Ontology-driven information systems

Ontology-Driven Information Systems

- Every IS ***has*** its own ontology (either implicit or explicit)
- The ODIS perspective: ***explicit*** ontologies play a ***central*** role, driving ***all*** aspects and components of an IS

- Two (main) dimensions to assess the role of an explicit ontology:
  - ***temporal dimension****:* development time vs. run time
  - ***structural dimension:*** impact on the various IS components:
    - ***database component***
    - ***application program***
    - ***user interface***

Temporal dimension: *development time*

- Two scenarios:
  - A pre-existing ***ontology library*** containing domain and task ontologies as "main building blocks" to be adapted and rused
    - standard IS: the ontology content is *embedded* in the standard components
    - ODIS: an *application ontology* is built by specializing domain and task ontologies taken from the library
  - Only an ***upper-level ontology*** available: not building blocks, but ***conceptual tools*** (analogous to other CASE tools)

- Two kinds of development:
  - IS ***engineering***
  - IS ***re-engineering***

Temporal dimension: *run time*

- Ontology-***aware*** IS: the IS just uses the ontology for some specific purpose
- Ontology-***driven*** IS: the ontology is a ***central component*** of the IS, cooperating at run time towards its “higher” overall goal

- Important application: ***inter-agent communication***

Structural dimension: *the database component*

- Development time:
  - support to ***requirement analysis and conceptual modelling*** (integrated with lexical resources like WordNet)
  - development of a ***global conceptual schema*** (DB integration)
- Run time:
  - mediation-based approach to ***information integration***
  - ***intensional queries***

Structural dimension: *the user-interface component*

- Development time:
  - Generation of ***form-based interfaces*** (constraints checking)
- Run time:
  - Support ***quering and browsing the ontology*** itself:
    - better understanding of the vocabulary
    - queries at the desired level of specificity
  - ***Vocabulary detaching***:
    - user free to adopt his own NL terms (mapped - after disambiguation - to the IS vocabulary with the help of the ontology)

Structural dimension:
the application program component

- Development time:
  - Generation of the static part of a program (type structure)
  - Support to OO design

- Run time:
  - Explicit account of the ***ontological commitment*** of an application program
  - Increase of the ***transparency*** of application software

The problem of primitives

The formal tools of ontological analysis

- Theory of Essence and Identity
- Theory of Parts (Mereology)
- Theory of Unity and Plurality
- Theory of Dependence
- Theory of Composition and Constitution
- Theory of Properties and Qualities

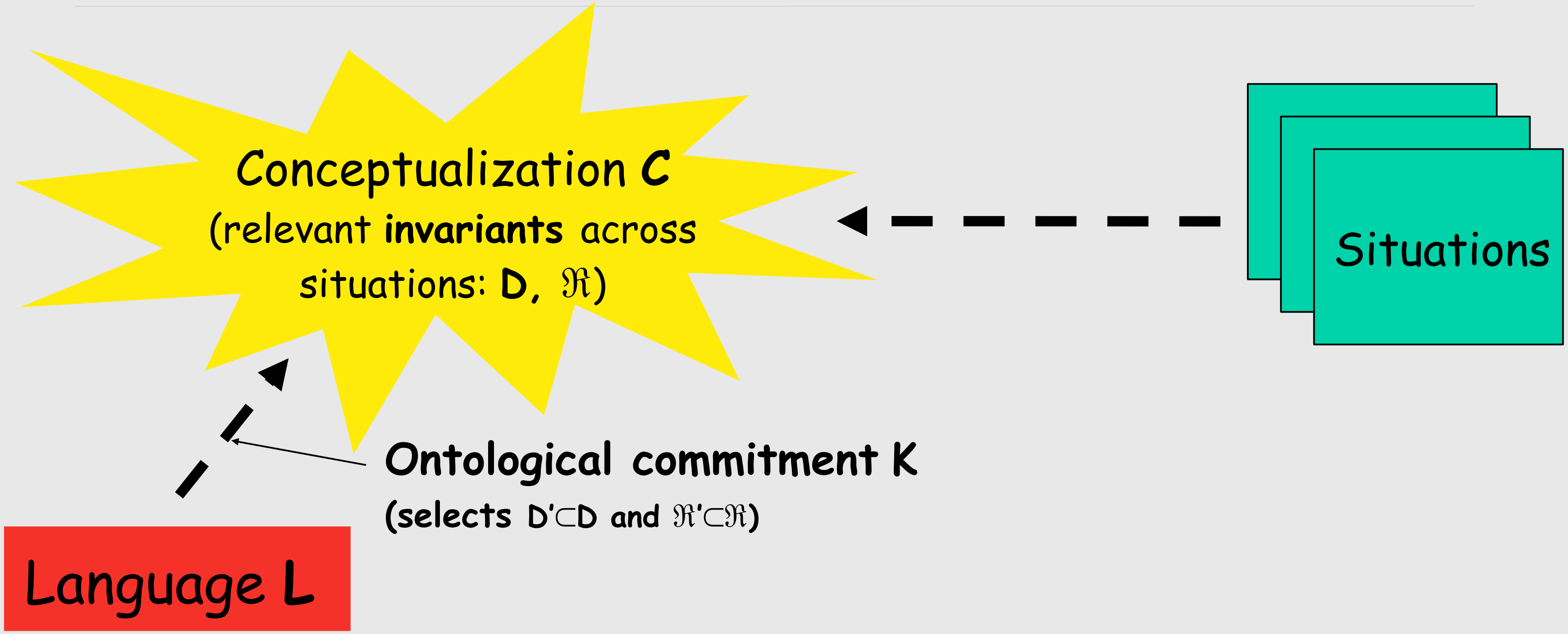
The basis for a common ontology vocabulary

