Ontology engineering

Concetti generali

Ontology engineering

- Criteri e linee guida per progettare ontologie interoperabili e ben fondate sul piano concettuale
- Design patterns: archivio di pattern riutilizzabili per il design di ontologie
- Metodologie: es. OntoClean, Neon
- Approcci collaborativi allo sviluppo di ontologie

OntoClean

- «It is based on highly general ontological notions drawn from philosophy, like essence, identity, and unity, which are used to characterize relevant aspects of the intended meaning of the properties, classes, and relations that make up an ontology.»

- «These aspects are represented by formal metaproperties, which impose several constraints on the taxonomic structure of an ontology.»

- «The analysis of these constraints helps in evaluating and validating the choices made.»

Guarino, N., & Welty, C. A. (2009). An overview of OntoClean. In Handbook on ontologies (pp. 201-220). Springer Berlin Heidelberg.

OntoClean

- Sorta di “critica” dell’ontologia orientata alla tassonomia delle classi
- Basato su 4 (meta)proprietà delle classi:
  - identity
  - unity
  - rigidity
  - dependence
- *Classe* intesa come “appartenenza alla classe” (proprietà che definisce l’appartenenza alla classe)
- Non è prescrittiva rispetto alle entità del mondo reale, ma fornisce proprietà con cui caratterizzare le classi

Meta-proprietà e vincoli

Dall'analisi delle meta-proprietà delle classi emergono alcuni vincoli sulla sussunzione tra classi

Per esempio, una classe anti-rigida, come un ruolo, non può sussumere una classe rigida

Ne consegue che la classe 'studente' non può sussumere la classe 'persona'

Questi vincoli possono essere utilizzati per verificare e ristrutturare la gerarchia delle classi

Si annotano le classi con meta proprietà

La classe possiede la proprietà: +P,

La classe non possiede la proprietà -P

anti-proprietà ~P

Meta-proprietà in Ontoclean

- Identità (I): proprietà *intrinseca* che identifica un tipo di oggetti («identity refers to the problem of being able to recognize individual entities in the world»).
  - Esempio: triangolo → lunghezza dei lati
  - *Sortal* = classe di oggetti che possono essere identificati nello stesso modo.
- Unità (U): proprietà di un tipo di oggetto di essere unitario («unity refers to the problem of describing the parts and boundaries of an object»)
  - Esempio: persona vs argilla
  - *Whole* = classe di oggetti che sono tali solo in quanto costituiti da parti collegate tra loro da relazioni strutturali
- Rigidità (R): proprietà di un tipo di oggetto che non è soggetta a cambiamenti
  - Esempio: persona vs studente (anti-rigido)
- Dipendenza (D): proprietà di un tipo di oggetti di dipendere da un altro per la propria definizione
  - Es. studente – scuola

Ruoli e *phased sortals*

- “Phased sortal is a property whose instances are allowed to change certain of their identity criteria during their existence, while remaining the same entity. The canonical example is a caterpillar.”

- Entità che rimangono le stesse pur modificando in parte i propri criteri di identità
  - Per esempio, il bruco (diventa farfalla)
- Sono indipendenti (-D), anti-rigidi (~R) e hanno un criterio di identità (+I)

- Inoltre, O rappresenta la proprietà di avere un criterio di identità proprio, non ereditato (O sta per “own”)

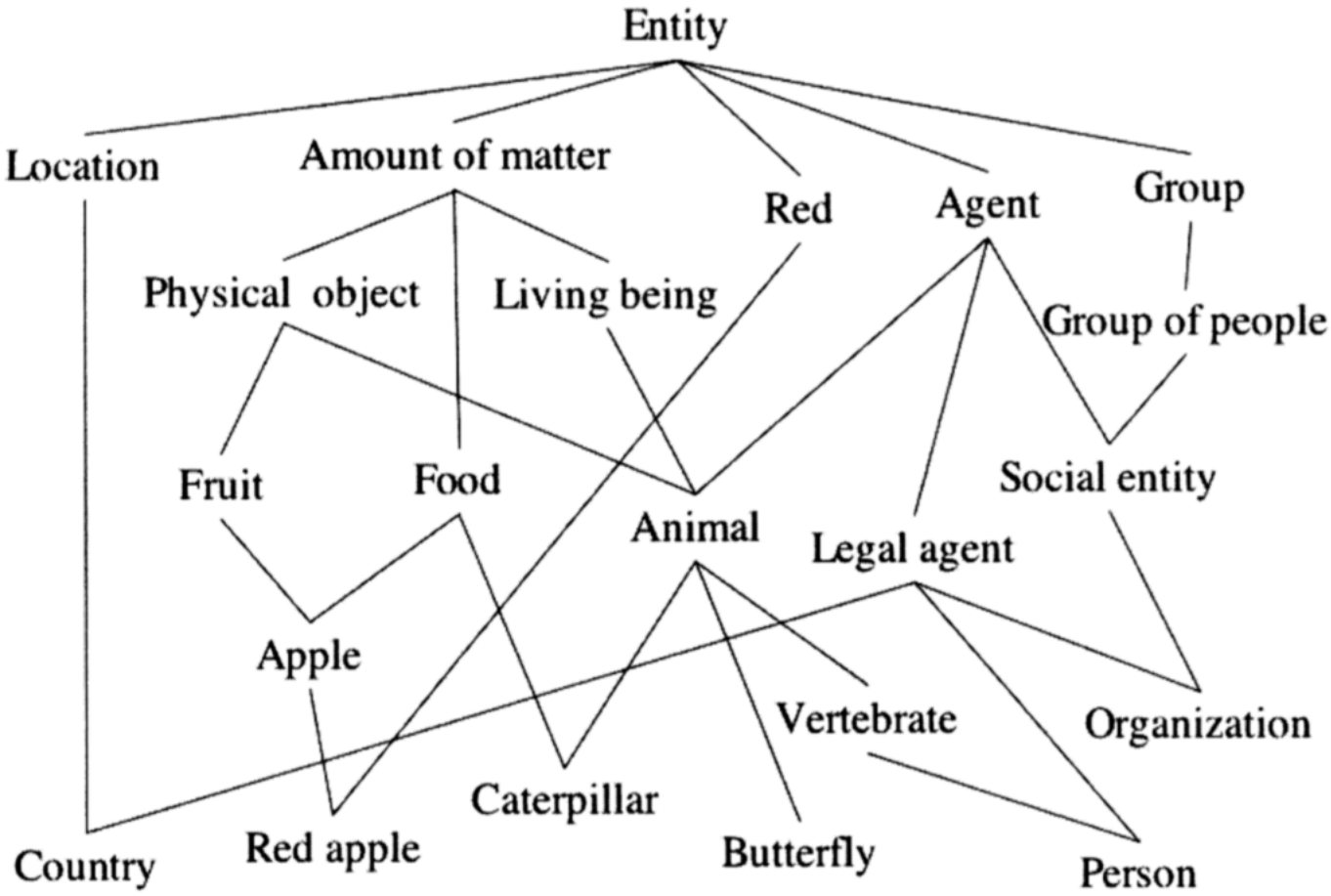
Meta proprietà e sussunzione tra classi

1. Identità (I): proprietà *intrinseca* che identifica un tipo di oggetti.
   - Viene ereditata dalle sottoclassi

1. Unità (U): proprietà di un tipo di oggetto di essere unitario.
   - Viene ereditata dalle sottoclassi (non vale per –U)

