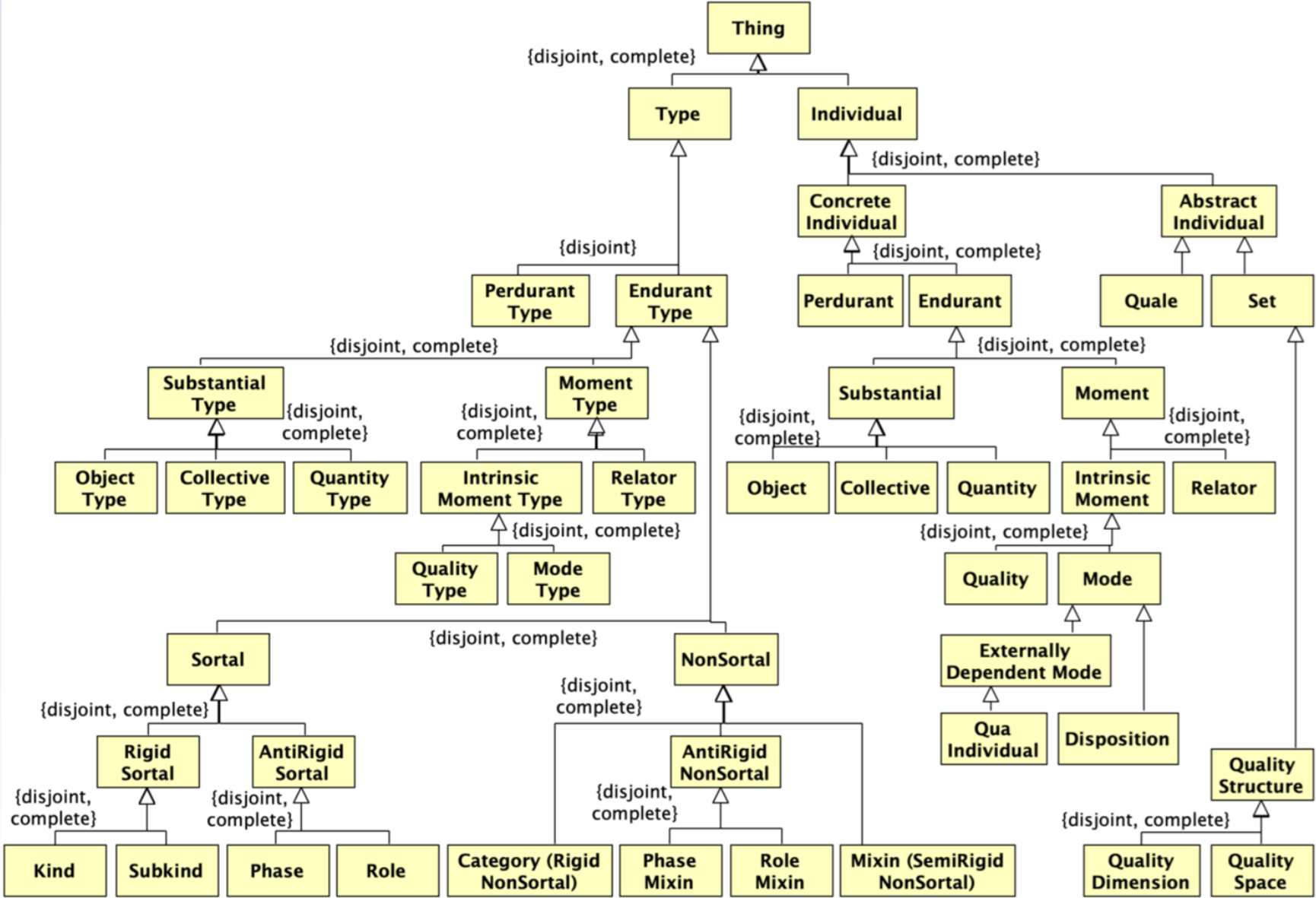
Describe un amplio conjunto de categorías de modelado conceptual

- Tipos de objetos
- Estructuras taxonómicas
- Relaciones parte-todo
- Propiedades
- Eventos

.....

Incluye elementos de otras ontologías fundacionales, ej. DOLCE

UFO – Unified Foundational Ontology



UFO – Unified Foundational Ontology

Tipos:

Sortal (identidad uniforme, ej. Persona, Niño), *Non-Sortal* (ej. Item asegurable)

Rígidos (sus instancias siempre son de ese tipo, ej. Persona), *anti-rígidos* (ej. Niño)

Kind: tipo *Sortal* y *rígido* (ej. *Persona*)

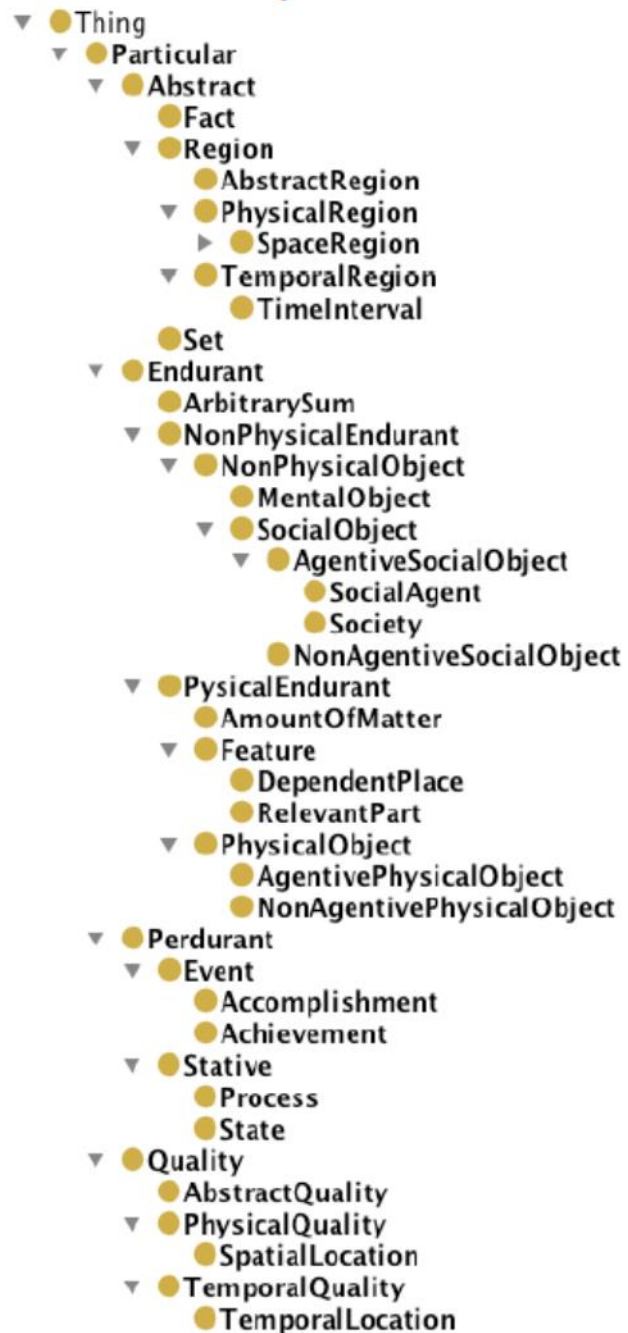
Tipos *Sortal* y *anti-rígidos*:

