COMP718: Ontologies and Knowledge Bases

Lecture 5: Top-down Ontology Development I

Maria Keet

email: keet@ukzn.ac.za

home: http://www.meteck.org

School of Mathematics, Statistics, and Computer Science
University of KwaZulu-Natal, South Africa

6 March 2012

Outline

Introduction

- Ontology development: what to represent, and how?
  - Where do you start?
  - How can you avoid reinventing the wheel?
  - What things can guide you to make the process easier to carry out successfully?
  - How can you make the best of legacy material?
  - How can you make it interoperable with other ontologies?
- Foundational ontologies provide principal categories of kinds of things and relations to give a basic structure to a domain ontology
- Legacy resources can provide useful classes and properties for domain ontologies

Introduction

- Ontology development: what to represent, and how?
  - Where do you start?
  - How can you avoid reinventing the wheel?
  - What things can guide you to make the process easier to carry out successfully?
  - How can you make the best of legacy material?
  - How can you make it interoperable with other ontologies?
- Foundational ontologies provide principal categories of kinds of things and relations to give a basic structure to a domain ontology
- Legacy resources can provide useful classes and properties for domain ontologies

Why use a foundational ontology?

- Pros:
  - don't have to 'reinvent the wheel' with respect to the basic categories and relationships to represent the subject domain
  - improves overall quality with modelling guidance
  - facilitates interoperability among ontologies
  - is useful when subtle distinctions, recognizing disagreement, rigorous referential semantics, general abstractions, careful explanation and justification of ontological commitment, and mutual understanding are important
- Cons:
  - too abstract
  - too expressive and comprehensive for the envisioned ontology-driven information system
  - takes excessive effort to understand them in sufficient detail

Why use a foundational ontology?

- Pros:
  - don't have to 'reinvent the wheel' with respect to the basic categories and relationships to represent the subject domain
  - improves overall quality with modelling guidance
  - facilitates interoperability among ontologies
  - is useful when subtle distinctions, recognizing disagreement, rigorous referential semantics, general abstractions, careful explanation and justification of ontological commitment, and mutual understanding are important
- Cons:
  - too abstract
  - too expressive and comprehensive for the envisioned ontology-driven information system
  - takes excessive effort to understand them in sufficient detail

General notions and principal choices

- Provide a top-level with basic categories of kinds of entities
- Principal choices on universals, particulars and individual properties:
  - Properties as repeatable universals, belonging to different entities or as non-repeatable tropes, inhering only in a specific entity
  - Particulars as aggregations (collections) of properties or the properties inhere in some substrate (bare particular)
- Persistence, principal choices:
  - How do entities persist? How do entities change in time? (Due to different phases or due to (whole) instantiation of different properties at different times?) How are change and persistence related?

General notions and principal choices

- Provide a top-level with basic categories of kinds of entities
- Principal choices on universals, particulars and individual properties:
  - Properties as repeatable universals, belonging to different entities or as non-repeatable tropes, inhering only in a specific entity
  - Particulars as aggregations (collections) of properties or the properties inhere in some substrate (bare particular)
- Persistence, principal choices:
  - How do entities persist? How do entities change in time? (Due to different phases or due to (whole) instantiation of different properties at different times?) How are change and persistence related?

General notions and principal choices

- Provide a top-level with basic categories of kinds of entities
- Principal choices on universals, particulars and individual properties:
  - Properties as repeatable universals, belonging to different entities or as non-repeatable tropes, inhering only in a specific entity
  - Particulars as aggregations (collections) of properties or the properties inhere in some substrate (bare particular)
- Persistence, principal choices:
  - How do entities persist? How do entities change in time? (Due to different phases or due to (whole) instantiation of different properties at different times?) How are change and persistence related?

General notions and principal choices

- More choices:
  - Are space and time absolute or relative, atomic or not?
  - Localization: are there entities that are not in space/time (i.e., abstract), and is it possible to have different entities spatially or spatio-temporally colocalized?
- Principal choices, with common terminology:
  - Endurantist vs. Perdurantist
  - Universals vs. Particulars
  - Descriptive vs. Prescriptive
  - (Onto)Logical economy and multiplicative vs. reductionist

General notions and principal choices

- More choices:
  - Are space and time absolute or relative, atomic or not?
  - Localization: are there entities that are not in space/time (i.e., abstract), and is it possible to have different entities spatially or spatio-temporally colocalized?
- Principal choices, with common terminology:
  - Endurantist vs. Perdurantist
  - Universals vs. Particulars
  - Descriptive vs. Prescriptive
  - (Onto)Logical economy and multiplicative vs. reductionist

Outline

Descriptive Ontology for Linguistic and Cognitive Engineering

- Strong cognitive/linguistic bias:
  - Descriptive (as opposite to prescriptive) attitude
  - Categories mirror cognition, common sense, and the lexical structure of natural language
- Emphasis on cognitive invariants
- Categories as conceptual containers: no 'deep' metaphysical implications
- Focus on design rationale to allow easy comparison with different ontological options
- Rigorous, systematic, interdisciplinary approach
- Rich axiomatization
  - 37 basic categories
  - 7 basic relations
  - 80 axioms, 100 definitions, 20 theorems
- Rigorous quality criteria
- Documentation

Descriptive Ontology for Linguistic and Cognitive Engineering

- Strong cognitive/linguistic bias:
  - Descriptive (as opposite to prescriptive) attitude
  - Categories mirror cognition, common sense, and the lexical structure of natural language
- Emphasis on cognitive invariants
- Categories as conceptual containers: no 'deep' metaphysical implications
- Focus on design rationale to allow easy comparison with different ontological options
- Rigorous, systematic, interdisciplinary approach
- Rich axiomatization
  - 37 basic categories
  - 7 basic relations
  - 80 axioms, 100 definitions, 20 theorems
- Rigorous quality criteria
- Documentation

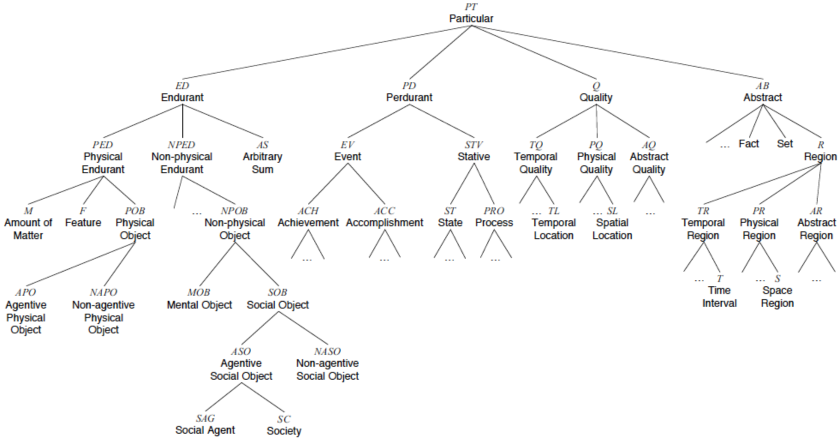
Descriptive Ontology for Linguistic and Cognitive Engineering

- Strong cognitive/linguistic bias:
  - Descriptive (as opposite to prescriptive) attitude
  - Categories mirror cognition, common sense, and the lexical structure of natural language
- Emphasis on cognitive invariants
- Categories as conceptual containers: no 'deep' metaphysical implications
- Focus on design rationale to allow easy comparison with different ontological options
- Rigorous, systematic, interdisciplinary approach
- Rich axiomatization
  - 37 basic categories
  - 7 basic relations
  - 80 axioms, 100 definitions, 20 theorems
- Rigorous quality criteria
- Documentation

