Semantic models: ontologies

MSc 2008/2009

Lecture 9/10

Summary of the previous lecture

- OO modeling.
  - Object-oriented analysis.
  - Modeling primitives.
  - UML.
  - Class and object diagrams.
  - Relationships to ER modeling.

Today‘s lecture

- Semantic models: ontologies.

- Semantic Web and the role of ontologies.
- RDF.
- RDFS.
- OWL.
- Differences to other modeling paradigms.

Fundamentals of ontologies

What is an ontology?

STI · INNSBRUCK

- ***An ontology defines the basic terms and relations comprising the vocabulary of a topic area, as well as the rules for combining terms and relations to define extensions to the vocabulary***

Neches, R.; Fikes, R.; Finin, T.; Gruber, T.; Patil, R.; Senator, T.; Swartout, W.R. *Enabling Technology for Knowledge Sharing*. **AI Magazine**. Winter 1991. 36-56

- ***An ontology is an explicit specification of a conceptualization***

Gruber, T. *A translation Approach to portable ontology specifications.* **Knowledge Acquisition**. Vol. 5. 1993. 199-220

What is an ontology (cont)?

- *An ontology is a hierarchically structured set of terms for describing a domain that can be used as a skeletal foundation for a knowledge base*

B. Swartout; R. Patil; k. Knight; T. Russ. *Toward Distributed Use of Large-Scale Ontologies* **Ontological Engineering.** AAAI-97 Spring Symposium Series. 1997. 138-148

- *An ontology provides the means for describing explicitly the conceptualization behind the knowledge represented in a knowledge base*

A. Bernaras;I. Laresgoiti; J. Correra. *Building and Reusing Ontologies for Electrical Network Applications* **ECAI96. 12th European conference on Artificial Intelligence.** Ed. John Wiley & Sons, Ltd. 298-302

What is an ontology (cont)?

"An ontology is a formal, explicit specification of a shared conceptualization"

- formal → Machine-readable
- explicit specification → Concepts, properties relations, functions, constraints, axioms, are explicitly defined
- shared → Consensual Knowledge
- conceptualization → Abstract model and simplified view of some phenomenon in the world that we want to represent

Studer, Benjamins, Fensel. Knowledge Engineering: Principles and Methods. *Data and Knowledge Engineering*. 25 (1998) 161-197

Examples

OpenCyc.org

OMWS

eCl@ss
CLASSIFICATION AND PRODUCT DESCRIPTION

OpenGALEN

W3C®

unspsc®

OWL

mount foafmore

WonderWeb

WordNet

creative commons

OBO
open biomedical ontologies

SUMO

FOAF

United States
National Library of Medicine
NLM
National Institutes of Health

Features of an ontology

- Modelled knowledge about a specific domain.
- Defines
  - A common vocabulary.
  - The meaning of terms.
  - How terms are interrelated.
- Consists of
  - Conceptualization and implementation.
- Contains
  - Ontological primitives.
- **An ontology is a shared, reusable model.**
- **An ontology is written in a formal language.**
  - **Machine-understandable, enables reasoning.**

Ontological primitives

- Concepts are organized in taxonomies
- Relations
  - $R: C_1 \times C_2 \times \ldots \times C_{n-1} \times C_n$

Subclass-of: Concept 1 x Concept2
Connected to: Component1 x Component2

- Functions
  - $F: C_1 \times C_2 \times \ldots \times C_{n-1}$

Mother-of: Person --> Women
Price of a used car: Model x Year x Kilometers --> Price

- Instances
  - Elements
- Axioms
  - Sentences which are always true

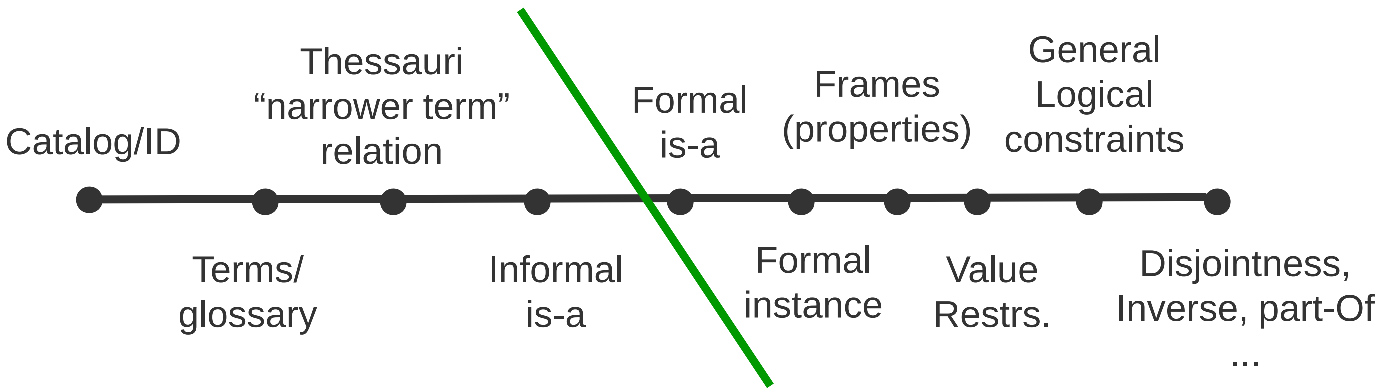
Classifications of ontologies (i)