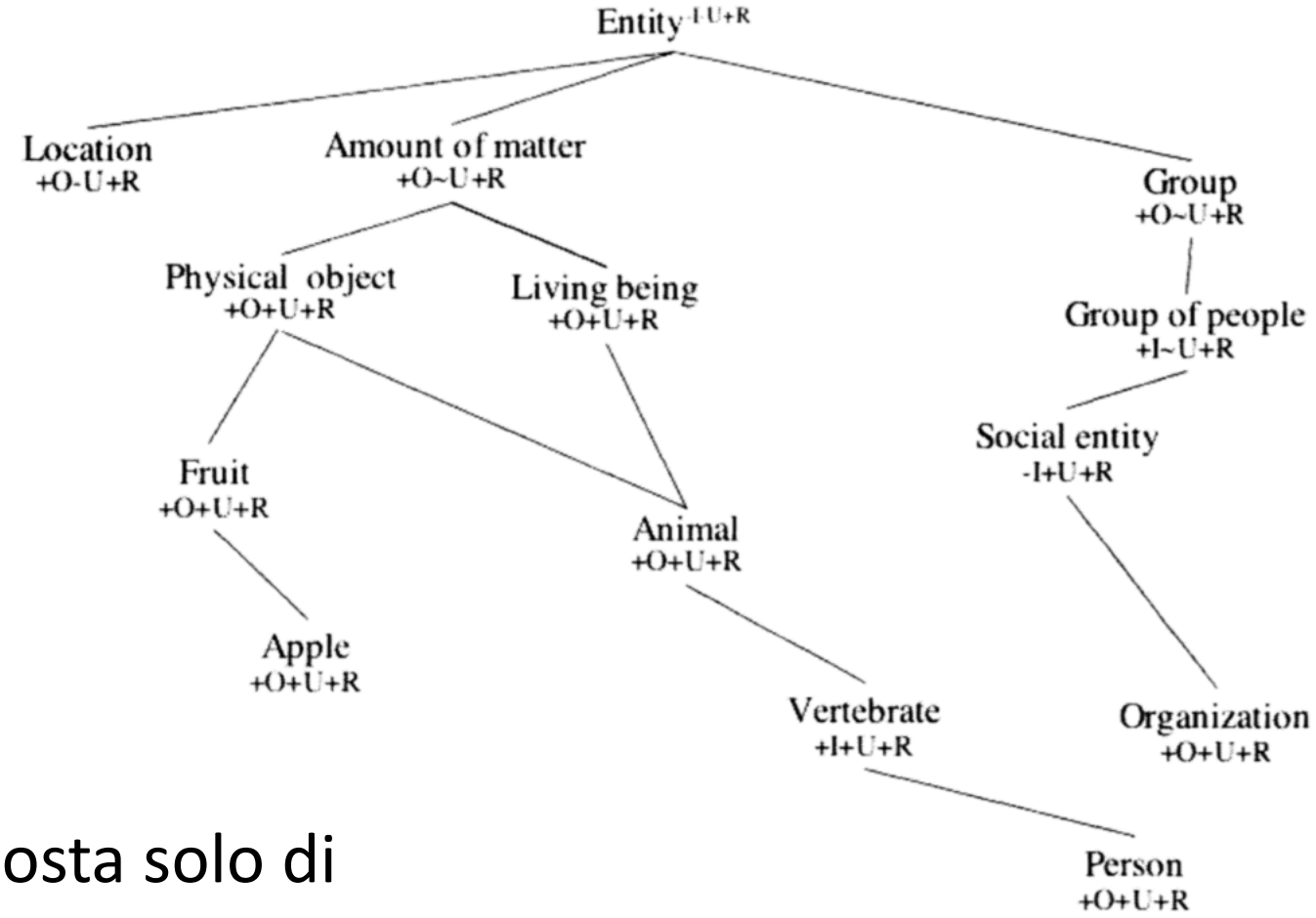
1. Rigidità (R): proprietà di un tipo di oggetto che non è soggetta a cambiamenti
   - Solo l'antirigidità (~R) viene ereditata

Esempio: tassonomia iniziale



Da: Guarino, N., & Welty, C. A. (2009). An overview of OntoClean. In Handbook on ontologies (pp. 201-220). Springer Berlin Heidelberg.

Esempio: “backbone taxonomy”



Composta solo di proprietà rigide

Esempio 1

- Errori rilevati!
- La classe Amout of matter sussume la classe Physical object
  - Physical Object è un tipo specifico di Amount of Matter
  - Ogni istanza di Physical Object è anche un'istanza di Amount of Matter, il che non è corretto per la regola 2:

- «[…] a ~U property can't subsume one with +U. This is yet another example of constitution being confused with subsumption.»
- «Physical objects are not themselves amounts of matter, they are constituted of matter. The solution is to make Physical Object subsumed directly by Entity.»