- *phases* (propiedades intrínsecas, ej. Adulto, Adolescente)
- *roles* (depende de una relación, ej. Estudiante, Esposo)

Relaciones:

- *formales* (propiedades intrínsecas, ej. María es más alta que Juan)
- *materiales* (existe un mediador: “relator”, ej. María está casada con Juan)

B. DOLCE taxonomy



Enfoque top-down - Ontologías fundacionales

Consiste en reusar un conjunto de categorías y relaciones de una ontología fundacional, adaptándolas al dominio que se está conceptualizando.

African Wildlife Ontology

AWO:Plant \sqsubseteq *dolce:PhysicalObject*

AWO:Tusk \sqsubseteq *dolce:PhysicalObject*

AWO:Apatite \sqsubseteq *dolce:Amount-Of-Matter*

Enfoque top-down - Ontologías fundacionales

Ventajas y desventajas

Si se usa una ontología fundacional para modelar una ontología de dominio:

- *No es necesario “reinventar la rueda”* respecto a las categorías básicas y sus relaciones
- *Mejora la calidad general de la ontología*, ya que se aplican decisiones de diseño basadas en principios
- Facilita la *interoperabilidad entre ontologías* que están alineadas a la misma ontología fundacional

Sin embargo:

- Las ontologías fundacionales son demasiado *abstractas, expresivas y exhaustivas* para el desarrollo de un sistema de información
- Toma *demasiado esfuerzo entenderlas* suficientemente para poder usarlas

Ontologías fundacionales – son guías en el diseño

Relación “es parte de” vs subclase

Un departamento es parte de un país, no una subclase

Clase vs instancia

Pink Lady es una subclase de Manzana, no una instancia

La manzana que estoy comiendo es una instancia, no es instanciable

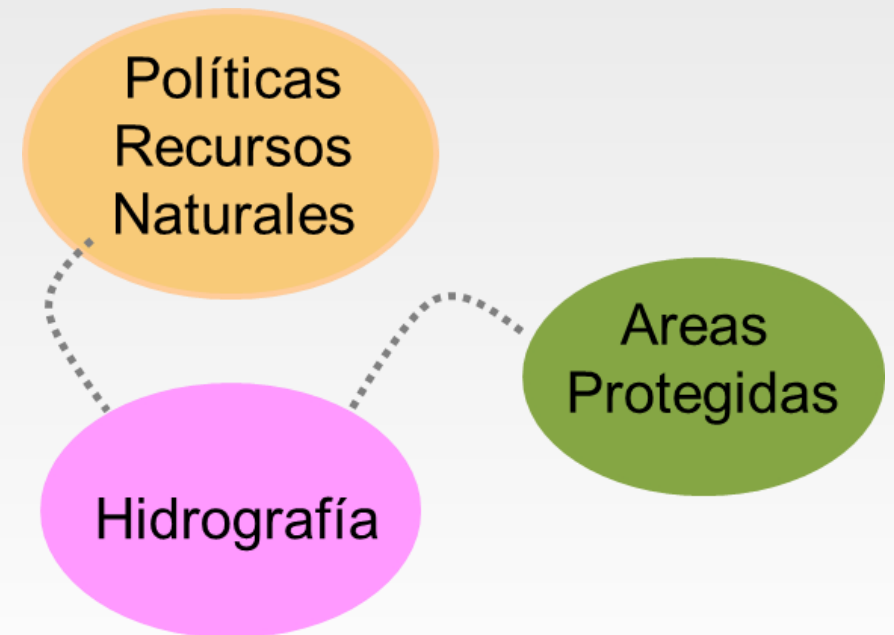
Enfoque bottom-up

Consiste en el reuso recursos ontológicos y no ontológicos (bases de datos, planillas) para construir una ontología

- Reingeniería y/o integración en una red de ontologías
- Varios de los escenarios de la metodología NeOn siguen este enfoque.

Ejemplo: Sistema de Gestión de Políticas de Recursos Naturales

*No es sencillo seleccionar los recursos
(ontológicos y no ontológicos, y fragmentos de ellos)
e integrarlos*