STI · INNSBRUCK

- Issue of the conceptualization
  - Upper-level/Top-level
  - Core
  - Domain
  - Task
  - Application
  - Representation

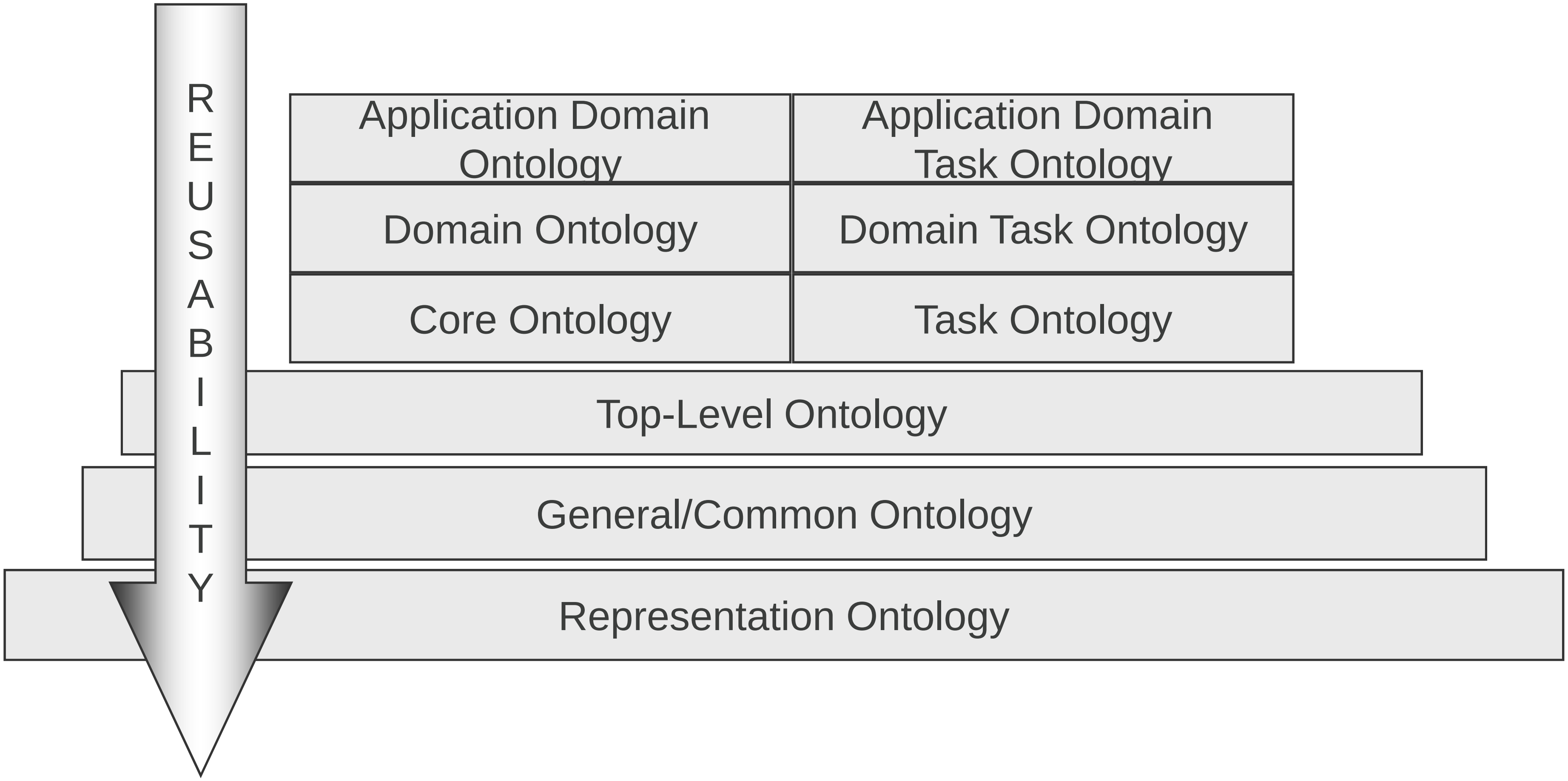
- Degree of formality
  - **Highly informal:** in natural language
  - **Semi-informal:** in a restricted and structured form of natural language
  - **Semi-formal:** in an artificial and formally defined language
  - **Rigorously formal:** in a language with formal semantics, theorems and proofs of such properties as soundness and completeness

Classifications of ontologies (ii)



Lassila O, McGuiness D. The Role of Frame-Based Representation on the Semantic Web. Technical Report. Knowledge Systems Laboratory. Stanford University. KSL-01-02. 2001.