Esempio 2

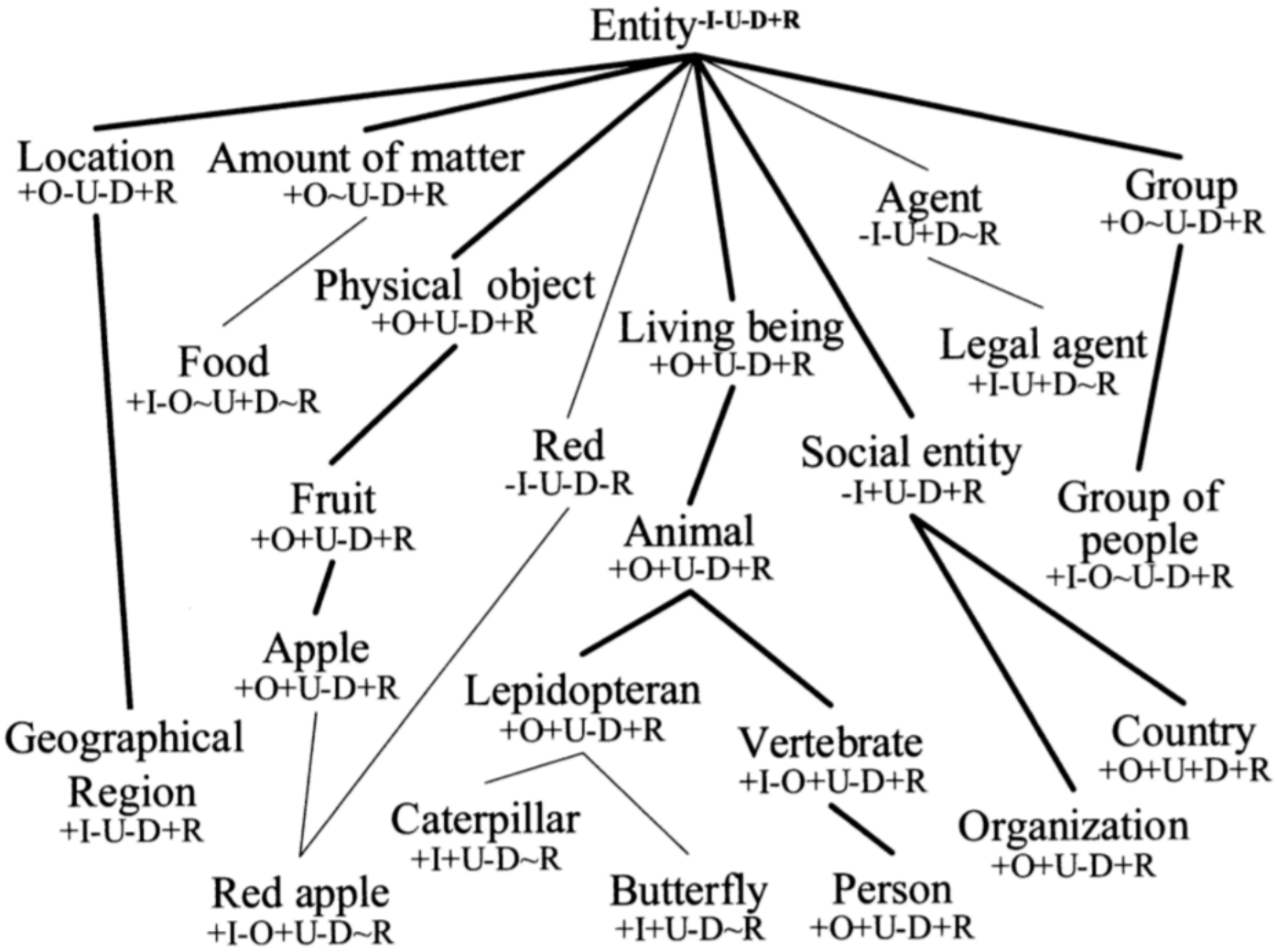
- Il processo di revisione può determinare la creazione di nuove classi
  - Per esempio, la classe Geografical Region, intesa come territorio occupato da una nazione e distinto dalla nazione come entità storico politica (Country).
  - Country *occupa* Geografical Region

"We assign +O+U+R to Country, and +I-U+R to Geographical Region.

The intuition is that countries have their own identity criteria, while geographical regions inherit the identity of locations.

Countries clearly have unity, while this is not the case for arbitrary geographical regions."

Esempio: gerarchia rivista



Neon Methodology

- Metodologia orientata agli aspetti collaborativi nello sviluppo e nel mantenimento di network di ontologie.

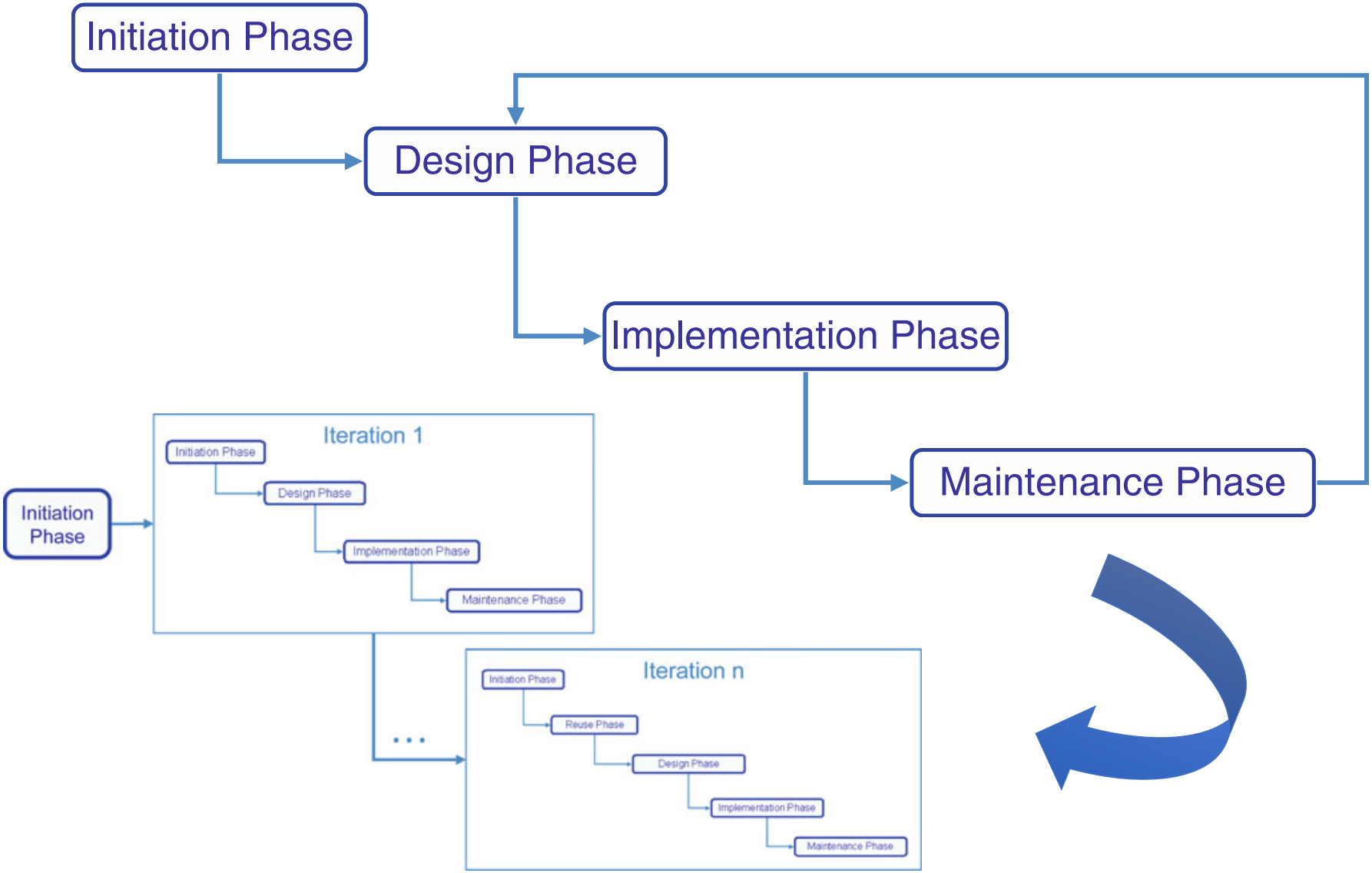
- Si articola in un insieme di 9 scenari, associati a specifiche attività e documenti
- Per esempio:
  - Lo Scenario 1 “From Specification to Implementation” prevede che si produca un documento di *requisiti* per l’ontologia (Ontology Requirements Specification Document, ORSD).
  - Lo Scenario 3 “Reusing Ontological Resources” comporta il fatto di *cercare (search)* le risorse ontologiche, *valutarle (assess)*, confrontarle (*compare*) e selezionarne (*select*) un sottoinsieme.

