TDT44 – Semantic Web

Måshgan Tævækålifærd
Mozhgan@idi.ntnu.no

Course Structure

- Self study
  - Textbook: A Developer's Guide to the Semantic Web. Liyang Yu.
- Four lectures by students covering the chapters of the book + two session for assignment discussion
- Only one mandatory assignment ☺
- Oral exam in December 1-5 (TBA)
- Webpage: http://www.idi.ntnu.no/emner/tdt44/

Agenda

- Introduction to the Semantic Web
- A brief introduction to set theory
- A brief introduction to Logic

Introduction to the Semantic Web

What is the Semantic Web?

- Chapter1 → a nice history of the semantic web
  It's history goes back to my favorite philosophers Aristotle and the colleagues

- The Semantic Web
  - Open Standards for describing information on the Web

  and
  - Methods for obtaining further information from such descriptions

- Application areas
  - Search engines
  - Browsing online stores (B2C)
  - Service description and integration (B2B)
  - E-learning

Why do we need it??!!!

- The problem
  - ***Information overload and knowledge representation***
    - too much information with too little structure
  - Content/knowledge can be accessed only by humans, not by machines and meaning (semantics) of transferred data is not accessible
- Need
  - To add semantic to the web of data
- Motivation
  - To get computers to do more of the hard work, i.e., linking and interpretation of data

Example (Search engines scenario)

- Problems with current search engines
  - Current search engines = keywords:
    - high recall, low precision
    - sensitive to vocabulary
    - insensitive to implicit content
  - Search engines on the Semantic Web
    - concept search instead of keyword search
    - semantic narrowing/widening of queries
    - query-answering over more than one document
    - document transformation operators

Recherche avancée Préférences Outils linguistiques Conseils de recherche

thé [Recherche Google]

Rechercher dans : ⊙ Web ○ Pages francophones ○ Pages : France

Conseil – Avec la plupart des navigateurs, vous pouvez lancer la recherche en appuyant sur la touche Entrée (au lieu de cliquer sur le bouton Recherche Google).

Web Images Groupes Répertoire

Google a recherché **thé** sur le Web. **1** - **10** résultats, sur un total d'environ **424,000**. Recherche effectuée en **0.17** sec

Adobe Acrobat Reader - Download - [Traduire cette page]
... documents to Adobe PDF files for viewing with Acrobat Reader for mobile devices. See **the** chart below for a comparison of **the** free Acrobat Reader and **the** full ...
www.adobe.com/prodindex/acrobat/readstep.html - 10k - 26 Mar 2002 - Copie cachée - Pages similaires

Adobe Acrobat Reader - Download - [Traduire cette page]
... With Acrobat Reader, you can also fill in and submit Adobe PDF forms online. And now Acrobat Reader for mobile devices lets you view Adobe PDF files on **the** go. ...
www.adobe.com/products/acrobat/readstep2.html - 43k - 26 Mar 2002 - Copie cachée - Pages similaires
[Autres résultats, domaine www.adobe.com]

Welcome to **the** White House - [Traduire cette page]
... Today at **the** White House, Mar. 26, 2002. President Recognizes Greek Independence Day President ...
Description: Official site. Features a virtual historical tour, history of American presidents and their families,...
Catégorie: Regional > North America > ... > Executive Office of the President > White House

Problem: the current Web does not make a distinction between French *thé* and the English definite article…

Google™ Recherche avancée Préférences Outils linguistiques Conseils de recherche

thé [Recherche Google]

Rechercher dans : ○ Web ◉ Pages francophones ○ Pages : France

Conseil – Avec la plupart des navigateurs, vous pouvez lancer la recherche en appuyant sur la touche Entrée (au lieu de cliquer sur le bouton Recherche Google).

Web Images Groupes Répertoire

Google a recherché **thé** dans les pages en langue **Français**. **1** - **10** résultats, sur un total d'environ **280,000**. Recherc

Site officiel du musée du Louvre
Site officiel du musée du Louvre.
Description: Official site. History, collections, virtual tour, schedule of all **the** events taking place at **the** museum.
Catégorie: Regional > Europe > ... > Ile-de-France > Paris > Arts and Entertainment > Museums
www.louvre.fr/ - 2k - 26 Mar 2002 - Copie cachée - Pages similaires

Louvre Museum Official Website
Louvre Museum Official Website.
Description: **The** official site of museum, with history, collections, and current exhibitions.
Catégorie: Regional > Europe > France > Arts and Entertainment > Museums > Louvre
www.louvre.fr/louvrea.htm - 2k - 26 Mar 2002 - Copie cachée - Pages similaires
[Autres résultats, domaine www.louvre.fr]

The European Commission
... de -, Willkommen bei der Europäischen Kommission ! el -, el. en -, Welcome to **the** European Commission ! ...
Description: Administrative institution implementing **the** policies, laws, and treaties of **the** European Union.
Catégorie: Society > Government > ... > Regional > European Union > European Commission
europa.eu.int/comm/ - 7k - 26 Mar 2002 - Copie cachée - Pages similaires

… even when you specify you want "French-speaking pages" only

We miss some *semantics* here…

Example (B2C scenario)

- Problems with online stores (B2C)
  - Manual browsing is time-consuming and inefficient
  - Every shopbot requires a series of wrappers
    - Work only partially
    - Extract only explicit information
    - Must be updated frequently

- B2C on the Semantic Web
  - Software agents "understand" product descriptions
    - enables automatic browsing
  - Procedural wrapper-coding becomes declarative ontology-mapping
    - improves robustness and simplifying maintenance

Example (e-learning scenario)