Classifications of ontologies (iii)

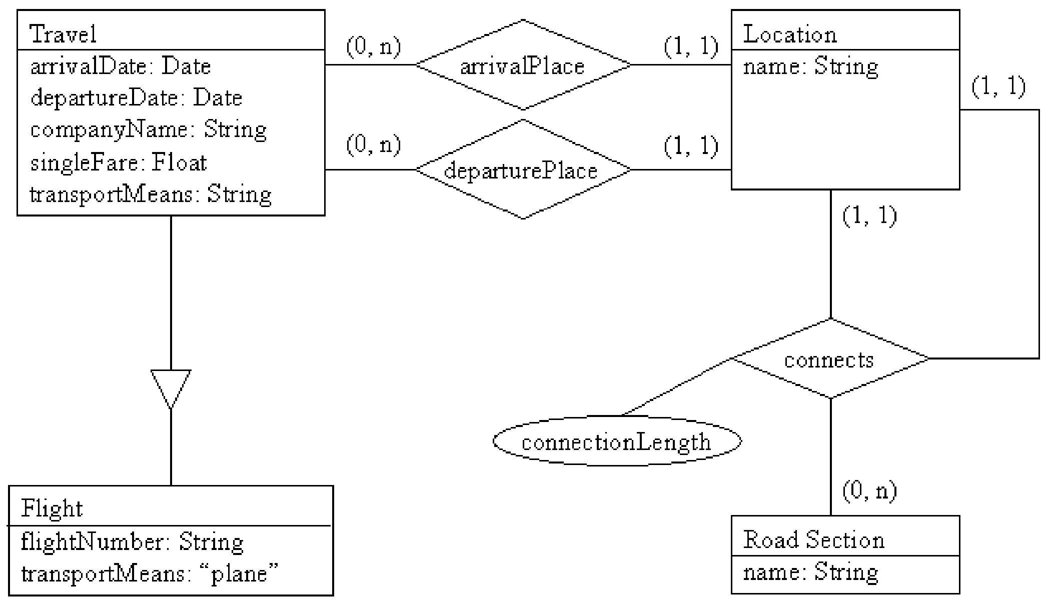


Languages for building ontologies

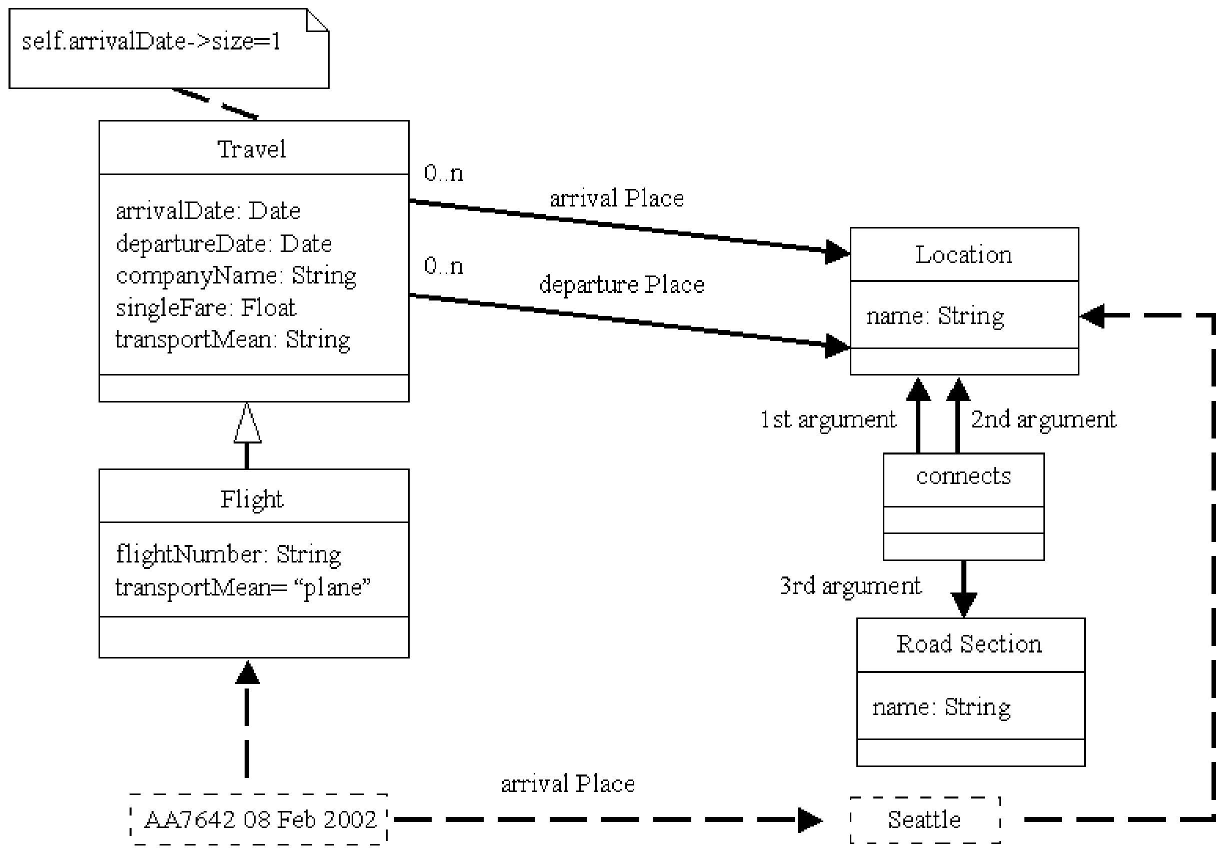
STI · INNSBRUCK

- Ontologies can be built using various languages with various degrees of formality
  - Natural language
  - UML
  - ER
  - OWL/RDFS
  - WSML
  - FOL
  - ...
- The formalism and the language limit the kind of knowledge that can be represented.
- A domain model is not necessarily a formal ontology only because it is written in OWL.
- Semantic Web ontologies are written in RDFS/OWL/WSML (more about this later).

Using the ER model for modeling ontologies



Using UML for modeling ontologies



Using Description Logics for modeling ontologies

```
(defconcept Travel
  "A journey from place to place"
:is-primitive
  (:and
    (:all arrivalDate Date)(:exactly 1 arrivalDate)
    (:all departureDate Date)(:exactly 1
departureDate)
    (:all companyName String)
    (:all singleFare Number)(:at-most singleFare 1)))
```

```
(tellm (AA7462 AA7462-08-Feb-2002)
       (singleFare AA7462-08-Feb-2002 300)
       (departureDate AA7462-08-Feb-2002 Feb8-2002)
       (arrivalPlace AA7462-08-Feb-2002 Seattle))
```

```
(defrelation Pays
:is
 (:function (?room ?Discount)
  (- (Price ?room) (/(*(Price ?room) ?Discount) 100)))
:domains (Room Number)
:range Number)
```

```
(defrelation connects
  "A road connects two different cities"
:arity 3
:domains (Location Location)
:range RoadSection
:predicate
  ((?city1 ?city2 ?road)
   (:not (part-of ?city1 ?city2))
   (:not (part-of ?city2 ?city1))
   (:or (:and (start ?road ?city1)(end ?road ?city2))
        (:and (start ?road ?city2)(end ?road ?city1)))))
```

Applications of ontologies

- Knowledge representation.
  - Ontology models domain knowledge
- Semantic annotation.
  - Ontology is used as a vocabulary, classification or indexing schema for a collection of items
- Semantic search.
  - Ontology is used as a query vocabulary or for query rewriting purposes
- Configuration.
  - Ontology defines correct configuration templates