Idea of Chris Welty, IBM Watson Research Centre, while visiting our lab in 2000

Formal Ontology

- Theory of ***formal distinctions and connections*** within:
  - entities of the world, as we perceive it (***particulars***)
  - categories we use to talk about such entities (***universals***)
- Why ***formal****?*
  - Two meanings: ***rigorous*** and ***general***
  - Formal logic: connections between truths - neutral wrt ***truth***
  - Formal ontology: connections between things - neutral wrt ***reality***
- ***NOTE:*** **"represented in a formal language" is not enough for being formal in the above sense!**
- **(*Analytic ontology* may be a better term to avoid this confusion)**

The first steps of ontological analysis



- Be clear about the ***domain of discourse*** (existence...)
- Choose the relevant ***concepts and conceptual relations***
- Choose the ***primive relations***
- Choose meaningful ***names*** for these

Mereology as an example of formal ontological analysis

- Primitive: ***proper part-of*** relation (PP)
  - asymmetric
  - transitive
  - **$Pxy =_{def} PPxy \vee x=y$**
  - **$Oxy =_{def} \exists z(Pzx \wedge Pzy)$**
- Axioms:

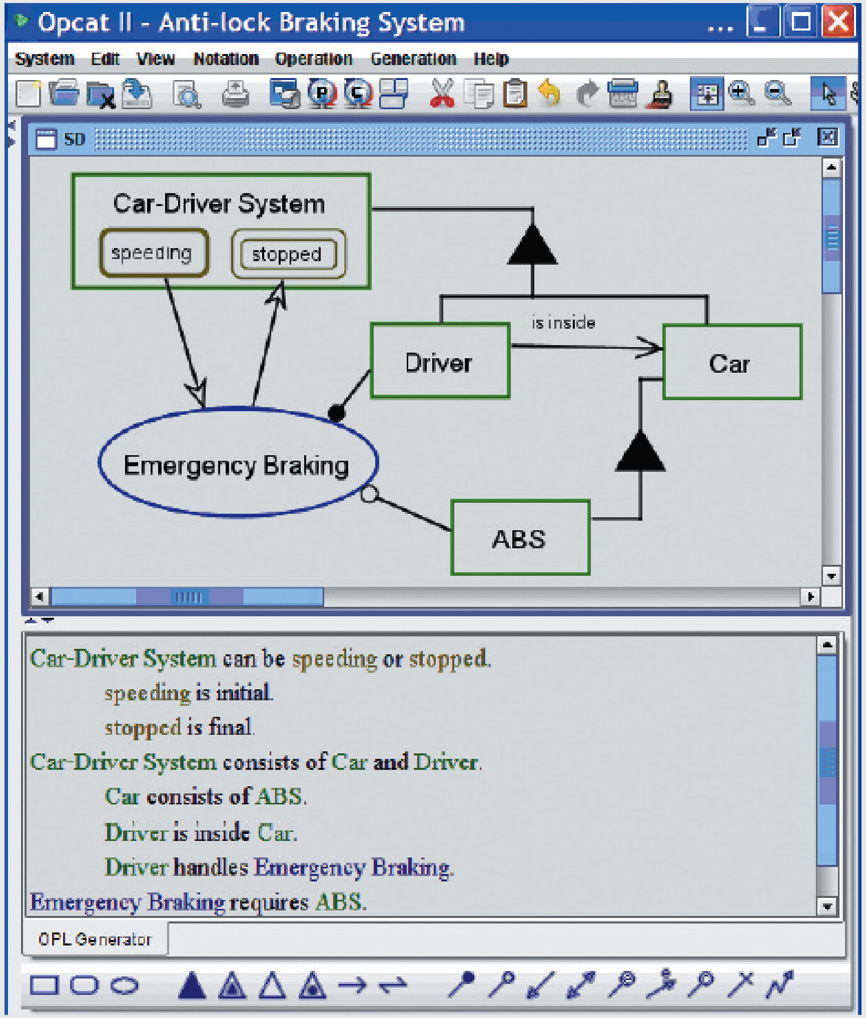
supplementation: **$PPxy \rightarrow \exists z\ (PPzy \wedge \neg Ozx)$**

principle of sum: **$\exists z\ \forall w\ (Owz \leftrightarrow (Owx \vee Owy\ ))$**

extensionality: **$x = y \leftrightarrow \forall w(Pwx \leftrightarrow Pwy)$**