Descriptive Ontology for Linguistic and Cognitive Engineering

- Strong cognitive/linguistic bias:
  - Descriptive (as opposite to prescriptive) attitude
  - Categories mirror cognition, common sense, and the lexical structure of natural language
- Emphasis on cognitive invariants
- Categories as conceptual containers: no 'deep' metaphysical implications
- Focus on design rationale to allow easy comparison with different ontological options
- Rigorous, systematic, interdisciplinary approach
- Rich axiomatization
  - 37 basic categories
  - 7 basic relations
  - 80 axioms, 100 definitions, 20 theorems
- Rigorous quality criteria
- Documentation

Outline of DOLCE categories



The African Wildlife Ontology and DOLCE

- Where does `Plant` fit in DOLCE?
  - as a subtype of Non-Agentive Physical Object
- Giraffes drink `Water`: where should we put `Water`?
  - as a subtype of Amount of Matter
- Impalas run (fast). Where should we put `Running`?
  - as a subtype of Process
- Lions eat impalas, and in the process, the impalas die. Where should we put `Death`?
  - as a subtype of Achievement...
- Generic examples of DOLCE's 'leaf' categories: see Table 1, p21 in the D18.pdf

The African Wildlife Ontology and DOLCE

- Where does `Plant` fit in DOLCE?
  - as a subtype of Non-Agentive Physical Object
- Giraffes drink `Water`: where should we put `Water`?
  - as a subtype of Amount of Matter
- Impalas run (fast). Where should we put `Running`?
  - as a subtype of Process
- Lions eat impalas, and in the process, the impalas die. Where should we put `Death`?
  - as a subtype of Achievement...
- Generic examples of DOLCE's 'leaf' categories: see Table 1, p21 in the D18.pdf

The African Wildlife Ontology and DOLCE

- Where does `Plant` fit in DOLCE?
  - as a subtype of Non-Agentive Physical Object
- Giraffes drink `Water`: where should we put `Water`?
  - as a subtype of Amount of Matter
- Impalas run (fast). Where should we put `Running`?
  - as a subtype of Process
- Lions eat impalas, and in the process, the impalas die. Where should we put `Death`?
  - as a subtype of Achievement...
- Generic examples of DOLCE's 'leaf' categories: see Table 1, p21 in the D18.pdf

DOLCE's basic relations

- Parthood
  - Between quality regions (immediate)
  - Between arbitrary objects (temporary)
- Constitution
- Participation
- Representation
- Dependence: Specific/generic constant dependence
- Inherence (between a quality and its host)
- Quale
  - Between a quality and its region (immediate, for unchanging entities)
  - Between a quality and its region (temporary, for changing entities)

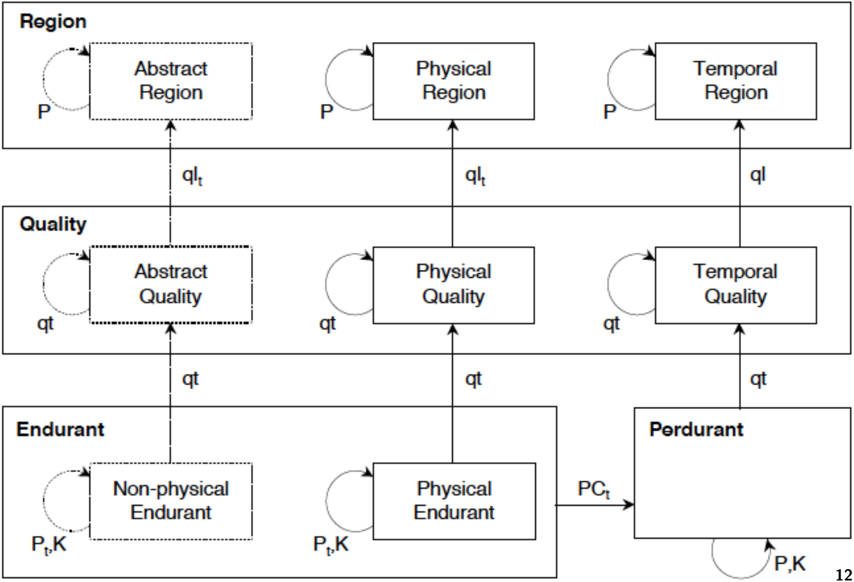
DOLCE's basic relations

- Parthood
  - Between quality regions (immediate)
  - Between arbitrary objects (temporary)
- Constitution
- Participation
- Representation
- Dependence: Specific/generic constant dependence
- Inherence (between a quality and its host)
- Quale
  - Between a quality and its region (immediate, for unchanging entities)
  - Between a quality and its region (temporary, for changing entities)

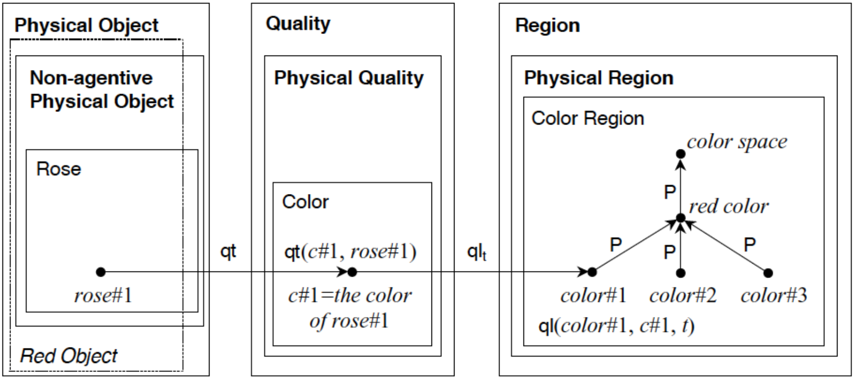
DOLCE's basic relations

- Parthood
  - Between quality regions (immediate)
  - Between arbitrary objects (temporary)
- Constitution
- Participation
- Representation
- Dependence: Specific/generic constant dependence
- Inherence (between a quality and its host)
- Quale
  - Between a quality and its region (immediate, for unchanging entities)
  - Between a quality and its region (temporary, for changing entities)

DOLCE's primitive relations between basic categories



DOLCE's basic relations w.r.t. qualities



Various commitments regarding 'attributes'

- Options:

Universalism

$$\begin{tikzcd} a \arrow[r, "\mathsf{inst}"] & F \\ b \arrow[ur, "\mathsf{inst}"'] & \end{tikzcd}$$

Trope theory

$$\begin{tikzcd} a & a_F \arrow[l, "\mathrm{I}"'] \arrow[r, "\in"] \arrow[d, leftrightarrow, "\approx"'] & {|F|_{\approx}} \\ b & b_F \arrow[l, "\mathrm{I}"] \arrow[ur, "\in"'] & \end{tikzcd}$$

Universals+Tropes

$$\begin{tikzcd} a & a_F \arrow[l, "\mathrm{I}"'] \arrow[r, "\mathsf{inst}"] & F \\ b & b_F \arrow[l, "\mathrm{I}"] \arrow[ur, "\mathsf{inst}"'] & \end{tikzcd}$$

- DOLCE: [*PerDurant*/*EnDurant*] *–qt– Quality –ql– Region*: use `Quality` and `Abstract` branches with `qt` (inherence) and `ql` (quale) object properties
- OWL: `DataProperty` with as domain class and range a datatype
  - More compact notation
  - But modelling based on arbitrary (and practical, application) decisions, increasing the chance of incompatibilities and less reusable

Various commitments regarding ‘attributes’

- Options:

Universalism

$$\begin{tikzcd} a \arrow[r, "\mathsf{inst}"] & F \\ b \arrow[ur, "\mathsf{inst}"'] & \end{tikzcd}$$

Trope theory

$$\begin{tikzcd} a & a_F \arrow[l, "\mathsf{I}"'] \arrow[r, "\in"] \arrow[d, leftrightarrow, "\approx"'] & {|F|_{\approx}} \\ b & b_F \arrow[l, "\mathsf{I}"] \arrow[ur, "\in"'] & \end{tikzcd}$$