Principles for the design of ontologies

STI · INNSBRUCK

- Clarity:
  - To communicate the intended meaning of defined terms
- Coherence:
  - To sanction inferences that are consistent with definitions
- Extendibility:
  - To anticipate the use of the shared vocabulary
- Minimal Encoding Bias:
  - To be independent of the symbolic level
- Minimal Ontological Commitments:
  - To make as few claims as possible about the world

Gruber, T.: *Towards Principles for the Design of Ontologies*. KSL-93-04. Knowledge Systems Laboratory. Stanford University. 1993

Clarity

STI · INNSBRUCK

An ontology should communicate effectively the intended meaning of defined terms. Definitions should be objective. Definitions can be stated on formal axioms, and a complete definition (defined by necessary and sufficient conditions) is preferred over a partial definition (defined by only necessary or sufficient conditions)…

```
(define-class Travel (?travel)
   "A journey from place to place"
:axiom-def
 (and (Superclass-Of Travel Flight)
      (Subclass-Of Travel Thing)
      (Template-Facet-Value Cardinality
            arrivalDate Travel 1)
      (Template-Facet-Value Cardinality
            departureDate Travel 1)
      (Template-Facet-Value Maximum-Cardinality
            singleFare Travel 1))
:def
  (and (arrivalDate ?travel Date)
       (departureDate ?travel Date)
       (singleFare ?travel Number)
       (companyName ?travel String)))
```

No clarity

Clarity

```
(define-class Travel (?travel)
   "A journey from place to place"
:axiom-def
 (and (Superclass-Of Travel Flight)
      (Subclass-Of Travel Thing)
      (Template-Facet-Value Cardinality
            arrivalDate Travel 1)
      (Template-Facet-Value Cardinality
            departureDate Travel 1)
      (Template-Facet-Value Maximum-Cardinality
            singleFare Travel 1))
:iff-def
  (and (arrivalDate ?travel Date)
       (departureDate ?travel Date))
:def
  (and (singleFare ?travel Number)
       (companyName ?travel String)))
```