Patrones de diseño

Templates para representar, y posiblemente **resolver, un problema de modelado**

Pueden/deberían ser usados para **describir una “mejor práctica” de modelado**

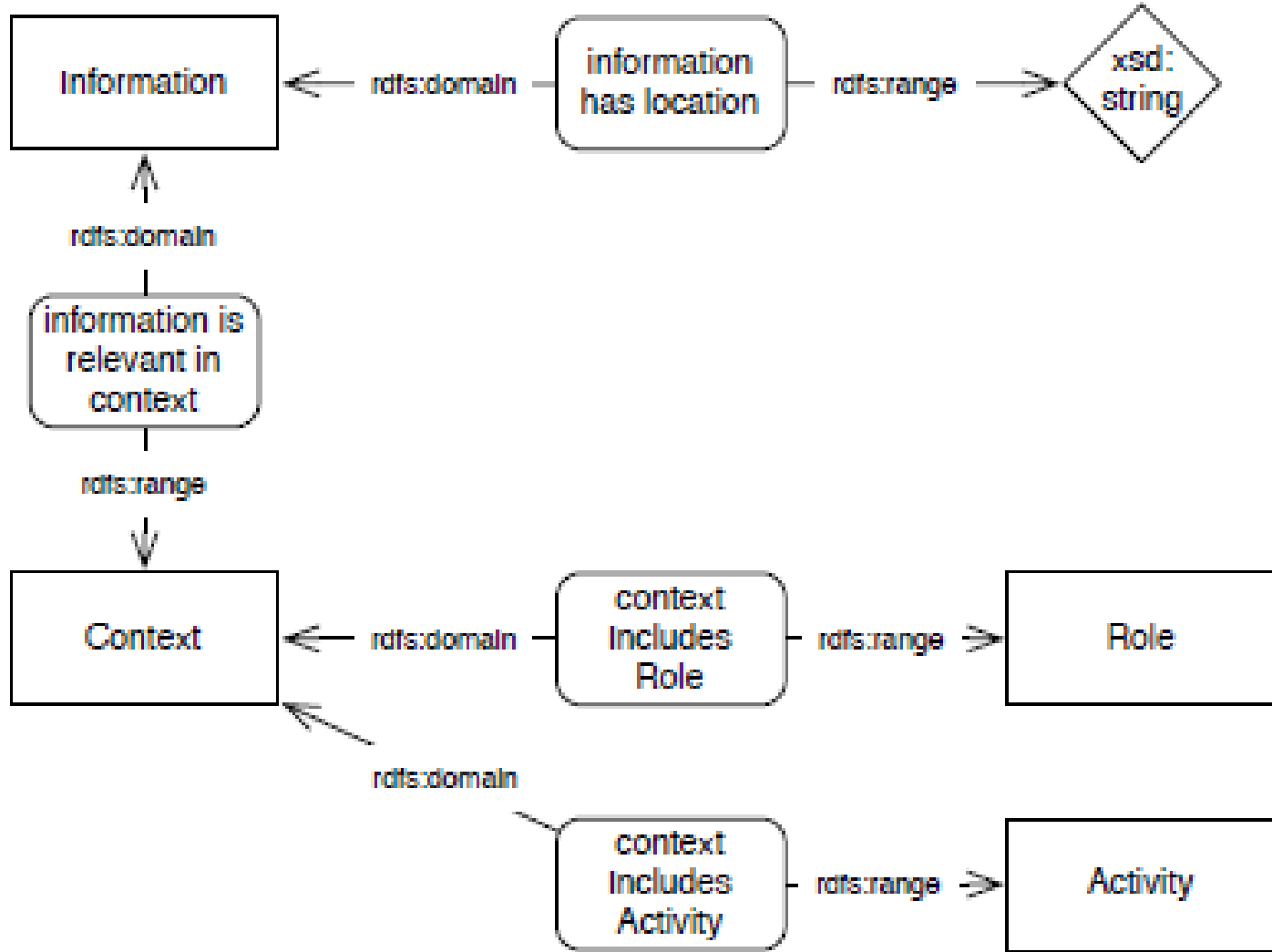
Soluciones de modelado para **resolver problemas recurrentes de Desarrollo de ontologías**

(Gangemi, Presutti)

Se pueden ver como un *enfoque intermedio entre top-down y bottom-up*, podrían ser:

- Fragmentos reusables de una ontología fundacional
- Recursos que se reusan, e.g. buenas prácticas de diseño.

Patrones de diseño - Ejemplo



Context dependant information

Patrones de diseño - Ejemplo

Name	Context dependant information
Intent	To model the case when some information is deemed especially relevant for a particular role performing a particular action.
Competency questions	What information is available that in some way deals with task X? What documents are available that are relevant only for an Astronomer (role) doing task Y? I am a PhD Student (role). What documents are there that I could be interested in, of any topic?
Solution description	One or more roles are assigned to a person. The activities that are performed in the target domain are modelled as Activity instances. Both Role and Activity can be subclassed depending on one's needs. Roles and Activities are joined by context, for instance "Doctor doing diagnosis" or "Medically unskilled person doing diagnostics". What Information instance is deemed relevant for each context is decided by way of the "informationIsRelevantInContext" property.
Reusable OWL building block	http://www.infoeng.se/~karl/images/f/f5/Context_Dependant_Information.owl
Consequences	No known consequences.
Scenarios	Medical doctors using different diagnostics manuals than non-medically trained people when diagnosing illnesses. Car mechanics using different guidelines when servicing exhaust systems than brake pedals.