Universals+Tropes

$$\begin{tikzcd} a & a_F \arrow[l, "\mathsf{I}"'] \arrow[r, "\mathsf{inst}"] & F \\ b & b_F \arrow[l, "\mathsf{I}"] \arrow[ur, "\mathsf{inst}"'] & \end{tikzcd}$$

- DOLCE: [*PerDurant* / *EnDurant*] *–qt– Quality –ql– Region*: use `Quality` and `Abstract` branches with `qt` (inherence) and `ql` (quale) object properties
- OWL: `DataProperty` with as domain class and range a datatype
  - More compact notation
  - But modelling based on arbitrary (and practical, application) decisions, increasing the chance of incompatibilities and less reusable

Various commitments regarding ‘attributes’

- Options:

Universalism

$$\begin{tikzcd} a \arrow[r, "\mathsf{inst}"] & F \\ b \arrow[ur, "\mathsf{inst}"'] & \end{tikzcd}$$

Trope theory

$$\begin{tikzcd} a & a_F \arrow[l, "\mathsf{I}"'] \arrow[r, "\in"] \arrow[d, leftrightarrow, "\approx"'] & {|F|_{\approx}} \\ b & b_F \arrow[l, "\mathsf{I}"] \arrow[ur, "\in"'] & \end{tikzcd}$$

Universals+Tropes

$$\begin{tikzcd} a & a_F \arrow[l, "\mathsf{I}"'] \arrow[r, "\mathsf{inst}"] & F \\ b & b_F \arrow[l, "\mathsf{I}"] \arrow[ur, "\mathsf{inst}"'] & \end{tikzcd}$$

- DOLCE: [*PerDurant*/*EnDurant*] *–qt– Quality –ql– Region*: use `Quality` and `Abstract` branches with `qt` (inherence) and `ql` (quale) object properties
- OWL: `DataProperty` with as domain class and range a datatype
  - More compact notation
  - But modelling based on arbitrary (and practical, application) decisions, increasing the chance of incompatibilities and less reusable

The Wildlife Ontology and DOLCE

- Giraffes eat leaves and twigs. how do `Plant` and `Twig` relate?
  -
- The elephant's tusks (ivory) are made of apatite (calcium phosphate); which DOLCE relation can be reused?
  -
- How would you represent the `Size` (`Height`, `Weight`, etc.) of an average adult elephant?
  - with *quality* and *quale*
  - OWL data properties
    -
    -
    -

The Wildlife Ontology and DOLCE

- Giraffes eat leaves and twigs. how do `Plant` and `Twig` relate?
  - (some type of) parthood relation
- The elephant's tusks (ivory) are made of apatite (calcium phosphate); which DOLCE relation can be reused?
  - constitution
- How would you represent the `Size` (`Height`, `Weight`, etc.) of an average adult elephant?
  - with *quality* and *quale*
  - OWL data properties
    - What is the data type; integer, float, real, string?
    - Measure in meter, feet, kg, lb?
    - Introduce "`ElephantHeight`", and also "`LionHeight`", "`GiraffeHeight`', "`ImpalaHeight`", etc.?

The Wildlife Ontology and DOLCE

- Giraffes eat leaves and twigs. how do `Plant` and `Twig` relate?
  - (some type of) parthood relation
- The elephant's tusks (ivory) are made of apatite (calcium phosphate); which DOLCE relation can be reused?
  - constitution
- How would you represent the `Size` (`Height`, `Weight`, etc.) of an average adult elephant?
  - with *quality* and *quale*
  - OWL data properties
    - What is the data type; integer, float, real, string?
    - Measure in meter, feet, kg, lb?
    - Introduce "`ElephantHeight`", and also "`LionHeight`", "`GiraffeHeight`', "`ImpalaHeight`", etc.?

The Wildlife Ontology and DOLCE

- Giraffes eat leaves and twigs. how do `Plant` and `Twig` relate?
  - (some type of) parthood relation
- The elephant's tusks (ivory) are made of apatite (calcium phosphate); which DOLCE relation can be reused?
  - constitution
- How would you represent the `Size` (`Height`, `Weight`, etc.) of an average adult elephant?
  - with *quality* and *quale*
  - OWL data properties
    - What is the data type; integer, float, real, string?
    - Measure in meter, feet, kg, lb?
    - Introduce "`ElephantHeight`", and also "`LionHeight`", "`GiraffeHeight`', "`ImpalaHeight`", etc.?

DOLCE's basics on universals

(Dd1) $\mathsf{RG}(\phi) \triangleq \Box\forall x(\phi(x) \rightarrow \Box\phi(x))$ (ϕ *is Rigid*)

(Dd2) $\mathsf{NEP}(\phi) \triangleq \Box\exists x(\phi(x))$ (ϕ *is Non-Empty*)

(Dd3) $\mathsf{DJ}(\phi,\psi) \triangleq \Box\neg\exists x(\phi(x) \wedge \psi(x))$ (ϕ *and* ψ *are Disjoint*)

(Dd4) $\mathsf{SB}(\phi,\psi) \triangleq \Box\forall x(\psi(x) \rightarrow \phi(x))$ (ϕ *Subsumes* ψ)

(Dd5) $\mathsf{EQ}(\phi,\psi) \triangleq \mathsf{SB}(\phi,\psi) \wedge \mathsf{SB}(\psi,\phi)$ (ϕ *and* ψ *are Equal*)

(Dd6) $\mathsf{PSB}(\phi,\psi) \triangleq \mathsf{SB}(\phi,\psi) \wedge \neg\mathsf{SB}(\phi,\psi)$ (ϕ *Properly Subsumes* ψ)

(Dd7) $\mathsf{L}(\phi) \triangleq \Box\forall\psi(\mathsf{SB}(\phi,\psi) \rightarrow \mathsf{EQ}(\phi,\psi))$ (ϕ *is a Leaf*)

(Dd8) $\mathsf{SBL}(\phi,\psi) \triangleq \mathsf{SB}(\phi,\psi) \wedge \mathsf{L}(\psi)$ (ψ *is a Leaf Subsumed by* ϕ)

(Dd9) $\mathsf{PSBL}(\phi,\psi) \triangleq \mathsf{PSB}(\phi,\psi) \wedge \mathsf{L}(\psi)$ (ψ *is a Leaf Properly Subsumed by* ϕ)

....

DOLCE's characterisation of categories

Physical Object

(Ad32)* $\mathsf{GK}(SC, SAG)$

(Ad30)* $\mathsf{GK}(NAPO, M)$

(Ad70)* $\mathsf{OGD}(F, NAPO)$

(Ad71)* $\mathsf{OSD}(MOB, APO)$

(Ad72)* $\mathsf{OGD}(SAG, APO)$

Feature

(Ad70)* $\mathsf{OGD}(F, NAPO)$

Non-physical Endurant

(Ad12)* $\mathsf{P}(x,y,t) \rightarrow (NPED(x) \leftrightarrow NPED(y))$

(Ad22)* $\mathsf{K}(x,y,t) \rightarrow (NPED(x) \leftrightarrow NPED(y))$

(Ad41)* $\mathsf{qt}(x,y) \rightarrow (AQ(x) \leftrightarrow (AQ(y) \vee NPED(y)))$

(Ad48)* $AQ(x) \rightarrow \exists! y(\mathsf{qt}(x,y) \wedge NPED(y))$

(Ad51)* $NPED(x) \rightarrow \exists \phi, y(\mathsf{SBL}(AQ, \phi) \wedge \mathsf{qt}(\phi, y, x))$

(Ad74)* $\mathsf{OD}(NPED, PED)$

... etc...

Can all that be used?