Excluded models:

A Violation of Supplementation Axiom



Dov Dory, Words from pictures for dual-channel processing, *Communications of the ACM* 51, 2008

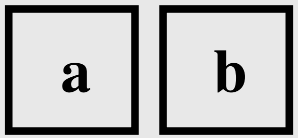
Sets and mereological sums

(note to the last lecture)

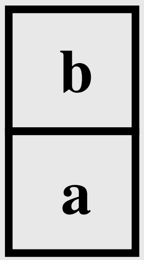
- Sets of concrete things are abstract
- Sums of concrete things are concrete!
- No analogous of membership relation and empty set for mereological sums

Part, Constitution, and Identity

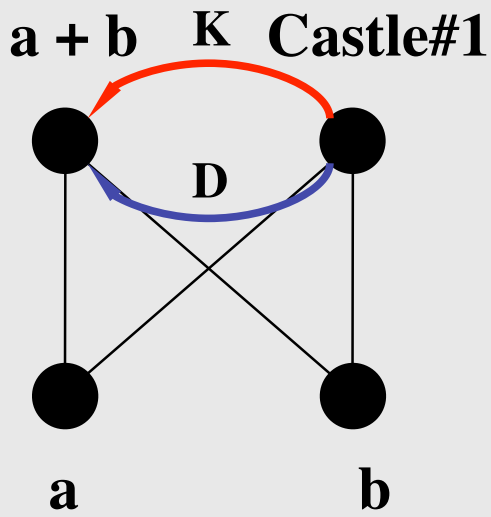
- **Parts not enough** to make the whole: structure creates a new entity
- **Mereological extensionality** is lost
- **Constitution** links the two entities
- **Constitution is** asymmetric (implies **dependence**)



Two blocks



A castle



Essence and Identity

Essential properties

- For an individual
  - John must have a brain
  - John must be a human
  - John must be alive
- For a type
  - All human beings must have a brain
  - All human beings must be “a whole” (all of a piece)

Essential properties and rigidity

- Certain entities must have some properties in order to *exist*
  - John must have a brain
  - John must be a person.
- Certain properties are essential to ***all*** their instances (***being a person*** vs. ***being hard***).
- These properties are ***rigid*** - Their extension is the same in all possible worlds. If an entity is ever an instance of a rigid property, it must necessarily be such.