Clarity → :iff-def

Minimal encoding bias

The conceptualization should be specified at the knowledge level without depending on a particular symbol-level encoding.

```
(define-class Travel (?travel)
   "A journey from place to place"
:axiom-def
 (and (Superclass-Of Travel Flight)
      (Subclass-Of Travel Thing)
      (Template-Facet-Value Cardinality
            arrivalDate Travel 1)
      (Template-Facet-Value Cardinality
            departureDate Travel 1)
      (Template-Facet-Value Maximum-Cardinality
            singleFare Travel 1))
:iff-def
  (and (arrivalDate ?travel Date)
       (departureDate ?travel Date))
:def
  (and (singleFare ?travel Number)
       (companyName ?travel String)))
```

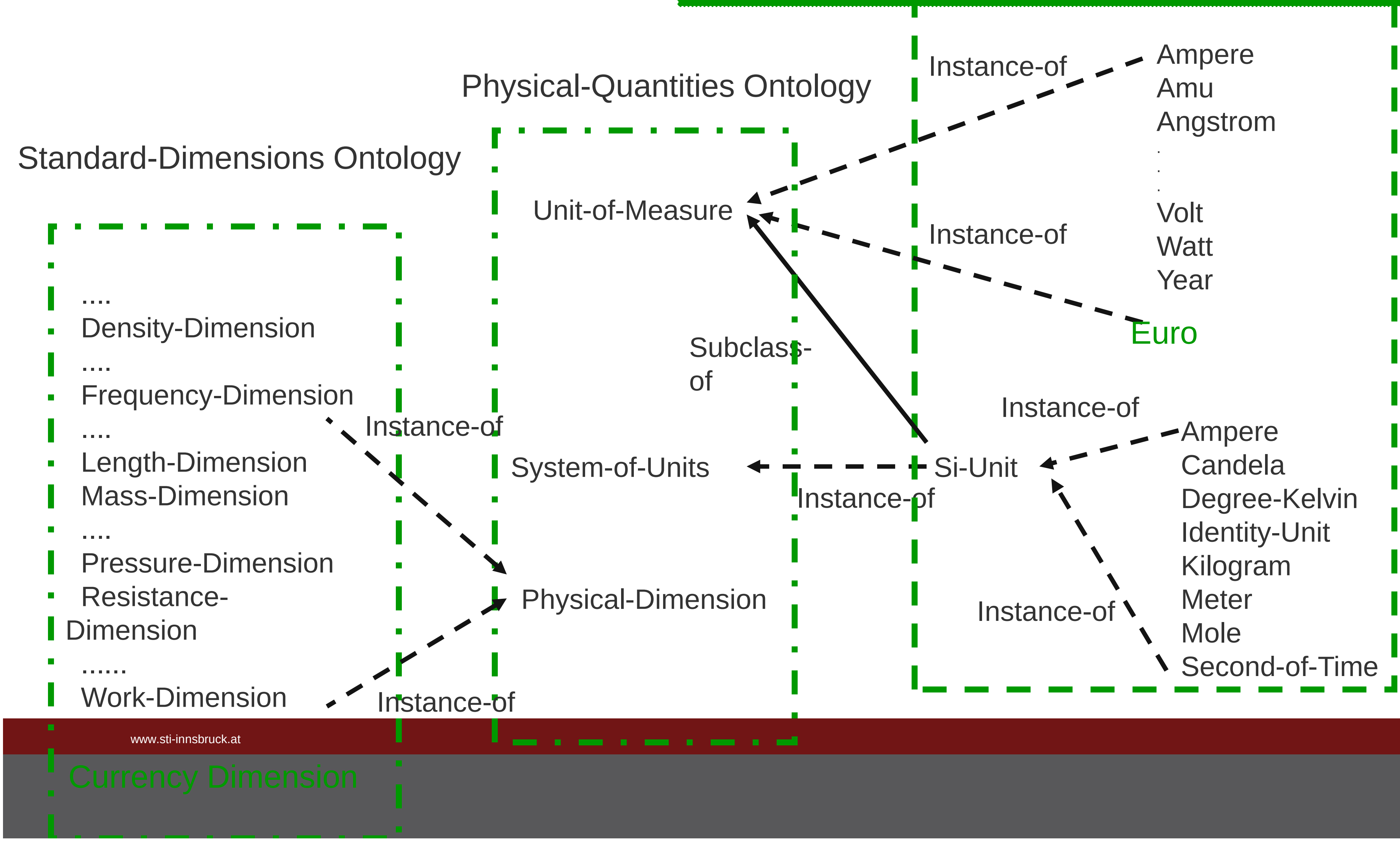
No minimal encoding bias

Minimal Encoding Bias

```
(singleFare ?travel Number)
```

should be substituted by:

```
(singleFare ?travel CurrencyQuantity)
```



Extensibility

> *One should be able to define new terms*
> *for special uses based on the existing vocabulary,*
> *in a way that does not require the revision of the existing definitions.*