- DOLCE in KIF
- DOLCE in OWL:
  - DOLCE-Lite: simplified translations of Dolce2.0
  - Does *not* consider: modality, temporal indexing, relation composition
  - Different names are adopted for relations that have the same name but different arities in the FOL version
  - Some commonsense concepts have been added as examples
- DOLCE-2.1-Lite-Plus version includes some modules for Plans, Information Objects, Semiotics, Temporal relations, Social notions (collectives, organizations, etc.), a Reification vocabulary, etc.

Can all that be used?

- DOLCE in KIF
- DOLCE in OWL:
  - DOLCE-Lite: simplified translations of Dolce2.0
  - Does *not* consider: modality, temporal indexing, relation composition
  - Different names are adopted for relations that have the same name but different arities in the FOL version
  - Some commonsense concepts have been added as examples
- DOLCE-2.1-Lite-Plus version includes some modules for Plans, Information Objects, Semiotics, Temporal relations, Social notions (collectives, organizations, etc.), a Reification vocabulary, etc.

Can all that be used?

- DOLCE in KIF
- DOLCE in OWL:
  - DOLCE-Lite: simplified translations of Dolce2.0
  - Does *not* consider: modality, temporal indexing, relation composition
  - Different names are adopted for relations that have the same name but different arities in the FOL version
  - Some commonsense concepts have been added as examples
- DOLCE-2.1-Lite-Plus version includes some modules for Plans, Information Objects, Semiotics, Temporal relations, Social notions (collectives, organizations, etc.), a Reification vocabulary, etc.

Can all that be used?

- DOLCE in KIF
- DOLCE in OWL:
  - DOLCE-Lite: simplified translations of Dolce2.0
  - Does *not* consider: modality, temporal indexing, relation composition
  - Different names are adopted for relations that have the same name but different arities in the FOL version
  - Some commonsense concepts have been added as examples
- DOLCE-2.1-Lite-Plus version includes some modules for Plans, Information Objects, Semiotics, Temporal relations, Social notions (collectives, organizations, etc.), a Reification vocabulary, etc.

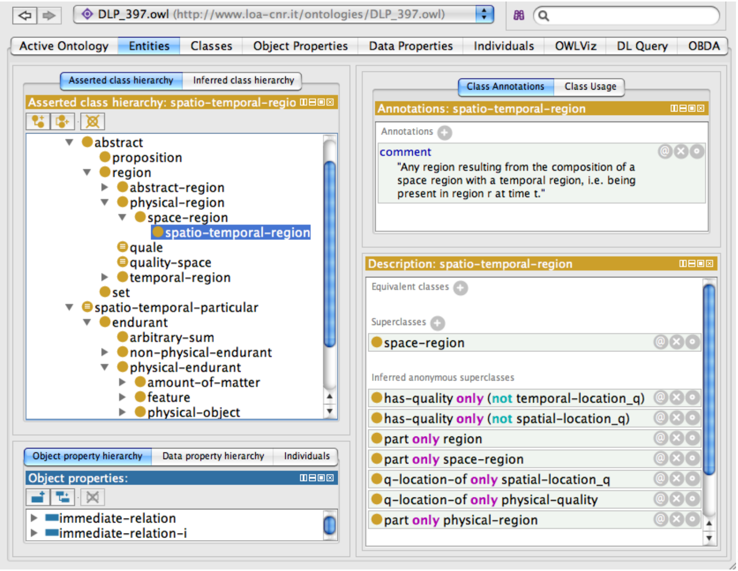
DLP3971

- Several Modules for (re)use: DOLCE-Lite, SocialUnits, SpatialRelations, ExtendedDnS, and others
- Still rather complex to understand (aside from using OWL-DL): Full DOLCE-Lite-Plus with 208 classes, 313 object properties, etc (check the "Active ontology" tab in Protégé) and basic DOLCE-Lite 37 classes, 70 object properties etc (in $\mathcal{SHI}$)
- Time for a DOLCE-Lite ultra-"ultralight"? e.g. for use with OWL 2 QL or OWL 2 EL
  - Current DOLCE Ultra Lite—DUL—uses friendly names and comments for classes and properties, has simple restrictions for classes, and includes into a unique file the main parts of DOLCE, D&S and other modules of DOLCE Lite+
  - BUT... is still in OWL-DL (OWL-Lite+Disjointness)
- http://wiki.loa-cnr.it/index.php/LoaWiki:Ontologies

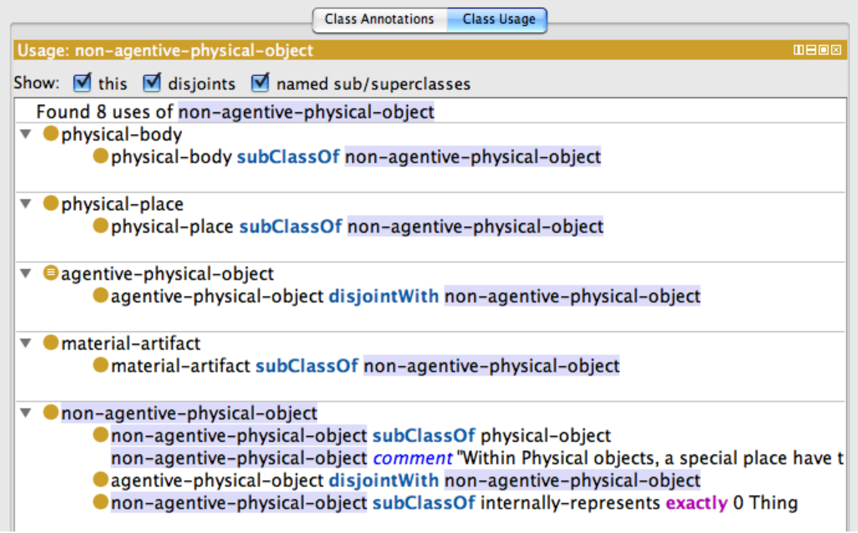
DLP3971

- Several Modules for (re)use: DOLCE-Lite, SocialUnits, SpatialRelations, ExtendedDnS, and others
- Still rather complex to understand (aside from using OWL-DL): Full DOLCE-Lite-Plus with 208 classes, 313 object properties, etc (check the "Active ontology" tab in Protégé) and basic DOLCE-Lite 37 classes, 70 object properties etc (in $\mathcal{SHI}$)
- Time for a DOLCE-Lite ultra-"ultralight"? e.g. for use with OWL 2 QL or OWL 2 EL
  - Current DOLCE Ultra Lite—DUL—uses friendly names and comments for classes and properties, has simple restrictions for classes, and includes into a unique file the main parts of DOLCE, D&S and other modules of DOLCE Lite+
  - BUT... is still in OWL-DL (OWL-Lite+Disjointness)
- `http://wiki.loa-cnr.it/index.php/LoaWiki:Ontologies`

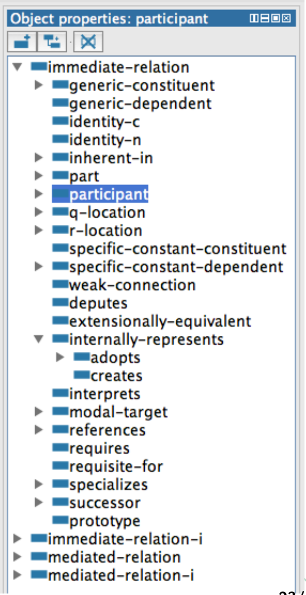
Examples


DLP_397.owl (http://www.loa-cnr.it/ontologies/DLP_397.owl)
Active Ontology
Entities
Classes
Object Properties
Data Properties
Individuals
OWLViz
DL Query
OBDA
Asserted class hierarchy
Inferred class hierarchy
Asserted class hierarchy: spatio-temporal-regio
abstract
proposition
region
abstract-region
physical-region
space-region
spatio-temporal-region
quale
quality-space
temporal-region
set
spatio-temporal-particular
endurant
arbitrary-sum
non-physical-endurant
physical-endurant
amount-of-matter
feature
physical-object
Object property hierarchy
Data property hierarchy
Individuals
Object properties:
immediate-relation
immediate-relation-i
Class Annotations
Class Usage
Annotations: spatio-temporal-region
Annotations
comment
"Any region resulting from the composition of a space region with a temporal region, i.e. being present in region r at time t."
Description: spatio-temporal-region
Equivalent classes
Superclasses
space-region
Inferred anonymous superclasses
has-quality only (not temporal-location_q)
has-quality only (not spatial-location_q)
part only region
part only space-region
q-location-of only spatial-location_q
q-location-of only physical-quality
part only physical-region