Patrones de diseño - Ventajas

Promover la aplicación de buenas prácticas

Reusar soluciones probadas e implementadas

Reducir la complejidad de la actividad de diseño de ontologías

Acelerar el proceso de desarrollo de ontologías

Mejorar la calidad y usabilidad de las ontologías

Facilitar la comunicación con expertos de dominio, lenguaje común

Patrones en ingeniería de software

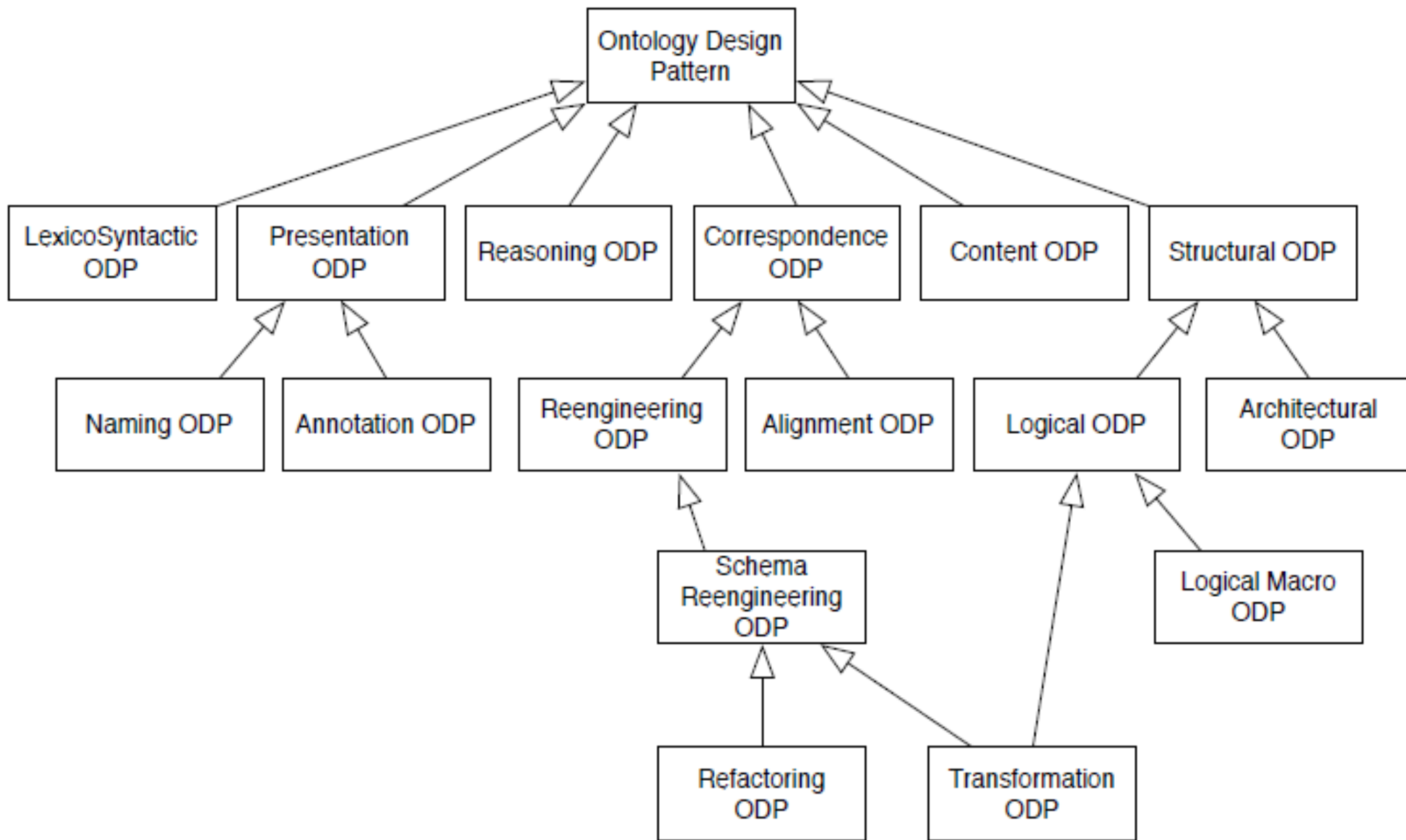
“A pattern describes a particular **recurring design problem** that arises in **specific design contexts** and presents a **well-proven solution** for the problem. The **solution** is specified by **describing** the **roles of its constituent participants, their responsibilities and relationships**, and the ways in which they **collaborate**”.

(Buchmann)

Diferentes tipos de patrones para resolver problemas en las diferentes fases del proceso de Desarrollo

- Patrones de análisis: fragmentos de modelos conceptuales (independientes de tecnologías)
- Patrones de diseño: dependientes de requerimientos no funcionales

Patrones de diseño de ontologías – Clasificación (NeOn)



Patrones de diseño – Clasificación (NeOn)

Content ODP: modelado de contenido de un dominio, pequeños fragmentos de modelos conceptuales

- Patrones de ontologías fundacionales
- Patrones de dominio

Structural ODP: estructura de la ontología, por ejemplo cómo modelar determinadas relaciones y tratar con las limitaciones del lenguaje

Correspondence ODP: reingeniería, refactoring, mapping e interoperabilidad entre ontologías

Reasoning ODP: Normalización y estandarización de ontologías

Presentation ODP: nombramiento, notación, representación gráfica (usabilidad).

Lexico-sintactic ODP: mapping de las estructuras del lenguaje con estructuras de la ontología.

Lenguaje de representación de patrones - OntoUML

Es un lenguaje de modelado conceptual **basado en la ontología fundacional UFO**

- Primitivas de modelado que corresponden a las categorías de UFO
- Restricciones formales que rigen cómo combinar estos constructores
- Estas restricciones son en realidad *patrones de modelado conceptual* generales

UFO – Unified Foundational Ontology

Tipos:

Sortal (identidad uniforme, ej. Persona, Niño), *Non-Sortal* (ej. Item asegurable)

Rígidos (sus instancias siempre son de ese tipo, ej. Persona), *anti-rígidos* (ej. Niño)

Kind: tipo *Sortal* y *rígido* (ej. *Persona*)

Tipos *Sortal* y *anti-rígidos*:

- *phases* (propiedades intrínsecas, ej. Adulto, Adolescente)
- *roles* (depende de una relación, ej. Estudiante, Esposo)

Relaciones:

- *formales* (propiedades intrínsecas, ej. María es más alta que Juan)
- *materiales* (existe un mediador: “relator”, ej. María está casada con Juan)