Suárez-Figueroa, M. C., Gómez-Pérez, A., Motta, E., & Gangemi, A. (Eds.). (2012). Ontology engineering in a networked world. Springer Science & Business Media.

NEON: scenari

- Scenario 1: From specification to implementation
- Scenario 2: Reusing and re-engineering non-ontological resources
- Scenario 3: Reusing ontological resources
- Scenario 4: Reusing and re-engineering ontological resources
- Scenario 5: Reusing and merging ontological resources
- Scenario 6: Reusing, merging, and re-engineering ontological resources
- Scenario 7: Reusing ontology design patterns (ODPs)
- Scenario 8: Restructuring ontological resources
- Scenario 9: Localizing ontological resources

NEON: ontology lifecycle

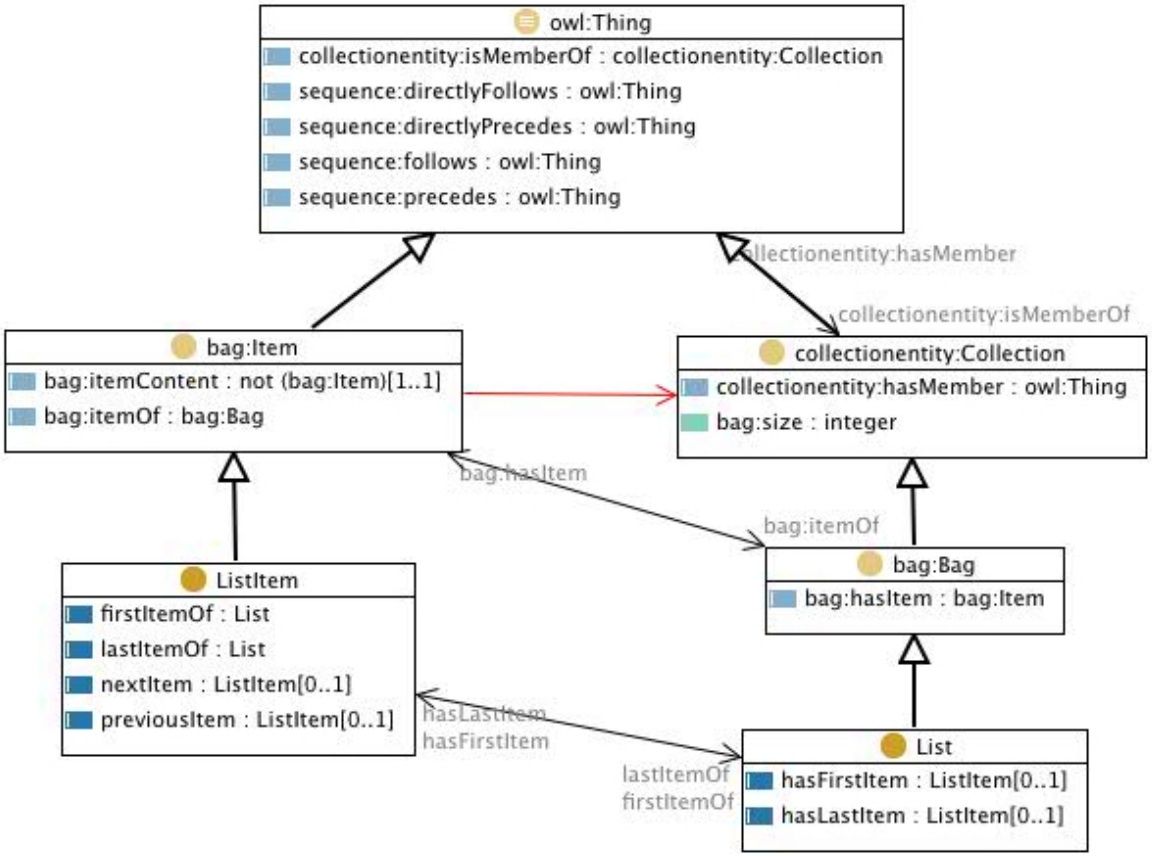


Ontology design patterns (ODP)

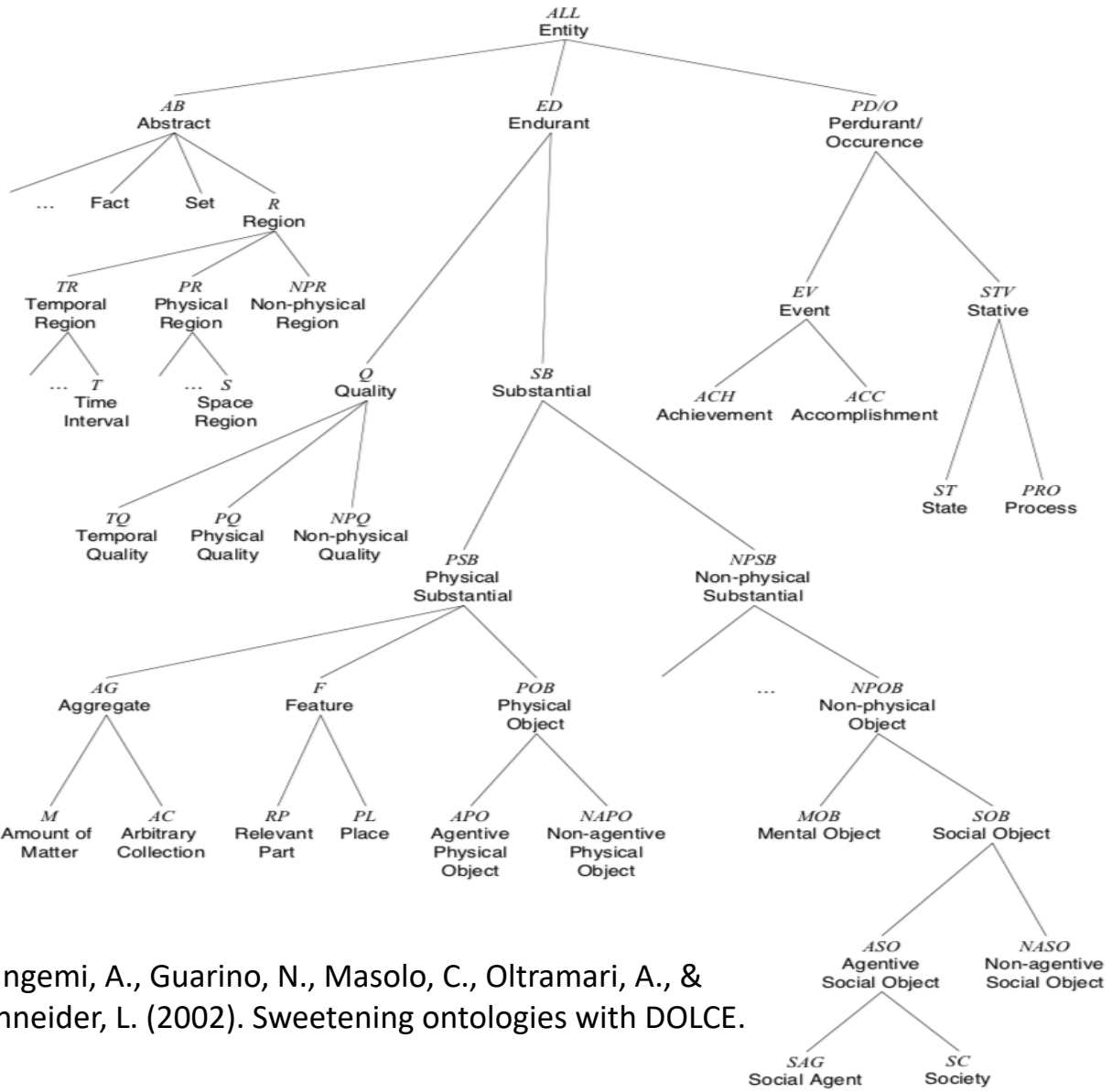
- Sorta di mattoncini per la creazione di ontologie secondo schemi (pattern) convidisi
- Sono divisi in categorie:
  - Structural ODPs
    - Logical ODPs
    - Architectural ODPs
  - Correspondence ODPs
    - Re-engineering ODPs
    - Alignment ODPs
  - Content ODPs (CPs)
  - Reasoning ODPs
  - Presentation ODPs
    - Naming ODPs
    - Annotation ODPs
  - Lexico-Syntactic ODPs