Examples


Class Annotations
Class Usage
Usage: non-agentive-physical-object
Show: this disjoints named sub/superclasses
Found 8 uses of non-agentive-physical-object
physical-body
physical-body subClassOf non-agentive-physical-object
physical-place
physical-place subClassOf non-agentive-physical-object
agentive-physical-object
agentive-physical-object disjointWith non-agentive-physical-object
material-artifact
material-artifact subClassOf non-agentive-physical-object
non-agentive-physical-object
non-agentive-physical-object subClassOf physical-object
non-agentive-physical-object comment "Within Physical objects, a special place have t
agentive-physical-object disjointWith non-agentive-physical-object
non-agentive-physical-object subClassOf internally-represents exactly 0 Thing

Comment: "The immediate relation holding between endurants and perdurants (e.g. in 'the car is running').Participation can be constant (in all parts of the perdurant, e.g. in 'the car is running'), or temporary (in only some parts, e.g. in 'I'm electing the president').A 'functional' participant is specialized for those forms of participation that depend on the nature of participants, processes, or on the intentionality of agentive participants. Traditional 'thematic role' should be mapped to functional participation.For relations holding between participants in a same perdurant, see the co-participates relation."



Outline

BFO Overview

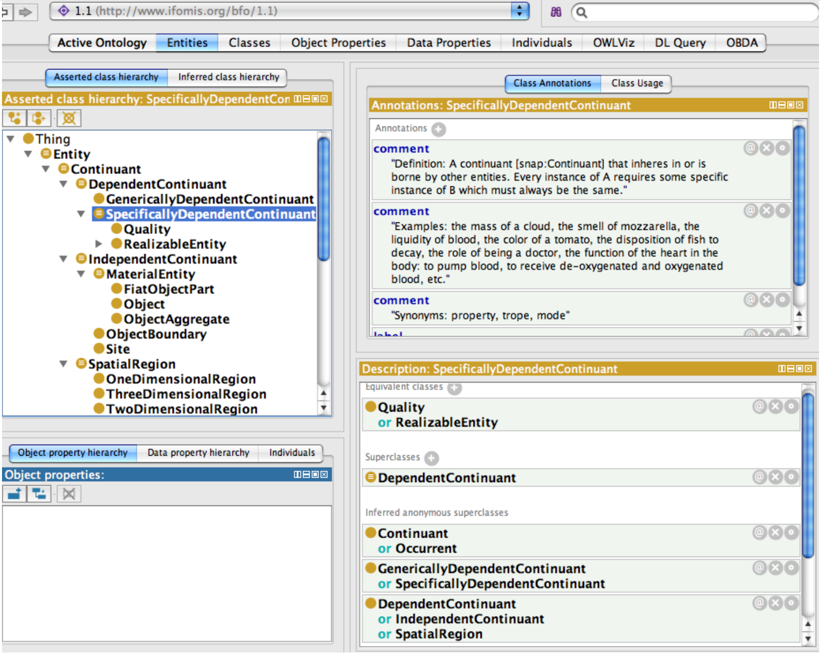
- Ontology as reality representation
- Aims at reconciling the so-called three-dimensionalist and four-dimensionalist views
  - A Snap ontology of endurants which is reproduced at each moment of time and is used to characterize static views of the world
  - Span ontology of happenings and occurrents and, more generally, of entities which persist in time by perduring
  - Endurants (Snap) or perdurants (Span)
- Limited granularity
- Heavily influenced by parthood relations, boundaries, dependence

BFO Taxonomy

```
bfo:Entity
  snap:Continuant
    snap:DependentContinuant
      snap:GenericalyDependentContinuant
      snap:SpecificalyDependentContinuant
        snap:Quality
        snap:RealizableEntity
          snap:Disposition
          snap:Function
          snap:Role
    snap:IndependentContinuant
      snap:MaterialEntity
          snap:Object
          snap:FiatObjectPart
          snap:ObjectAggregate
      snap:ObjectBoundary
      snap:Site
    snap:SpatialRegion
      snap:ZeroDimensionalRegion
      snap:OneDimensionalRegion
      snap:TwoDimensionalRegion
      snap:ThreeDimensionalRegion
```

```
span:Occurrent
  span:ProcessualEntity
    span:Process
    span:ProcessBoundary
    span:FiatProcessPart
    span:ProcessAggregate
    span:ProcessualContext
  span:SpatiotemporalRegion
    span:ConnectedTemporalRegion
      span:SpatiotemporalInstant
      span:SpatiotemporalInterval
    span:ScatteredSpatiotemporalRegion
  span:TemporalRegion
    span:ConnectedSpatiotemporalRegion
      span:TemporalInstant
      span:TemporalInterval
    span:ScatteredTemporalRegion
```

Overview


1.1 (http://www.ifomis.org/bfo/1.1)
Active Ontology
Entities
Classes
Object Properties
Data Properties
Individuals
OWLViz
DL Query
OBDA
Asserted class hierarchy
Inferred class hierarchy
Asserted class hierarchy: SpecificallyDependentCon
Thing
Entity
Continuant
DependentContinuant
GenericallyDependentContinuant
SpecificallyDependentContinuant
Quality
RealizableEntity
IndependentContinuant
MaterialEntity
FiatObjectPart
Object
ObjectAggregate
ObjectBoundary
Site
SpatialRegion
OneDimensionalRegion
ThreeDimensionalRegion
TwoDimensionalRegion
Object property hierarchy
Data property hierarchy
Individuals
Object properties:
Class Annotations
Class Usage
Annotations: SpecificallyDependentContinuant
Annotations
comment
"Definition: A continuant [snap:Continuant] that inheres in or is borne by other entities. Every instance of A requires some specific instance of B which must always be the same."
comment
"Examples: the mass of a cloud, the smell of mozzarella, the liquidity of blood, the color of a tomato, the disposition of fish to decay, the role of being a doctor, the function of the heart in the body: to pump blood, to receive de-oxygenated and oxygenated blood, etc."
comment
"Synonyms: property, trope, mode"
Description: SpecificallyDependentContinuant
Equivalent classes
Quality
or RealizableEntity
Superclasses
DependentContinuant
Inferred anonymous superclasses
Continuant
or Occurrent
GenericallyDependentContinuant
or SpecificallyDependentContinuant
DependentContinuant
or IndependentContinuant
or SpatialRegion

The Wildlife Ontology and BFO

- Exercise: revisit the Wildlife & DOLCE and find corresponding BFO categories
  - Non-Agentive Physical Object, Amount of Matter, Process, and Achievement
  - parthood, constitution, quality & quale
- Issues
  - Generally: to do this in a transparent and reusable way, we need a mapping between the two foundational ontologies
  - Immediacy: What with the relations?
  - There is a bfo-ro.owl to integrate relations of the Relation Ontology with BFO (extensions under consideration)

The Wildlife Ontology and BFO

- Exercise: revisit the Wildlife & DOLCE and find corresponding BFO categories
  - Non-Agentive Physical Object, Amount of Matter, Process, and Achievement
  - parthood, constitution, quality & quale
- Issues
  - Generally: to do this in a transparent and reusable way, we need a mapping between the two foundational ontologies
  - Immediacy: What with the relations?
  - There is a `bfo-ro.owl` to integrate relations of the Relation Ontology with BFO (extensions under consideration)