Lenguaje de representación de patrones - OntoUML

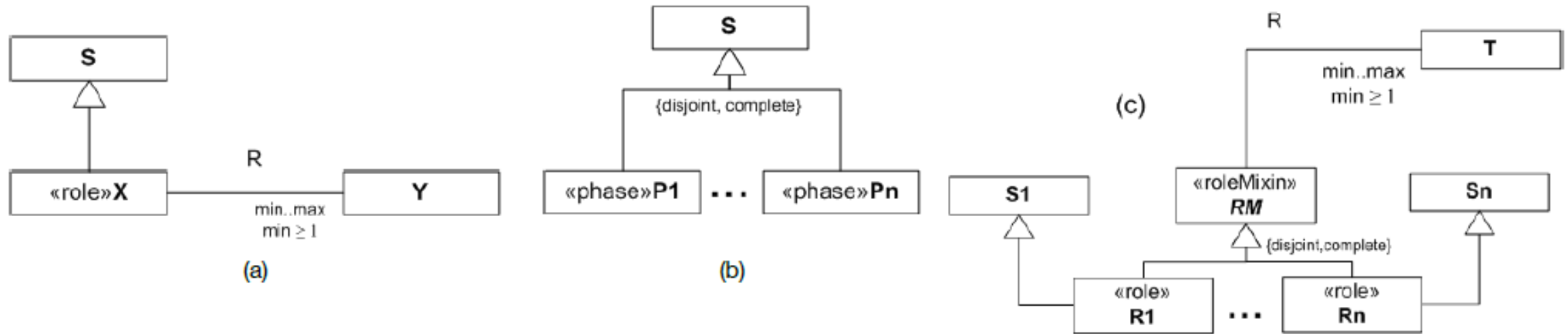
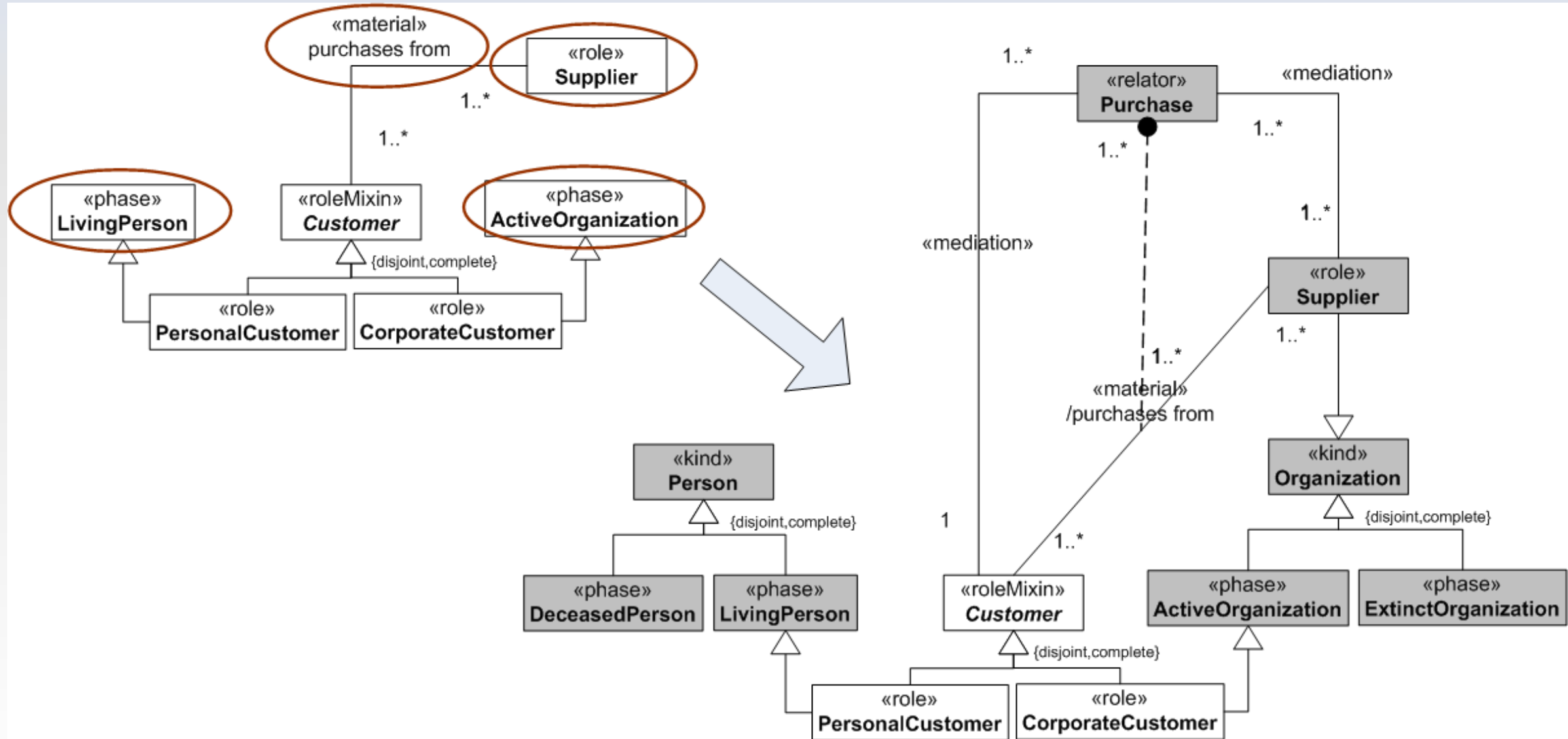


Fig. 2. Role Pattern (a), Phase Partition Pattern (b) and the Role Mixin Pattern (c).

Ejemplo de aplicación de patrones - OntoUML



Formas de reusar patrones

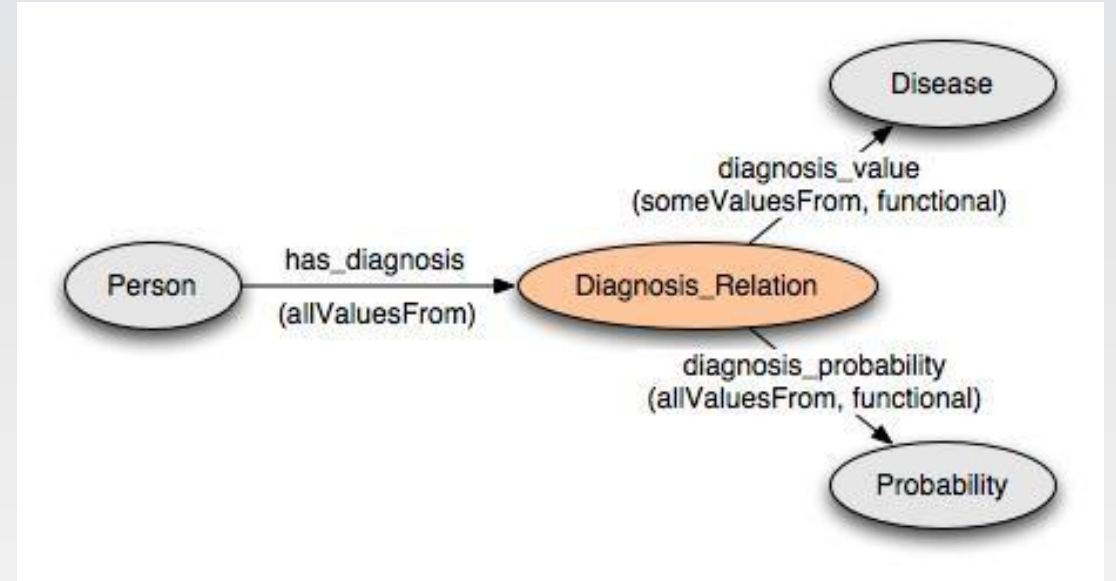
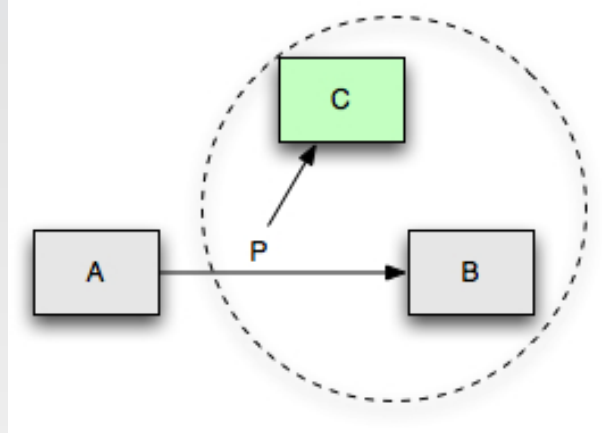
Por analogía