- By the way, what's the meaning of *exist*?
  - Being an element of the domain of discourse
  - Being present *at a certain time* (or in a certain world...)

Formal Rigidity

- ϕ is rigid (+R): $\forall x\ (\Diamond\phi(x) \rightarrow \Box\phi(x))$
  - e.g. Person, Apple
- ϕ is non-rigid (-R): $\exists\ x\ (\Diamond\phi(x) \wedge \neg\ \Box\phi(x))$
  - e.g. Red, Male
- ϕ is anti-rigid (~R): $\forall\ x\ (\Diamond\phi(x) \rightarrow \neg\ \Box\phi(x))$ e.g. Student, Agent

Meta-properties

Formal rigidity - variations

- **Takint actual existence into account:**

$$\Box\forall x(\ \phi(x) \rightarrow \Box(E(x) \rightarrow \phi(x))\ )$$

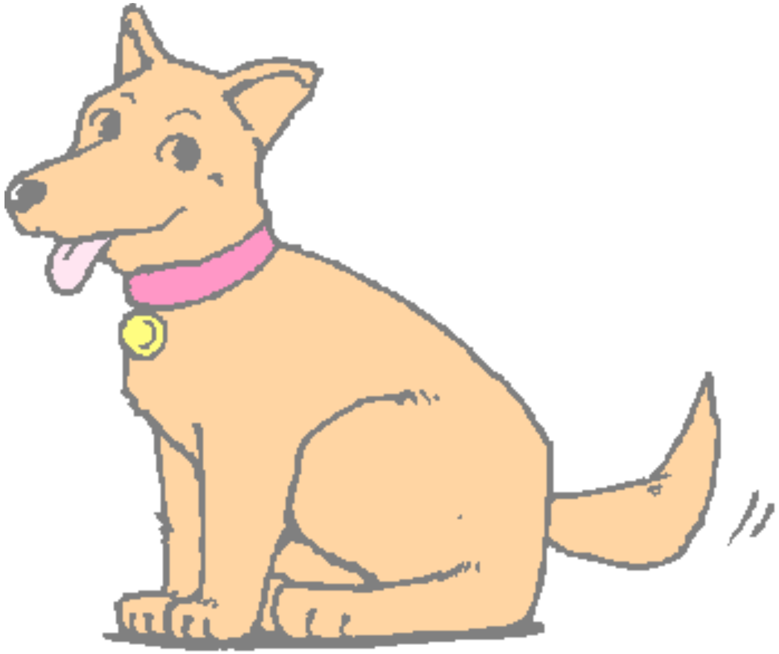
- **Taking time and actual existence into account:**

$$\Box\forall xt(\ (E(x,t) \wedge\ \phi(x,t)) \rightarrow \Box\forall t'(E(x,t') \rightarrow \phi(x)))$$

- *Welty, C. and Andersen, W. Towards OntoClean 2.0: A framework for rigidity (Applied Ontology 1(1), 2006)*

Unity and Essence

- **Unity**: is the collar part of my dog?
  - ***Being a whole*** is often a (very relevant) essential property
  - Dogs are ***essential wholes***...



Defining unity

- A tentative formulation: *x* ***is a whole*** under a unifying relation U iff U is an *equivalence relation* that binds together all the parts of *x*, such that, ***necessarily***,

$$P(y,x) \rightarrow (P(z,x) \leftrightarrow U(y,z))$$

but not

$$U(y,z) \leftrightarrow \exists x(P(y,x) \wedge P(z,x))$$

- P is the ***part-of*** relation
- U can be seen as a ***generalized indirect connection***

Unity Refined

Problem: the unity relation may not link together all the parts (think of a family as a whole)

$\delta_U(x) =_{\mathbf{df}} U(x, x)$ ***(x belongs to the domain of U)***

$\mathbf{U}_U(x) =_{\mathbf{df}} \Sigma_{\delta_U}(x) \wedge \forall y,z((\delta_U(y) \wedge \delta_U(z) \wedge P(y, x) \wedge P(z, x)) \rightarrow U(y, z))$

(x is unified by U)

$\mathbf{W}_U(x) =_{\mathbf{df}} Max_{\mathbf{U}_U}(x)$ ***(x is a whole under U)***