The Wildlife Ontology and BFO

- Exercise: revisit the Wildlife & DOLCE and find corresponding BFO categories
  - Non-Agentive Physical Object, Amount of Matter, Process, and Achievement
  - parthood, constitution, quality & quale
- Issues
  - Generally: to do this in a transparent and reusable way, we need a mapping between the two foundational ontologies
  - Immediacy: What with the relations?
  - There is a `bfo-ro.owl` to integrate relations of the Relation Ontology with BFO (extensions under consideration)

Overview

- BFO 1.1 in OWL with 39 classes, no object or data properties, in $\mathcal{ALC}$.
- There is a `bfo-ro.owl` to integration relations of the Relation Ontology with BFO (extensions under consideration)
- Version in Isabelle (mainly part-wholes, but not all categories)
- Version in OBO (the original Gene Ontology format, with limited, but expanding, types of relationships)
- Version in Prover9 (first order logic model checker and theorem prover)

Overview

- BFO 1.1 in OWL with 39 classes, no object or data properties, in $\mathcal{ALC}$.
- There is a `bfo-ro.owl` to integration relations of the Relation Ontology with BFO (extensions under consideration)
- Version in Isabelle (mainly part-wholes, but not all categories)
- Version in OBO (the original Gene Ontology format, with limited, but expanding, types of relationships)
- Version in Prover9 (first order logic model checker and theorem prover)

Overview

- BFO 1.1 in OWL with 39 classes, no object or data properties, in $\mathcal{ALC}$.
- There is a `bfo-ro.owl` to integration relations of the Relation Ontology with BFO (extensions under consideration)
- Version in Isabelle (mainly part-wholes, but not all categories)
- Version in OBO (the original Gene Ontology format, with limited, but expanding, types of relationships)
- Version in Prover9 (first order logic model checker and theorem prover)

BFO Core

- A non-extensional temporal mereology with collections, sums, and universals
- BFO as a collection of smaller theories
  - EMR, QSizeR, RBG, QDiaSizeR, ..., Adjacency, Collections, SumsPartitions, Universals, Instantiation, ExtensionsOfUniversals, PartonomicInclusion, UniversalParthood
- Reference material `http://www.ifomis.org/bfo/fol` and `http://www.acsu.buffalo.edu/~bittner3/Theories/BFO/`

Section of one of the sub-theories in BFO Core

```
theory UniversalParthood

imports ExtensionsOfUniversals PartonomicInclusion

begin

consts


UPt1 :: Un => Un => Ti => o
UPt2 :: Un => Un => Ti => o
UPt12 :: Un => Un => Ti => o

UP1 :: Un => Un => o
UP2 :: Un => Un  => o
UP12 :: Un => Un  => o

defs

UPt1-def: UPt1(c,d,t) == (ALL x. (Inst(x,c,t) --> (EX y. (Inst(y,d,t) & P(x,y,t)))))
UPt2-def: UPt2(c,d,t) == (ALL y. (Inst(y,d,t) --> (EX x. (Inst(x,c,t) & P(x,y,t)))))
UPt12-def: UPt12(c,d,t) == UPt1(c,d,t) & UPt2(c,d,t)

UP1-def: UP1(c,d) == (ALL t. UPt1(c,d,t))
UP2-def: UP2(c,d) == (ALL t. UPt2(c,d,t))
UP12-def: UP12(c,d) == (ALL t. UPt12(c,d,t))
```

The Relation Ontology

- Definitions for *is_a*, *part_of*, *integral_part_of*, *proper_part_of*, *located_in*, *contained_in*, *adjacent_to*, *transformation_of*, *derives_from*, *preceded_by*, *has_participant*, *has_agent*, *instance_of*
- Proposed extensions under consideration, among others:
  - Relations between generically dependent continuants and specifically dependent continuants (a.o., concretizes, *has_quality*, *has_function*, ...)
  - A relation between a process and a process or quality (*regulates*)
  - Refinements on *derived_from*
  - Measurements (*has_value*, *of_dimension*, ...)

The Relation Ontology

- Definitions for *is_a*, *part_of*, *integral_part_of*, *proper_part_of*, *located_in*, *contained_in*, *adjacent_to*, *transformation_of*, *derives_from*, *preceded_by*, *has_participant*, *has_agent*, *instance_of*
- Proposed extensions under consideration, among others:
  - Relations between generically dependent continuants and specifically dependent continuants (a.o., concretizes, *has_quality*, *has_function*, ...)
  - A relation between a process and a process or quality (*regulates*)
  - Refinements on *derived_from*
  - Measurements (*has_value*, *of_dimension*, ...)

The Relation Ontology

- **Note**: The OBO Relation ontology is undergoing substantial changes: Core domain-independent relations will live in BFO, Biology specific relations (defined in terms of core relations) will live in RO (http://groups.google.com/group/obo-relations/browse_thread/thread/29fc616eb570f7dc/fc0647f190b5f178)
- BFO will likely include the follow relations:
  BFO_0000050 part of
  BFO_0000051 has part
  BFO_0000056 participates in
  BFO_0000057 has participant
  BFO_0000062 preceded by
  BFO_0000063 precedes
  BFO_0000060 immediately preceded by
  BFO_0000061 immediately precedes
- Discuss.

Outline

Ontologies and choices

- Other more or less used foundational ontologies, a.o.:
  - GFO
  - SUMO
  - OCHRE
  - UFO
  - ...
- Within WonderWeb project: a (future) aim to develop a library of foundational ontologies with mappings between them: choose your pet ontology and be interoperable with the others
- Exercise: examine `DolceliteBFOinDLandMSyntax.pdf` (or their respective OWL files) and spot commonalities and differences between DOLCE and BFO (or any two other foundational ontologies)

Ontologies and choices

- Other more or less used foundational ontologies, a.o.:
  - GFO
  - SUMO
  - OCHRE
  - UFO
  - ...
- Within WonderWeb project: a (future) aim to develop a library of foundational ontologies with mappings between them: choose your pet ontology and be interoperable with the others
- Exercise: examine `DolceliteBFOinDLandMSyntax.pdf` (or their respective OWL files) and spot commonalities and differences between DOLCE and BFO (or any two other foundational ontologies)

Ontologies and choices

- Other more or less used foundational ontologies, a.o.:
  - GFO
  - SUMO
  - OCHRE
  - UFO
  - ...
- Within WonderWeb project: a (future) aim to develop a library of foundational ontologies with mappings between them: choose your pet ontology and be interoperable with the others
- Exercise: examine `DolceliteBFOinDLandMSyntax.pdf` (or their respective OWL files) and spot commonalities and differences between DOLCE and BFO (or any two other foundational ontologies)

A relation ontology?

- What are the 'core' and primitive relations necessary to develop a domain ontology?
- Do we need a *separate* ontology for relations, or integrated in a foundational ontology?
- Philosophers do not agree on the answers, but the modellers and engineers need agreement to facilitate interoperability among ontologies

A relation ontology?

- What are the ‘core’ and primitive relations necessary to develop a domain ontology?
- Do we need a *separate* ontology for relations, or integrated in a foundational ontology?
- Philosophers do not agree on the answers, but the modellers and engineers need agreement to facilitate interoperability among ontologies

A relation ontology?