Se identifica qué conceptos del dominio a modelar corresponden a los conceptos del patrón y se reproduce la estructura del patrón en la ontología que se está desarrollando

Por extensión

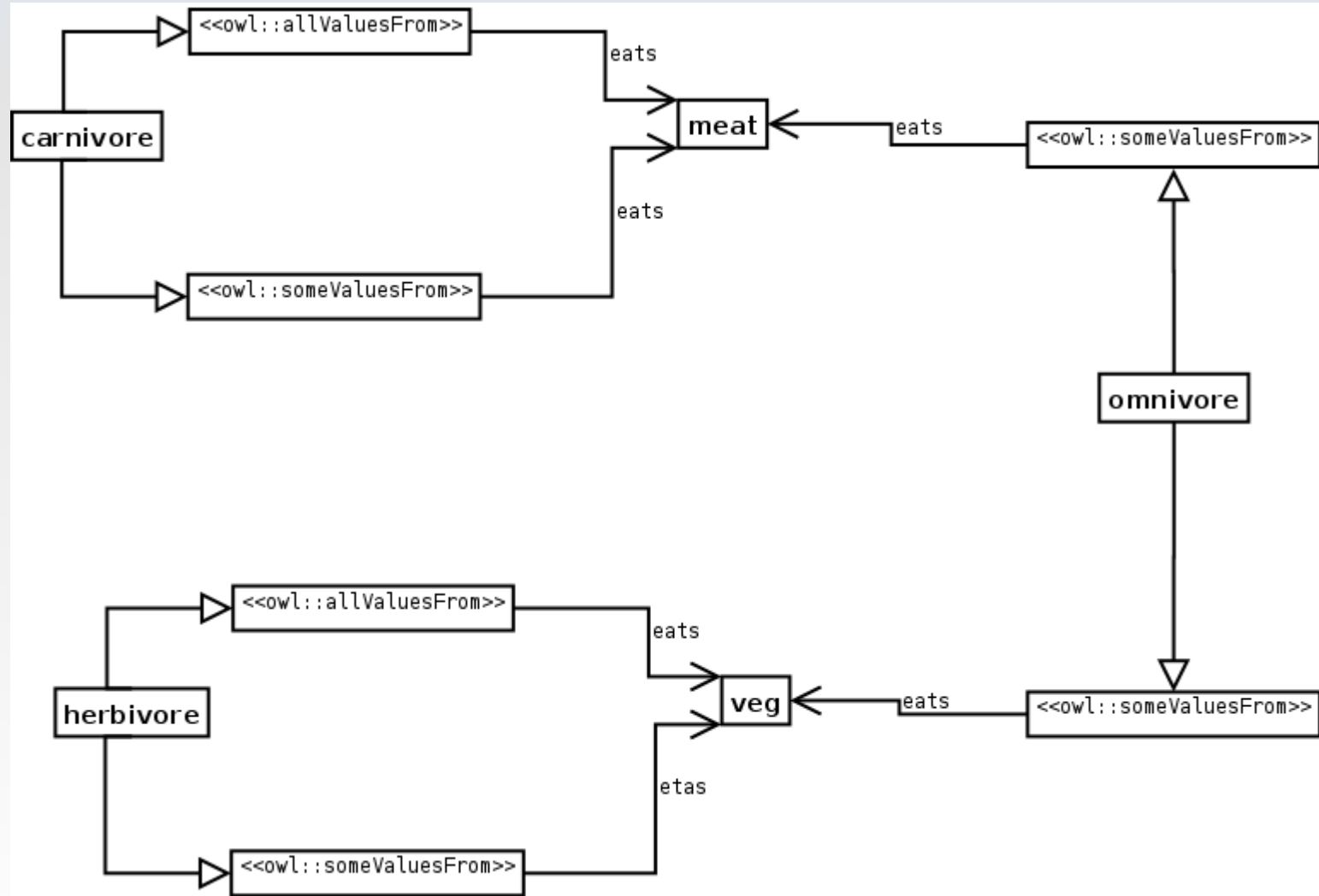
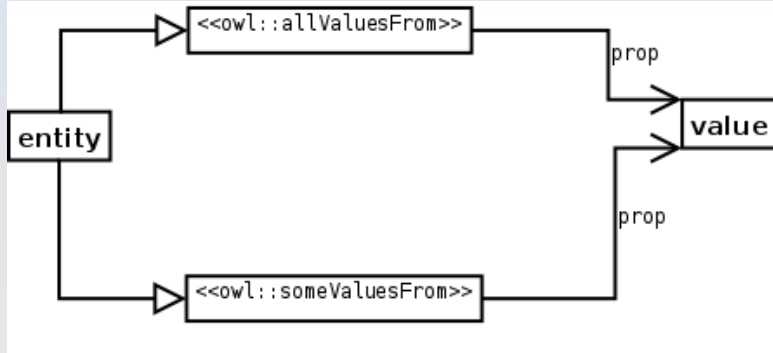
Se incorpora la estructura del patrón en la ontología que se está desarrollando y se extiende con nuevos conceptos y relaciones