$\Sigma_\phi(x) =_{\mathbf{df}} \forall y(P(y, x) \rightarrow \exists z(\phi(z) \wedge P(z, x) \wedge O(z, y))$ ***(sum of φs)***

Kinds of Whole

- Depending on the **nature of the *unifying relation***, we can distinguish:
  - ***Topological wholes*** (a piece of coal, a heap of coal)
  - ***Morphological wholes*** (a constellation)
  - ***Functional wholes*** (a hammer, a bikini)
  - ***Social wholes*** (a population)

*   a whole can have ***parts that are themselves wholes*** (with a different unifying relation)

Unity and Plurality

- *Ordinary objects:* ***wholes or sums of wholes***
  - *Singular*: ***no wholes as proper parts***
  - *Plural*: ***sums of wholes***
    - *Plural wholes* (the sum is ***also a whole***)
    - *Collections* (the sum is not a whole)

A note on pluralities: Instances vs. members

- Often we use the same names for classes and their *characteristic properties*
- John is a member of "Person" ↔ Person(John)
- Tree#1 is a member of "TheBlackForest" ↔ TheBlackForest(Tree1) ??
    - violates usual intended interpretation of unary predicates: property shared by all instances of the corresponding class.
    - doesn't pass is-a test
- Membership is a relation between *individuals*

Identity criteria

- *Classic formulation:*

$$\phi(x) \wedge \phi(y) \rightarrow (\rho(x,y) \leftrightarrow x = y)$$

(ϕ *carries the identity criterion* ρ)

- *Generalization:*

$$\phi(x,t) \wedge \phi(y,t') \rightarrow (\Gamma(x,y,t,t') \leftrightarrow x = y)$$

(synchronic: $t = t'$; diachronic: $t \neq t'$)

- In most cases, Γ is based on the ***sameness*** of certain ***characteristic features****:*

$$\Gamma(x,y,t,t') = \forall z\, (\chi(x,z,t) \wedge \chi(y,z,t'))$$

- **Non-triviality condition:**
  - $\Gamma(x,y,t,t')$ must not contain an identity statement between x and y!

From identity *criteria* to weak identity *conditions*

- Finding necessary *and* sufficient ICs for a given property may be ***very hard***.
- In most cases, to apply the OntoClean methodology it is enough to detect whether a certain property *P* carries *supplementary membership conditions* (in addition to those logically implied by *P* itself)
- A property *P* carries an identity *condition C* if all its instances necessarily satisfy *C*, and *C* is not logically implied by *P*
- Typical example: ***having some essential parts or qualities***

Sortals and other properties

- **Sortals** (*horse, triangle, amount of matter, person, student...)*
  - Carry identity conditions
  - Usually correspond to ***nouns***
  - High organizational utility
- **Non-sortals** *(red, big, old, decomposable, dependent...)*
  - No identity
  - Usually correspond to ***adjectives***
  - Span across different sortals
  - Limited organizational utility (but high semantic value)

What about our rocks?

- *Igneous rock, metamorphic rock, sedimentary rock* do supply identity conditions.
- *Large rock, grey rock, pet rock DO NOT!*
- Not all properties are the same...

Carrying vs. Supplying Identity

- ***Supplying*** identity (**+O**)
  - Carrying an IC (or relevant essential property) that doesn't hold for *all* directly subsuming properties
- ***Carrying*** identity (**+I**)
  - Not supplying identity, while being subsumed by a property that does.
- ***Common sortal principle***: x=y -> there is a common sortal supplying their identity

- Theorem: only rigid properties supply identity

Identity, Countability, and Mass Nouns

- Nouns vs. adjectives
- Countability implies identity
- The problem with mass nouns: does the viceversa hold?
  - Being [an amount of] water:
    - Uncountable if arbitrarily divisible (but still carries identity!)
    - Countable if we assume molecules
      - We do have criteria for distinguishing and counting water molecules
      - We do have criteria for distinguishing and counting sums of water molecules
      - [compare with “being a group of people”]
  - Being made of water:
    - if x and y are made of water, nothing helps us to decide whether they are identical or not
  - So, “Being an amount of water” is a sortal,”Being made of water” is not.