http://ontologydesignpatterns.org/

List (e Bag)


owl:Thing
collectionentity:isMemberOf : collectionentity:Collection
sequence:directlyFollows : owl:Thing
sequence:directlyPrecedes : owl:Thing
sequence:follows : owl:Thing
sequence:precedes : owl:Thing
collectionentity:hasMember
collectionentity:isMemberOf
bag:Item
bag:itemContent : not (bag:Item)[1..1]
bag:itemOf : bag:Bag
collectionentity:Collection
collectionentity:hasMember : owl:Thing
bag:size : integer
bag:hasItem
bag:itemOf
ListItem
firstItemOf : List
lastItemOf : List
nextItem : ListItem[0..1]
previousItem : ListItem[0..1]
bag:Bag
bag:hasItem : bag:Item
hasLastItem
hasFirstItem
lastItemOf
firstItemOf
List
hasFirstItem : ListItem[0..1]
hasLastItem : ListItem[0..1]

Dolce



Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., & Schneider, L. (2002). Sweetening ontologies with DOLCE.

Ontologia orientata alla cognizione e al linguaggio

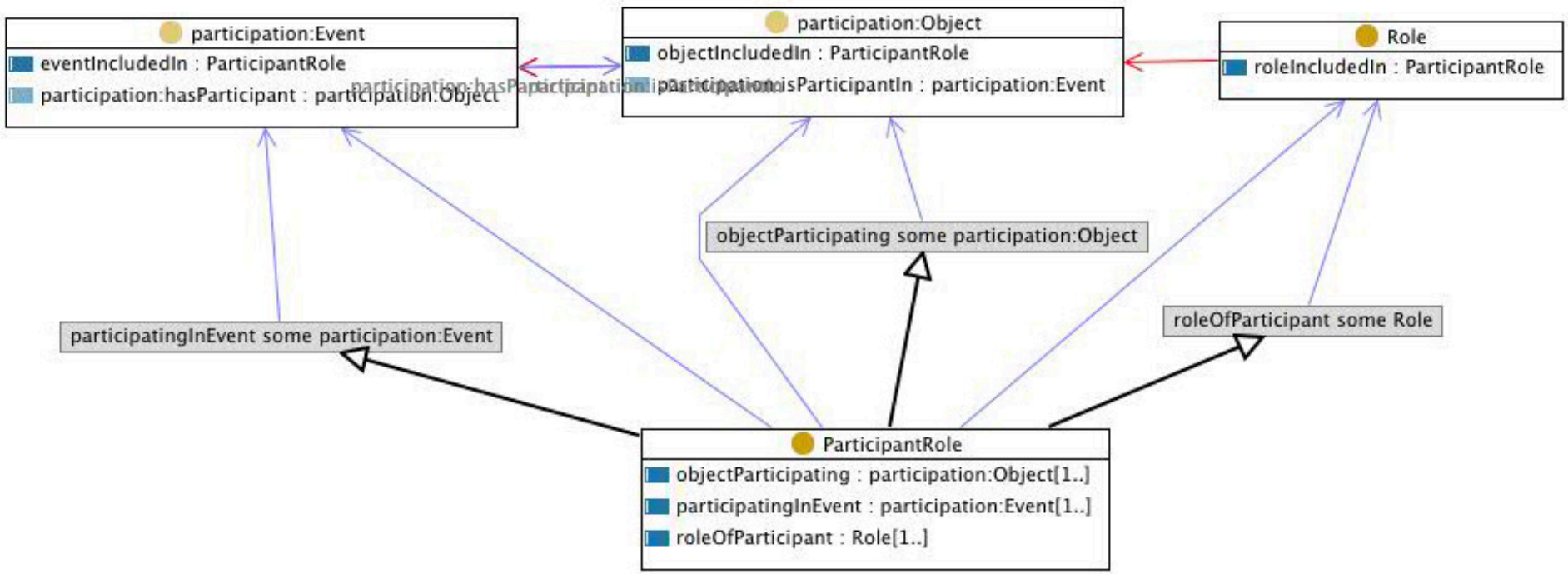
Distinzione chiave tra perduranti e enduranti
Basata sul loro comportamento rispetto al tempo

“Endurants are wholly present at any time they are present. Perdurants just extend in time by accumulating different temporal parts, so that, at any time they are present, they are only partially present.”