Patrones – Relaciones n-arias

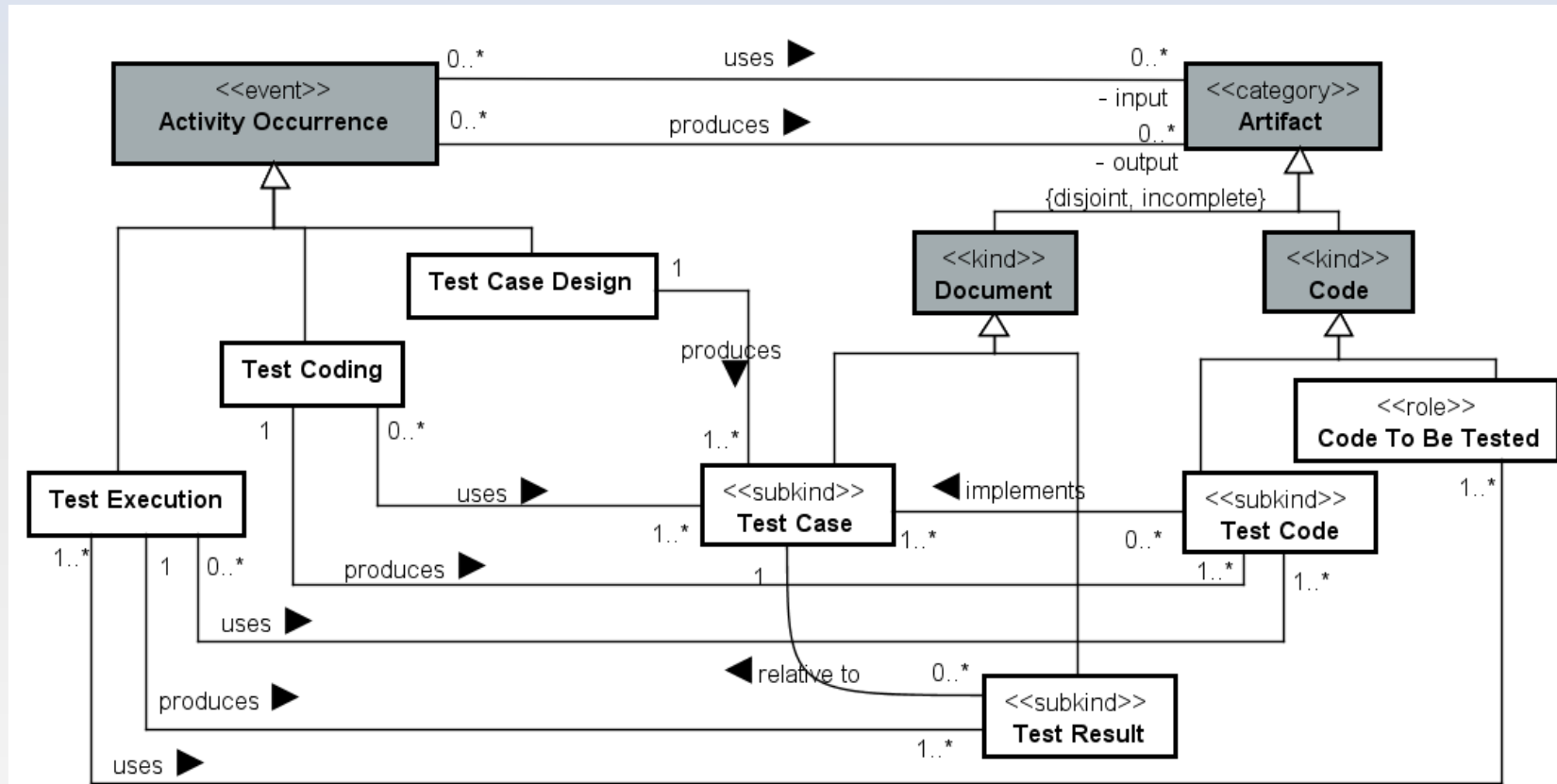


<https://www.w3.org/TR/swbp-n-aryRelations/>

Patrones - Clausura



Patrones de dominio



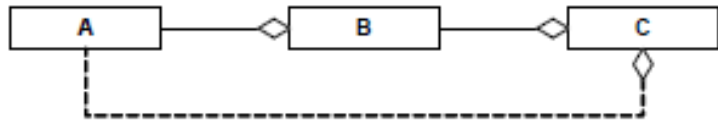
Se extiende el patrón Work Product Participation (WPPA) pattern

Antipatrones

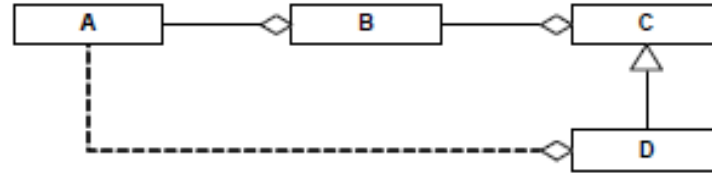
Un antipatrón es una estructura de modelado de un problema tal que su interpretación no corresponde a la realidad que se pretende conceptualizar.