- What are the ‘core’ and primitive relations necessary to develop a domain ontology?
- Do we need a *separate* ontology for relations, or integrated in a foundational ontology?
- Philosophers do not agree on the answers, but the modellers and engineers need agreement to facilitate interoperability among ontologies

Other relation ontologies

- The Relation Ontology (Smith et al, 2005, Genome Biol.) is not the only 'relation ontology'—but no other claims to be *the* relation ontology
- There are "**RBox**es" that can be seen as a relation ontology, e.g., containing
  - Part-whole relations (next lecture)
  - Spatial relations (RCC)
  - Temporal relations (Allen)

Other relation ontologies

- The Relation Ontology (Smith et al, 2005, Genome Biol.) is not the only ‘relation ontology’—but no other claims to be *the* relation ontology
- There are “**RBox**es” that can be seen as a relation ontology, e.g., containing
  - Part-whole relations (next lecture)
  - Spatial relations (RCC)
  - Temporal relations (Allen)

The General Formal Ontology

- "A Foundational Ontology for Conceptual Modelling" (Herre, 2010) [Note: actually, UFO is more so]
- A component of an Integrated System of Foundational Ontologies
- (3D) objects and (4D) processes
- Admitting universals, concepts, and symbol structures and their interrelations
- GFO is intended to be the basis for a novel theory of ontological modelling which combines declarative specifications with algorithmic procedures
- Module for functions and a module for roles
- GFO is designed for applications, firstly in medical, biological, and biomedical areas, but also in the fields of economics and sociology

The General Formal Ontology

- “A Foundational Ontology for Conceptual Modelling” (Herre, 2010) [Note: actually, UFO is more so]
- **A component of an Integrated System of Foundational Ontologies**
- (3D) objects and (4D) processes
- Admitting universals, concepts, and symbol structures and their interrelations
- GFO is intended to be the basis for a novel theory of ontological modelling which combines declarative specifications with algorithmic procedures
- Module for functions and a module for roles
- GFO is designed for applications, firstly in medical, biological, and biomedical areas, but also in the fields of economics and sociology

The General Formal Ontology

- "A Foundational Ontology for Conceptual Modelling" (Herre, 2010) [Note: actually, UFO is more so]
- **A component of an Integrated System of Foundational Ontologies**
- (3D) objects and (4D) processes
- Admitting universals, concepts, and symbol structures and their interrelations
- GFO is intended to be the basis for a novel theory of ontological modelling which combines declarative specifications with algorithmic procedures
- Module for functions and a module for roles
- GFO is designed for applications, firstly in medical, biological, and biomedical areas, but also in the fields of economics and sociology

The General Formal Ontology

- "A Foundational Ontology for Conceptual Modelling" (Herre, 2010) [Note: actually, UFO is more so]
- **A component of an Integrated System of Foundational Ontologies**
- (3D) objects and (4D) processes
- Admitting universals, concepts, and symbol structures and their interrelations
- GFO is intended to be the basis for a novel theory of ontological modelling which combines declarative specifications with algorithmic procedures
- Module for functions and a module for roles
- GFO is designed for applications, firstly in medical, biological, and biomedical areas, but also in the fields of economics and sociology

The General Formal Ontology

- “A Foundational Ontology for Conceptual Modelling” (Herre, 2010) [Note: actually, UFO is more so]
- **A component of an Integrated System of Foundational Ontologies**
- (3D) objects and (4D) processes
- Admitting universals, concepts, and symbol structures and their interrelations
- GFO is intended to be the basis for a novel theory of ontological modelling which combines declarative specifications with algorithmic procedures
- Module for functions and a module for roles
- GFO is designed for applications, firstly in medical, biological, and biomedical areas, but also in the fields of economics and sociology

The General Formal Ontology

- “A Foundational Ontology for Conceptual Modelling” (Herre, 2010) [Note: actually, UFO is more so]
- **A component of an Integrated System of Foundational Ontologies**
- (3D) objects and (4D) processes
- Admitting universals, concepts, and symbol structures and their interrelations
- GFO is intended to be the basis for a novel theory of ontological modelling which combines declarative specifications with algorithmic procedures
- Module for functions and a module for roles
- GFO is designed for applications, firstly in medical, biological, and biomedical areas, but also in the fields of economics and sociology

The General Formal Ontology

- "A Foundational Ontology for Conceptual Modelling" (Herre, 2010) [Note: actually, UFO is more so]
- **A component of an Integrated System of Foundational Ontologies**
- (3D) objects and (4D) processes
- Admitting universals, concepts, and symbol structures and their interrelations
- GFO is intended to be the basis for a novel theory of ontological modelling which combines declarative specifications with algorithmic procedures
- Module for functions and a module for roles
- GFO is designed for applications, firstly in medical, biological, and biomedical areas, but also in the fields of economics and sociology

The General Formal Ontology

- Three-layered meta-ontological architecture
  - Abstract core level (ACO)
  - The entities of the world (ATO) are exhaustively divided into *categories* and *individuals*, where individuals instantiate categories, and among individuals, there is a distinction between objects and attributives
  - Basic level ontology: contains all relevant top-level distinctions and categories

The General Formal Ontology

- Three-layered meta-ontological architecture
  - Abstract core level (ACO)
  - The entities of the world (ATO) are exhaustively divided into *categories* and *individuals*, where individuals instantiate categories, and among individuals, there is a distinction between objects and attributives
  - Basic level ontology: contains all relevant top-level distinctions and categories

The General Formal Ontology

- Three-layered meta-ontological architecture
  - Abstract core level (ACO)
  - The entities of the world (ATO) are exhaustively divided into *categories* and *individuals*, where individuals instantiate categories, and among individuals, there is a distinction between objects and attributives
  - Basic level ontology: contains all relevant top-level distinctions and categories

The General Formal Ontology

- Three-layered meta-ontological architecture
  - Abstract core level (ACO)
  - The entities of the world (ATO) are exhaustively divided into *categories* and *individuals*, where individuals instantiate categories, and among individuals, there is a distinction between objects and attributives
  - Basic level ontology: contains all relevant top-level distinctions and categories

Basic categories

- Category (concept, universal, symbol structure)
- Individuals, divided into
  - Space-time entities (something in which concrete entities can be located),
  - Abstract individuals (π, idealised prototypical individuals),
  - Concrete individuals (this pen),
    - Presentials, perpetuants ($\sim$ endurant), with amount of substrate and material object
    - Processual structure ($\sim$ perdurant), with processes and occurrents
  - Attributives (a.o. properties, roles, functions, dispositions)

Basic categories

- Category (concept, universal, symbol structure)
- Individuals, divided into
  - Space-time entities (something in which concrete entities can be located),
  - Abstract individuals (π, idealised prototypical individuals),
  - Concrete individuals (this pen),
    - Presentials, perpetuants ($\sim$ endurant), with amount of substrate and material object
    - Processual structure ($\sim$ perdurant), with processes and occurrents
  - Attributives (a.o. properties, roles, functions, dispositions)

Basic categories

- Category (concept, universal, symbol structure)
- Individuals, divided into
    - Space-time entities (something in which concrete entities can be located),
    - Abstract individuals (π, idealised prototypical individuals),
    - Concrete individuals (this pen),
        - Presentials, perpetuants ($\sim$ endurant), with amount of substrate and material object
        - Processual structure ($\sim$ perdurant), with processes and occurrents
    - Attributives (a.o. properties, roles, functions, dispositions)