- Currency dimension
- Definition of currencies
- Relationship between currencies

```
(define-individual Euro (Unit-of-Measure)
   "An Euro is the currency on the European Union"
:= (* 0,96 USDollar)
:axiom-def
  (= (Quantity.dimension Euro) CurrencyDimension))
```

Coherence

An ontology should be coherent: that is, it should sanction inferences that are consistent with the definitions.[…]
If a sentence that can be inferred from the axioms contradicts a definition or example given informally, then the ontology is incoherent.

```
(define-axiom No-Train-between-USA-and-Europe
   "It is not possible to travel by train between the USA and Europe"
:= (forall (?travel)
    (forall (?city1)
     (forall (?city2)
      (=> (and (Travel ?travel)
               (arrivalPlace ?travel ?city1)
               (departurePlace ?travel ?city2)
               (or (and (EuropeanLocation ?city1)
                        (USALocation ?city2))
                   (and (EuropeanLocation ?city2)
                        (USALocation ?city1) )))
          (not (TrainTravel ?travel)))))))
(define-instance Madrid (EuropeanLocation))
(define-instance NewYork (USALocation))
```

Minimal ontological commitments

Since ontological commitment is based on the consistent use of the vocabulary, ontological commitment can be minimized by specifying the weakest theory and defining only those terms that are essential to the communication of knowledge consistent with the theory.

```
(define-class Travel (?travel)
   "A journey from place to place"
:axiom-def
 ( .... )
:iff-def
  (and (arrivalDate ?travel Date)
       (departureDate ?travel Date))
:def
  (and (singleFare ?travel Number)
       (companyName ?travel String)))
```

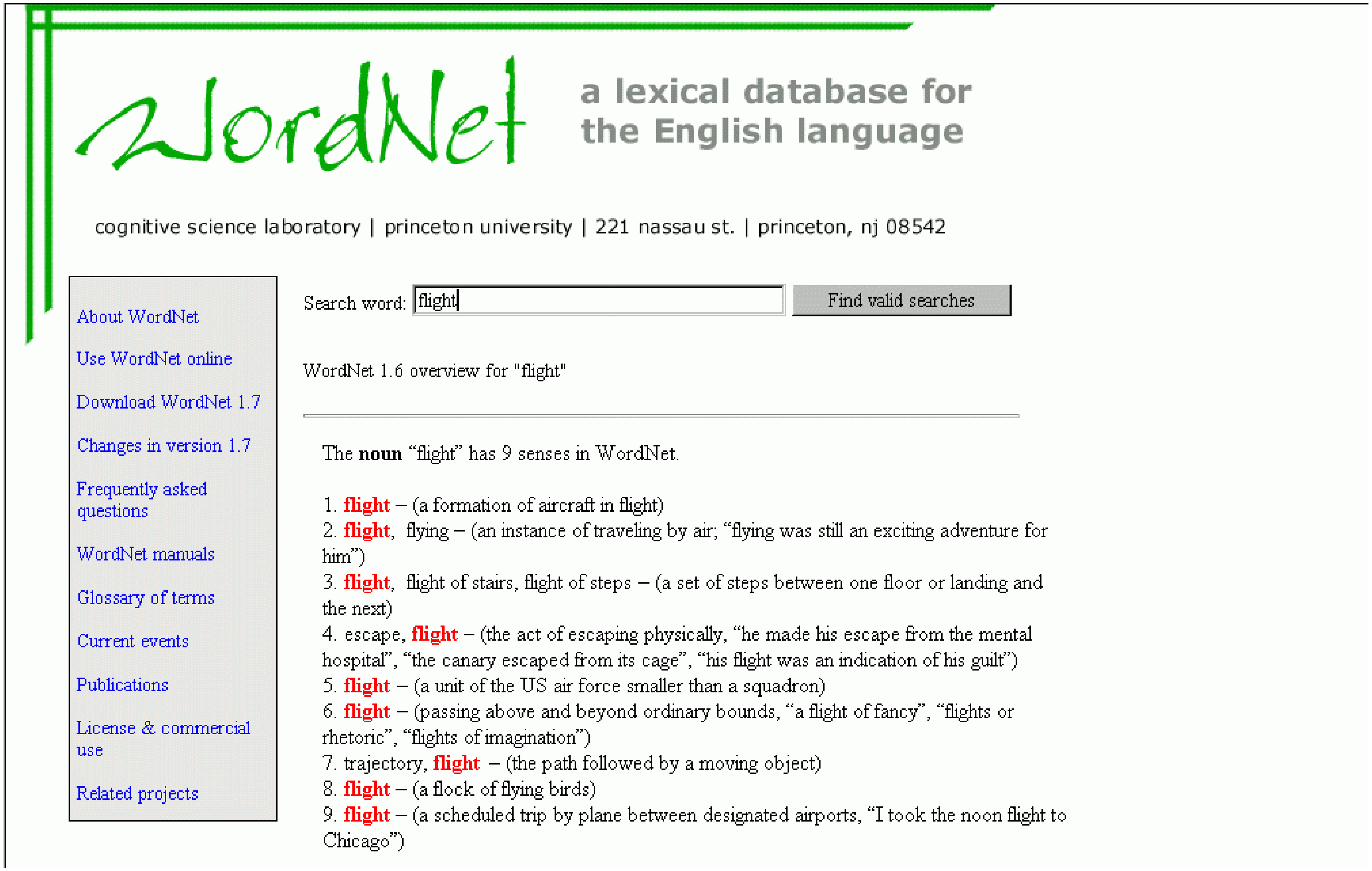
- Absolute/relative date?
- Format?