(Prince Sales, Guizzardi)

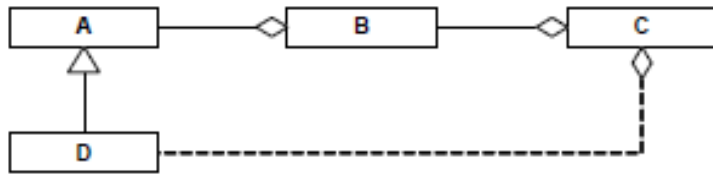
Antipatrones – Transitividad de relaciones parte-todo



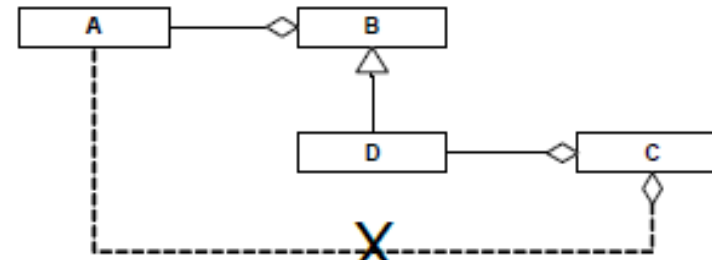
(a)



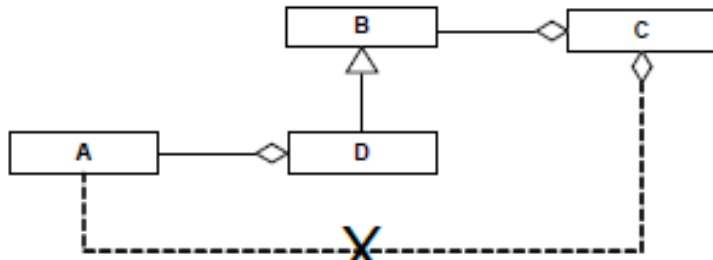
(b)



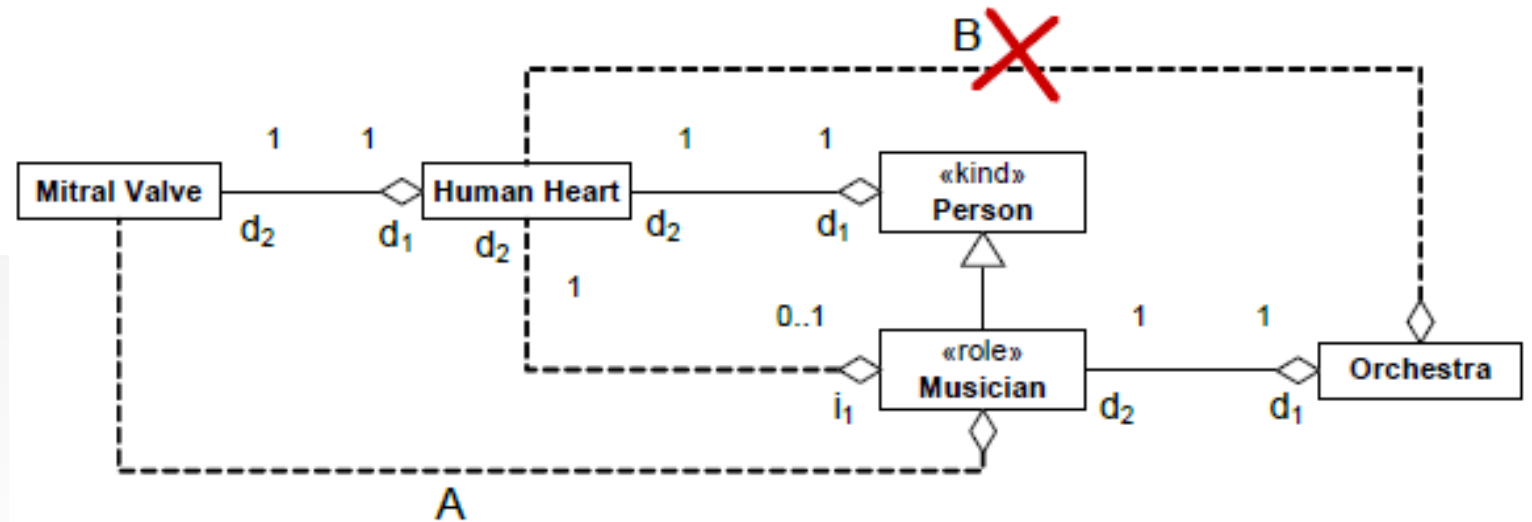
(c)



(d)



(e)



Antipatrones

$$C1 \sqsubseteq \forall R.C2 \quad C1 \sqsubseteq \forall R.C3 \quad C2 \sqcap C3 \sqsubseteq \perp$$

$$C1 \sqsubseteq \exists R.(C2 \sqcap C3) \quad C2 \sqcap C3 \sqsubseteq \perp$$

$$C1 \sqsubseteq \forall R.C2 \quad C1 \sqsubseteq \exists R.C3 \quad C2 \sqcap C3 \sqsubseteq \perp$$

$C1 \equiv C2$ sinónimos

$$C1 \sqsubseteq \exists R.C2 \quad C1 \sqsubseteq \geq 2R.C2$$

$$C1 \sqsubseteq \forall R.C2 \quad C1 \sqsubseteq \geq 2R.T \rightarrow \text{Agrupar restricciones: } C1 \sqsubseteq \forall R.C2 \sqcap \geq 2R.T$$

$$C1 \sqsubseteq \geq 0R.T$$

Bibliografía

An Introduction to Ontology Engineering. C. Maria Keet. 2020. Capítulo 5

Content Ontology Design Patterns: Qualities, Methods, and Tools. Karl Hammar. 2017 Capítulos 2 y 4

An overview of ontoclean. N. Guarino, C. Welty. Handbook on Ontologies, 2009.

Ontowlclean: cleaning OWL ontologies with OWL. Chris Welty. 2006.

Ontology Patterns: Clarifying Concepts and Terminology. Ricardo A. Falbo, Giancarlo Guizzardi, Aldo Gangemi, Valentina Presutti. 2013.

Valentina Presutti, Aldo Gangemi, Stefano David, Guadalupe Aguado de Cea, Mari Carmen Suárez-Figueroa, Elena Montiel-Ponsoda, María Poveda. D2.5.1: A Library of Ontology Design Patterns: reusable solutions for collaborative design of networked ontologies, 2005.

Ontological Patterns, Anti-Patterns and Pattern Languages for Next-Generation Conceptual Modeling. Giancarlo Guizzardi. 2014.

Catálogos de patrones: <http://ontologydesignpatterns.org/>, <https://www.w3.org/2001/sw/BestPractices/>,
<http://www.gong.manchester.ac.uk/odp/html>

Anti-patterns in Ontology-driven Conceptual Modeling: The Case of Role Modeling in OntoUML. Tiago Prince Sales, Giancarlo Guizzardi. 2016.

Ontological Anti-Patterns in Taxonomic Structures. Tiago Prince Sales¹ and Giancarlo Guizzardi. 2019.