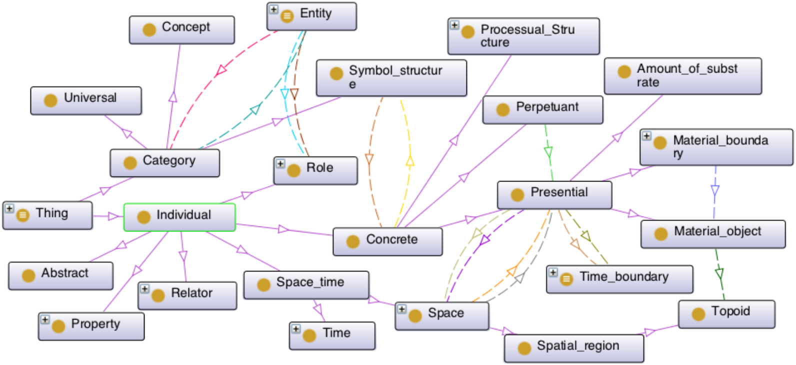
Basic categories

- Category (concept, universal, symbol structure)
- Individuals, divided into
  - Space-time entities (something in which concrete entities can be located),
  - Abstract individuals (π, idealised prototypical individuals),
  - Concrete individuals (this pen),
    - Presentials, perpetuants ($\sim$ endurant), with amount of substrate and material object
    - Processual structure ($\sim$ perdurant), with processes and occurrents
  - Attributives (a.o. properties, roles, functions, dispositions)

Basic relations

- Existential dependency
- instantiation
- parthood relations for time, space, material structures, processes
- coincidence, adjacent
- occupation
- participation
- causality

Section of GFO



Outline

1. DOLCE
   - Overview
   - Formalisations and implementations
2. BFO
   - Overview
   - Formalisations and implementations
   - Relation Ontology
3. More foundational ontologies
   - On relation ontologies and RBoxes
   - GFO as 'super' foundational
4. Making a difference?

Does using a foundational ontology in ontology development make a difference?

- Assumptions:
  - It facilitates ontology development because one does not have to reinvent the wheel concerning basic categories and relations, and
  - Using a foundational ontology improves overall quality and interoperability
- Criticisms (other assumptions):
  - Foundational ontologies are too abstract, too expressive, too comprehensive for 'simple' or domain ontologies,
  - It takes too much time to understand them in sufficient detail
  - Expressivity issues
  - Mismatches, such as foundational ontology's take on how to represent attributes vs. OWL's data properties

Does using a foundational ontology in ontology development make a difference?

- Assumptions:
  - It facilitates ontology development because one does not have to reinvent the wheel concerning basic categories and relations, and
  - Using a foundational ontology improves overall quality and interoperability
- Criticisms (other assumptions):
  - Foundational ontologies are too abstract, too expressive, too comprehensive for 'simple' or domain ontologies,
  - It takes too much time to understand them in sufficient detail
  - Expressivity issues
  - Mismatches, such as foundational ontology's take on how to represent attributes vs. OWL's data properties

Does using a foundational ontology in ontology development make a difference?

- Assumptions:
  - It facilitates ontology development because one does not have to reinvent the wheel concerning basic categories and relations, and
  - Using a foundational ontology improves overall quality and interoperability
- Criticisms (other assumptions):
  - Foundational ontologies are too abstract, too expressive, too comprehensive for 'simple' or domain ontologies,
  - It takes too much time to understand them in sufficient detail
  - Expressivity issues
  - Mismatches, such as foundational ontology's take on how to represent attributes vs. OWL's data properties

What to validate?

- Do developers *voluntarily* choose to commence domain ontology development with a foundational ontology?
- If so: is their ontology larger, do they really reuse the foundational ontology, is it of better quality and indeed better interoperable?
- (which easily can be written into falsifiable hypotheses)

What to validate?

- Do developers *voluntarily* choose to commence domain ontology development with a foundational ontology?
- If so: is their ontology larger, do they really reuse the foundational ontology, is it of better quality and indeed better interoperable?
- (which easily can be written into falsifiable hypotheses)

How to validate it? Controlled experiment (1/2)

- Lecture on purpose and usefulness of using a foundational ontology and overview of its contents (3-4 hours);
- Divide into smaller groups of 1-4 participants;
- Provide participants with instructions, being:
  - Develop a domain ontology about computers;
  - You have the following input options:
    – *tabula rasa*, i.e., start from scratch with an empty OWL ontology and do not import anything;
    – Use an OWLized foundational ontology (options provided: DOLCE, BFO, GFO);
    – And/or use the OWLized taxonomy of part-whole relations;
  - Name your ontology with the names of the group participants;
  - Time to develop the computer ontology: 24h from start to handing it in;
  - The ontology will not be graded, but is part of an experiment

How to validate it? Controlled experiment (2/2)

- Evaluation:
  - Assessment of the OWL files on usage of foundational ontologies, ontology metrics (language used, classes and object properties added etc.), errors made;
  - Open questions with the participants regarding motivations of (non-)usage and modelling issues.

Results

- 52 people, novice ontology developers, background in CS
- Developed 18 ontologies, 6 with a foundational ontology
- Quantitative and qualitative data:

Results: quantitative

Parameter ⇒ **Group ⇓**		New entities: class	obj. prop.	data prop.	individuals	New class axioms
All	Average	23.4	3.5	1.1	5.8	32.7
	Median	22	4	0	2	32
	StDev	10.1	2.0	2.0	8.8	14.3
Found. onto. reuse	Average	29	2	2,6	4.2	37.2
	Median	24	2	2	3	34
	StDev	11.9	2.1	3.0	3.4	14.0
From scratch	Average	21.1	4.1	0.4	6.5	30.8
	Median	18.5	4	0	1	27
	StDev	8.7	1.6	1.2	10.3	14.6
Inspect found. onto.	Average	22.8	3.8	0	5.4	31
	Median	17	4	0	0	22
	StDev	12.2	1.5	0	11.5	17.5

Table: Basic analysis of the new additions to the submitted ontologies; numbers are rounded off.

Results: quantitative

- 52 people, novice ontology developers, background in CS
- Developed 18 ontologies, 6 with a foundational ontology
- Student t-test:
  - For new classes, $p=0.145$, hence, barely not significant to claim starting with a foundational ontology significantly speeds up ontology development
  - For new class axioms, $p=0.420$, hence, one cannot conclude anything either way
  - For new object properties, $p=0.043$, hence, those who started with a foundational ontology added significantly less properties than those who started from scratch

Results: qualitative

- None of the 18 ontologies contained is-a vs. part-of errors
- Several is-a v.s instance-of mistakes (e.g., types of processors and motherboards were modelled as instances)
- Domain and range restrictions
- `NonSimpleRoleInNumberRestriction` (interaction of cardinality and characteristics of an object property)
- Naming the ontology vs. naming the OWL file
- Where in DOLCE to put `Computer`? How to define it?

Discussion

- Reuse of entities vs. too comprehensive and too complicated
  - Reuse: yes, significantly the object properties
  - But also that 2 groups deleted "unnecessary" branches of DOLCE
  - Unclear why DOLCE and not the simpler BFO or larger GFO
- Quality and interoperability
  - Quality difficult to measure anyhow
  - Integration easier among the 6 who used a foundational ontology, and with other ontologies (there is one about software and programs, also using DOLCE)
  - Integration/harmonisation hampered in the 12 others (es parte de, compuesta por, has part, etc.)
  - Using a foundational ontology may not help with ontological analysis of an entity (e.g.: is software a physical or a non-physical object?)

Discussion

- Other factors
  - English keywords vs. DL symbols in Protégé
  - Is 1/3 many or few?
  - Time allocated to the development
  - Enriching methodologies: we'll discuss this in lecture 8

Conclusions

- Investigation of assumptions surrounding foundational ontology reuse showed benefits of using one
- One third of the groups chose to use DOLCE (w./w.o. part-whole relations)
- On average, those who commenced with a foundational ontology added more classes, more class axioms, and significantly less object properties
- No errors in is-a vs. part-of
- Comprehensive results showed that the 'cost' incurred in spending time getting acquainted with a foundational ontology compared to starting from scratch was more than made up for in better quality and interoperability

Summary

1. DOLCE
   - Overview
   - Formalisations and implementations
2. BFO
   - Overview
   - Formalisations and implementations
   - Relation Ontology
3. More foundational ontologies
   - On relation ontologies and RBoxes
   - GFO as 'super' foundational
4. Making a difference?