Ontological commitments

STI · INNSBRUCK

- Agreements to use the vocabulary in a coherent and consistent manner (Gruber).
  - An agent commits (conforms) to an ontology if it “acts” consistently with the definitions
- Connection between the ontology vocabulary and the meaning of the terms of such vocabulary.

Ontological commitments (cont)



Steps to build an ontology

Natalya F. Noy and Deborah L. McGuinness. ``Ontology Development 101: A Guide to Creating Your First Ontology''. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.

Step 1: Determine the domain and scope of the ontology

STI · INNSBRUCK

- What is the domain that the ontology will cover?
- For what we are going to use the ontology?
- For what types of questions the information in the ontology should provide answers?
- Who will use and maintain the ontology?

Competency Questions

- A set of queries which place demands on the underlying ontology.
- Ontology must be able to represent the questions using its terminology and the answers based on the axioms
- Ideally, in a staged manner, where consequent questions require the input from the preceeding ones.
- A rationale for each competency question should be given.

Step 2: Consider reusing existing ontologies

- Reuse ensures interoperability and reduces costs
- Ontology libraries and tools for customization are required for this step
- Sub-steps
  - Discover potential reuse candidates
  - Evaluate their usability
  - Customize ontologies to be reused
  - Integrate and merge to the target ontology

Step 3: Enumerate important terms in the ontology

- What are the terms we would like to talk about?
- What properties do those terms have?
- What would we like to say about those terms?

Step 4: Define classes and class hierarchy

- A top-down development process starts with the definition of the most general concepts in the domain and subsequent specialization of the concepts.
- A bottom-up development process starts with the definition of the most specific classes, the leaves of the hierarchy, with subsequent grouping of these classes into more general concepts.
- Middle-out approach: define the more salient concepts first and then generalize and specialize them appropriately.

Step 4: Define classes and class hierarchy (cont)

- From the list created in Step 3, select the terms that describe objects having independent existence rather than terms that describe these objects.
  - These terms will be classes in the ontology.
- Organize the classes into a hierarchical taxonomy by asking if by being an instance of one class, the object will necessarily (i.e., by definition) be an instance of some other class.
  - *If a class A is a superclass of class B, then every instance of B is also an instance of A.*
- Classes as unary predicates—questions that have one argument. For example, “Is this object a wine?”
  - Later: binary predicates (or slots)—questions that have two arguments. For example, “Is the flavor of this object strong?” “What is the flavor of this object?”

How to find classes

See Lectures 3/4

- Typical candidates for classes: NOUNS.
  - But: actors of use cases do not necessarily correspond to classes.
- Other terms as well
  - Gerund: „My eyes glazing over…“ ~ withdrawal
  - Verbs: an association which starts to take on attributes and associations of its own turns into an entity: „Officer arrests suspect“.
  - Verbs: events: „Illness episode“.
  - Passive form: re-formulate in active form.
  - No pronouns.

Cohesion and identity

See Lectures 3/4

- A class should represent one thing, all of that thing and nothing but that thing.

- You can prove cohesion by
  - Giving the class a representative name.
  - Noun (+ adjective, sometimes however also captured as attribute value).
    - Blackmail victim, robbery victim.
    - Blue car, red car.
    - **Cars** is not cohesive.

- Avoid ambiguous terms.
  - Manager, handler, processor, list, information, item, data…

- Identity ~ individuality: entities change values, but are still the same entity
  - Child/Adult: age

Class hierarchy

- A subclass of a class represents a concept that is a "kind of" the concept that the superclass represents.
- Classes represent concepts in the domain and not the words that denote these concepts. Synonyms for the same concept do not represent different classes.
- All the siblings in the hierarchy (except for the ones at the root) must be at the same level of generality.
- If a class has only one direct subclass there may be a modeling problem or the ontology is not complete.
- If there are more than a dozen subclasses for a given class then additional intermediate categories may be necessary.
- Subclasses of a class usually (1) have additional properties that the superclass does not have, or (2) restrictions different from those of the superclass, or (3) participate in different relationships than the superclasses.
- If the concepts with different slot values become restrictions for different slots in other classes, then we should create a new class for the distinction. Otherwise, we represent the distinction in a slot value.