Enduranti e perduranti

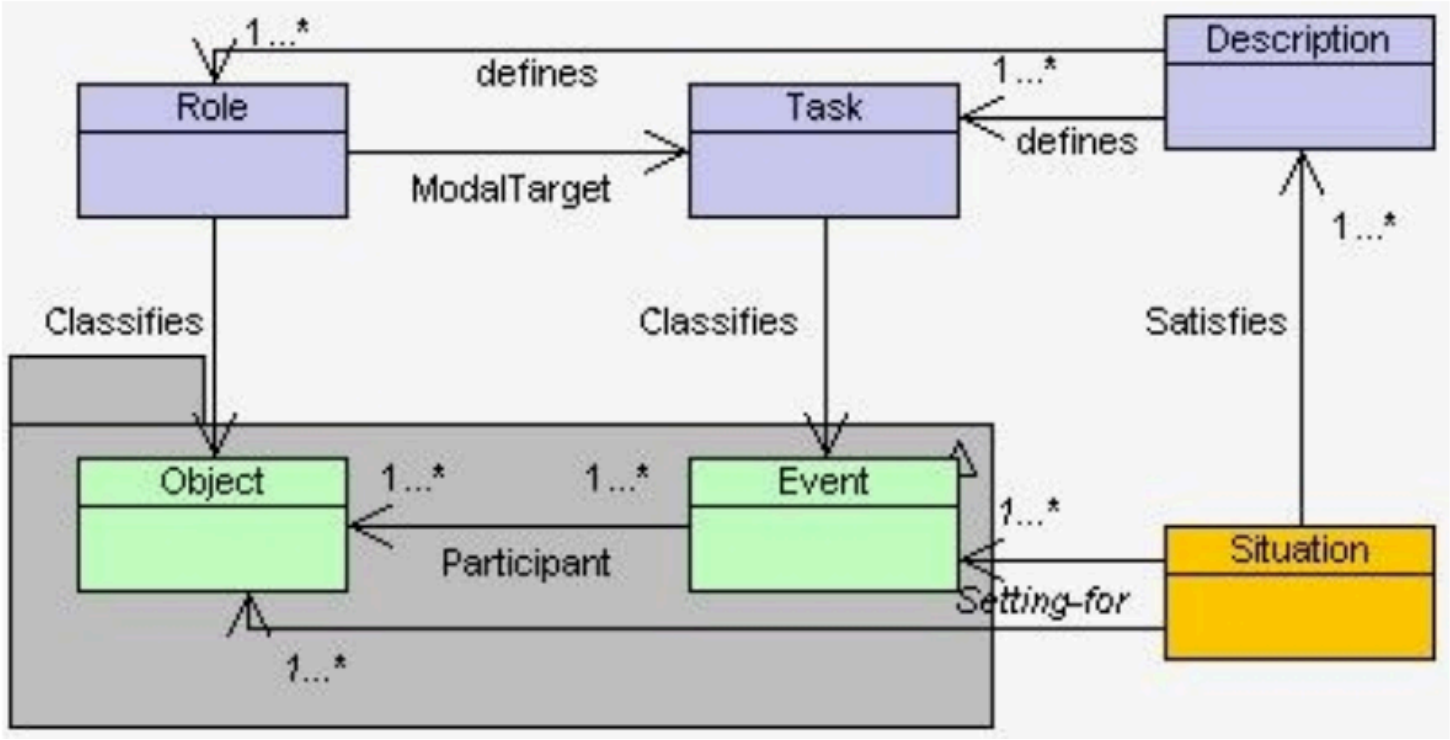
- La differenza 'ontologica' tra evento e partecipanti è catturata dalla differenza tra enduranti (*endurants*) e perduranti (*perdurants*) codificata in Dolce
  - Evento → Perdurant (ha natura temporale)
  - Partecipante → Endurant (non ha natura temporale)

Esempio: Participant Role ODP (Content Design Pattern)



ParticipantRole collega un Evento, un Oggetto che partecipa all'evento (per esempio un agente) con un Ruolo con cui avviene la partecipazione

ODP: Role and Task



Formulazioni *simili* e *in parte sovrapponibili* degli stessi pattern o di pattern collegati (Task)

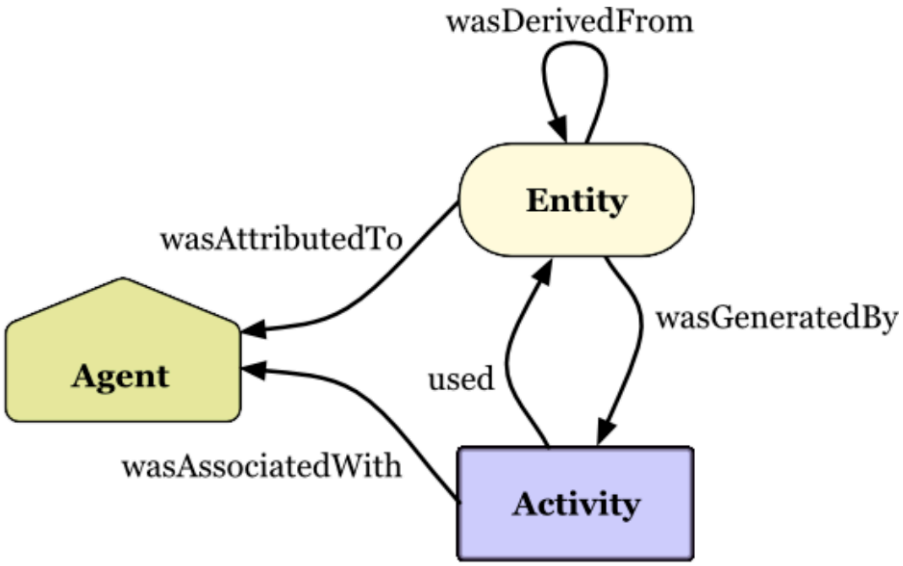
Gangemi, A. (2005, November). Ontology design patterns for semantic web content. In International semantic web conference (pp. 262-276). Springer, Berlin, Heidelberg.

Prov
The Prov-enance ontology

- Concepita per l'interscambio sul Web di informazioni riguardanti la provenienza delle entità

- Descrive l'origine delle entità intesa soprattutto come processi che hanno determinato la creazione di quelle entità
  - La provenienza degli oggetti è rilevante per determinarne il possesso (inteso come diritti) e l'affidabilità.

- I metadati propriamente detti delle entità e la descrizione degli agenti sono affidati a altre ontologie:
  - FOAF (agenti)
  - Dublin Core (entità)

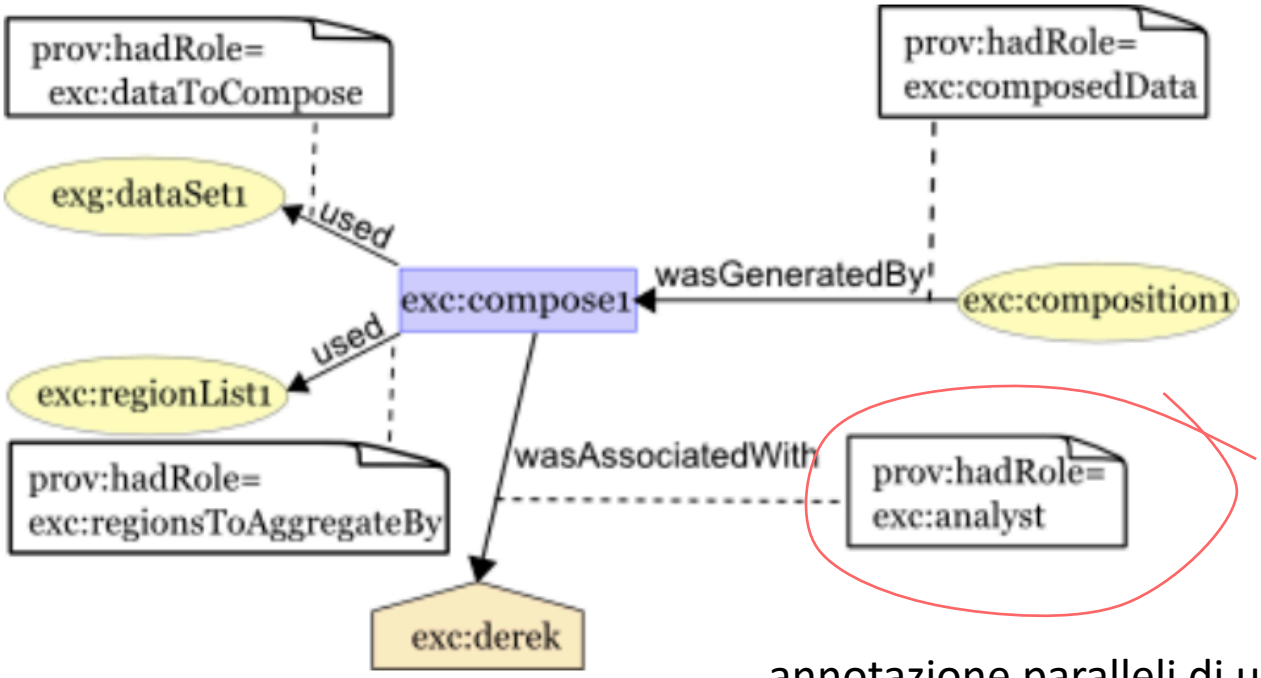


Prov: Overview

- An agent takes a role in an activity such that the agent can be assigned some degree of responsibility for the activity taking place.
  - An agent can be a person, a piece of software, an inanimate object, an organization, or other entities that may be ascribed responsibility.
- Activities are how entities come into existence and how their attributes change to become new entities, often making use of previously existing entities to achieve this.
- A role is a description of the function or the part that an entity played in an activity. Roles specify the relationship between an entity and an activity, i.e. how the activity used or generated the entity
  - Roles are application specific, so PROV does not define any particular roles

Da https://www.w3.org/TR/prov-primer/

Attività e ruoli in Prov



annotazione paralleli di una relazione generica con dettagli sul tipo di ruolo
(qualified association)

```
exc:compose1 prov:qualifiedAssociation [
        a  prov:Association ;
        prov:agent    exc:derek ;
        prov:hadRole  exc:analyst
 ] .
```

Problematiche

Come rappresentare i ruoli?

- Classi dell'ontologia (sottoclassi di Ruolo)
- Classe generica con etichette linguistiche

Classi dell'ontologia

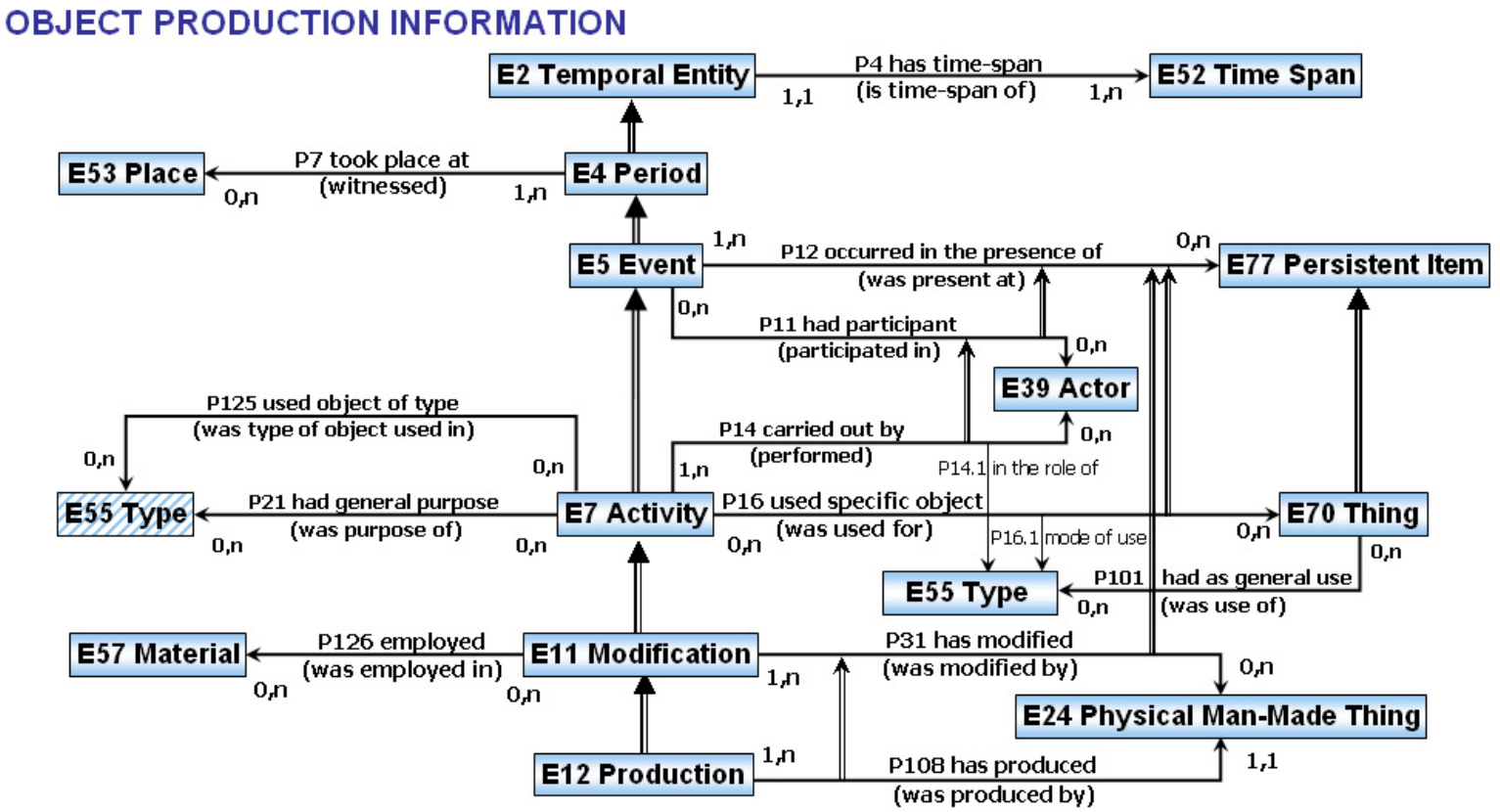
- Vantaggi: ragionamento
- Svantaggi: non aperto, non modulare