Step 5: Define attributes and relationships

- Step 4 selected classes from the list of terms we created in Step 3.
  - Most of the remaining terms are likely to be properties of these classes.
  - For each property in the list, we must determine which class it describes.
- Types of properties
  - “intrinsic” properties
  - “extrinsic” properties
  - parts, if the object is structured (physical or abstract).
  - relationships to other individuals.
- Properties are inherited and should be attached to the most general class in the hierarchy.

Characterizing classes

See Lectures 3/4

- Two types of principal characteristics:
  - Measurable properties: attributes.
  - Inter-class connections: relationships.

  - Use relationships to capture something with an identity.

  - Arrest details as attribute of the suspect vs. arrest as an relationship.
    - Do we measure degrees of arrestedness or do we want to be able to distinguish between arrests?

  - Color of an image as attribute vs. class.

  - A „pointing finger“ rather than a „ruler“ indicates identity.

How to find attributes

STI · INNSBRUCK

See Lectures 3/4

- Nouns in „-ness“
  - Velocity-ness, job-ness, arrested-ness…
- „How much, how many“ test.
  - If you evaluate this, then it is probably an attribute.
  - If you enumerate these, it is probably an entity.

Relationships

See Lectures 3/4

- Are defined on sets of instances.
- Properties: reflexivity, cardinality, functional, inverse-functional, discountinuous multiplicity, many-to-many, all values from, some values of, transitivity, symmetry etc.
- Arity.

How to find relationships

See Lectures 3/4

- Verbs, verbal phrases and things that could have been verbs.
  - „*The buttler murdered the duchess*“

Step 6: Define the restrictions of the properties

- Refine the semantics of the properties
  - Cardinality
  - Domain and range
    - When defining a domain or a range for a slot, find the most general classes or class that can be respectively the domain or the range for the slots .
    - Do not define a domain and range that is overly general

Step 7: Create instances

- Define an individual instance of a class requires
  - choose a class
  - create an individual instance of that class
  - filling in the values of the properties

Some modeling guidelines

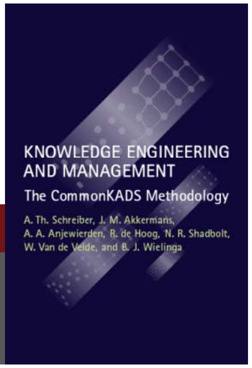
General issues

- There is no one correct way to model a domain— there are always viable alternatives. The best solution almost always depends on the application that you have in mind and the extensions that you anticipate.
- Ontology development is necessarily an iterative process.
- Classes in the ontology should be close to objects (physical or logical) and relationships in your domain of interest. These are most likely to be nouns (objects) or verbs (relationships) in sentences that describe your domain.

Domain analysis

- Ontology engineer needs to find a balance between learning about the domain without becoming a full domain expert.

- Talk to people in the organization who have to talk to experts, but are not experts themselves.
- Avoid diving into detailed, complicated theories unless their usefullness is proven.
- Construct a few typical scenarios which you understand at a global level.



Purpose and scope of the ontology

STI · INNSBRUCK

- Be clear about why the ontology is being built and what its intended usages are
  - Interoperability between systems
  - Systems engineering
  - Semantic search, semantic annotation
  - Communication between people and organizations
- Example: semantic search
  - Semi-formal ontology
  - Usage of natural language labels and naming conventions
  - Well-balanced at schema and instance level
  - Rich conceptualization
  - Syntactical and semantic correcteness
- The ontology should not contain all the possible information about the domain.

Degree of formality

- The more formal an ontology is, the more does it exclude unwanted interpretations.
- Also, it includes the amount of inferences that can be drawn.
- However, it is more costly in terms of labor and engineering delay to create a more formal conceptualization.
- Also, achieving concensus is more difficult and time-consuming for a higher degree of formality.
- A higher degree of formality also imposes high entry barriers on the ones who are to model the respective domains and thus excludes potential contributors.

Web 3.0/Semantic Web

See

http://www.slideshare.net/freekbijl/web-30-explained-with-a-stamp-pt-ii?nocache=8695

Summary

- What are ontologies and how can they be built?
- Differences to other modeling primitives.
- Ontologies and the Semantic Web/Web3.0

Assignments 9/10

STI · INNSBRUCK

- Imagine an online portal for sharing cooking recipes. Examples of such portals are http://www.rezepte-ideen.de/, http://www.cuisine.at/ and many others. Develop an ontology for this domain following steps1-7 presented in the lecture.

- Download and install the Protégé tool (4.x) from http://protege.stanford.edu/download/registered.html#p4 and create an OWL version of your ontology. Information about using Protégé is available at http://protegewiki.stanford.edu/index.php/Protege4GettingStarted, http://www.co-ode.org/resources/tutorials/protege-owl-tutorial.php.

Thank You!

Questions?