Classe generica

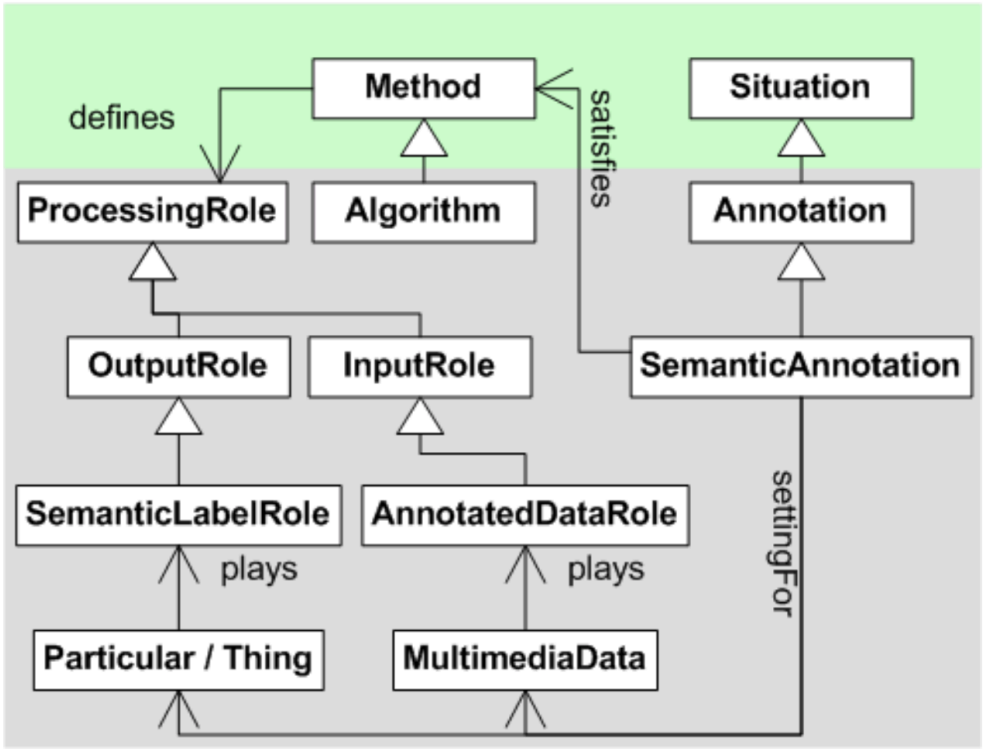
- Vantaggi: modulare
- Svantaggi: più difficile condurre inferenze

Rappresentazione di azione e eventi in CIDOC-crm (Production process)

Linee doppie = relazioni di sussunzione



COMM - Ontology of Multimedia



Semantic Annotation Pattern

Accuratezza vs. semplicità

Verso i Linked Data

- L'ontologia LODE (Linking Open Descriptions of Events) rappresenta gli eventi (per esempio eventi storici) con un vocabolario molto semplice
- Non rappresenta i ruoli ma solo i partecipanti e la collocazione nel tempo e nello spazio degli eventi
- Orientata alla costruzione di grandi data set (descrizione di eventi)
- Per ogni classe, corrispondenza con DOLCE

- http://linkedevents.org/ontology/

Vocabolario LODE: classi e proprietà

Term Name	Type	Definition
Event	*class*	"Something that happened," as might be reported in a news article or explained by a historian.
atPlace	*property*	a named or relatively specified place that is where an event happened.
atTime	*property*	an abstract instant or interval of time that is when an event happened.
circa	*property*	an interval of time that can be precisely described using calendar dates and clock times.
illustrate	*property*	an event illustrated by some thing (typically a media object)
inSpace	*property*	an abstract region of space (e.g. a geospatial point or region) that is where an event happened.
involved	*property*	a (physical, social, or mental) object involved in an event.
involvedAgent	*property*	an agent involved in an event.

LODE: Descrizione di eventi

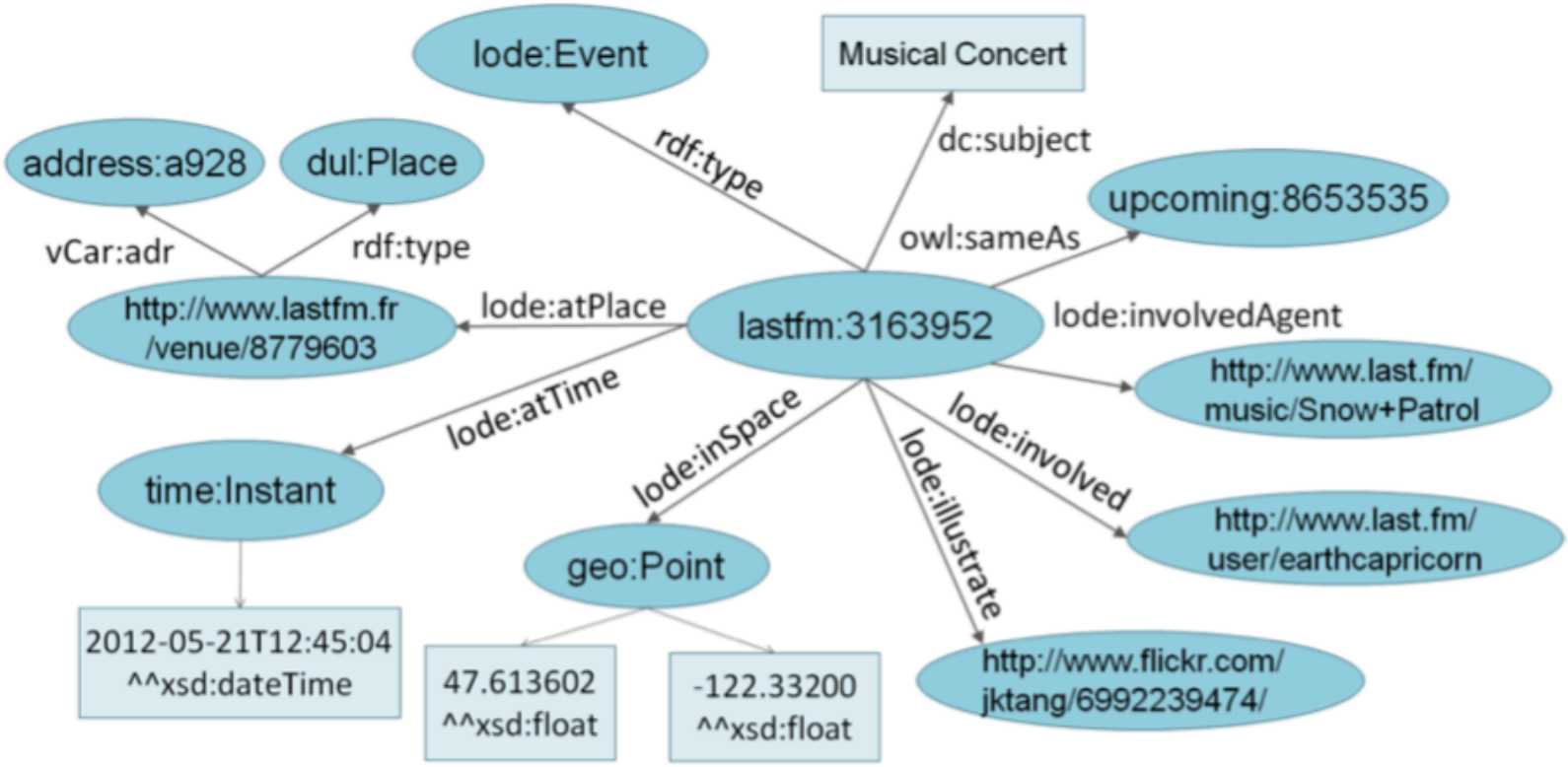


Fig. 2. The *Snow Patrol Concert* described with LODE ontology

Si noti l'utilizzo di vocabolari diversi che caratterizza il paradigma dei Linked Data