- Problems with E-learning on the web
  - Search problem for the material
  - Material is designed for “typical” students
    - No student is typical!!!
  - More adaptively is needed
    - There is some, e.g., links revealed once material has been covered
  - Student’s knowledge level is implicit

- E-learning on the Semantic Web
  - Students would be able to find suitable courses
  - Materials can be tailored for the individual
  - Materials can be re-used
  - Models can be made of the domain, learner profile, learning strategies, …
  - Student’s knowledge level can be make explicit
    - In terms of the domain model, learning strategy, …

The Web in Three Generations

- Hand-coded (HTML) Web Content
  - Easy access through uniform interface
  - Problems
    - Huge authoring and maintenance effort
    - Hard to deal with dynamically changing content

- Automated on-the fly content generation
  - Based on templates filled with database content
  - Later extended with XML document ("meaningful" tags) transformations
  - Problems:
    - Inflexible
    - Limited number of things can be expressed

- Automated processing of content
  - The Semantic Web
    - Any content may find its own place in a given ontology…
    - … So, you "just" need to link content to its relevant place in the relevant ontology(-ies)!

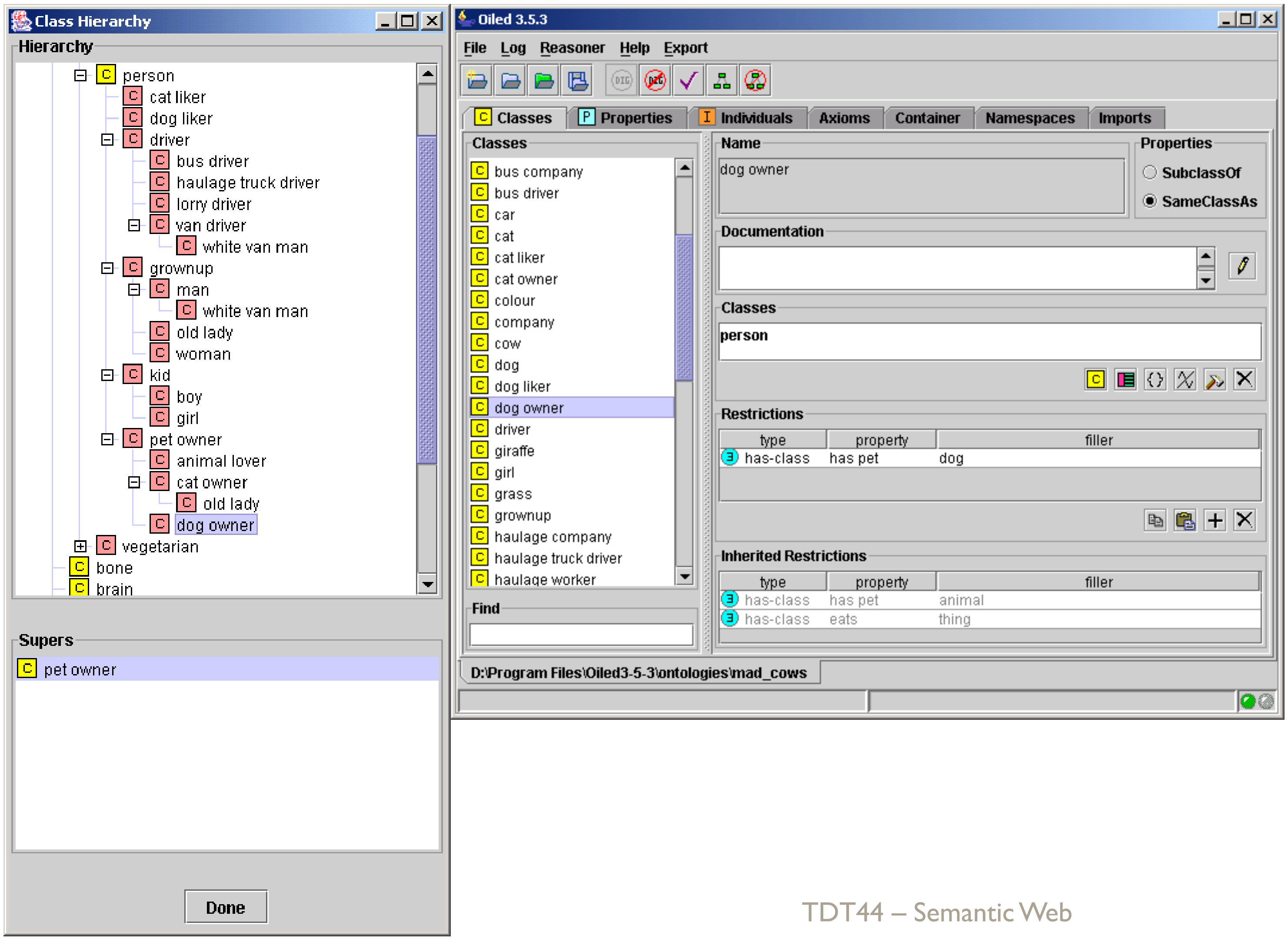
Ontologies

"An explicit specification of a conceptualisation" [Gruber93]

- An ontology is an engineering artifact:
  - Taxonomy
    - a specific vocabulary used to describe a certain reality → concepts
  - The background knowledge
    - a set of explicit assumptions regarding the intended meaning of the vocabulary
      - Almost always including how concepts should be classified
    - E.g.
      - Concepts:
        - Elephant is a concept whose members are a kind of animal
        - Adult_Elephant is a concept whose members are exactly those elephants whose age is greater than 20 years
      - Constraints
        - Adult_Elephants weigh at least 2,000 kg

- Thus, an ontology describes a formal specification of a certain domain:
  - Shared understanding of a domain of interest
  - Formal and machine manipulable model of a domain of interest

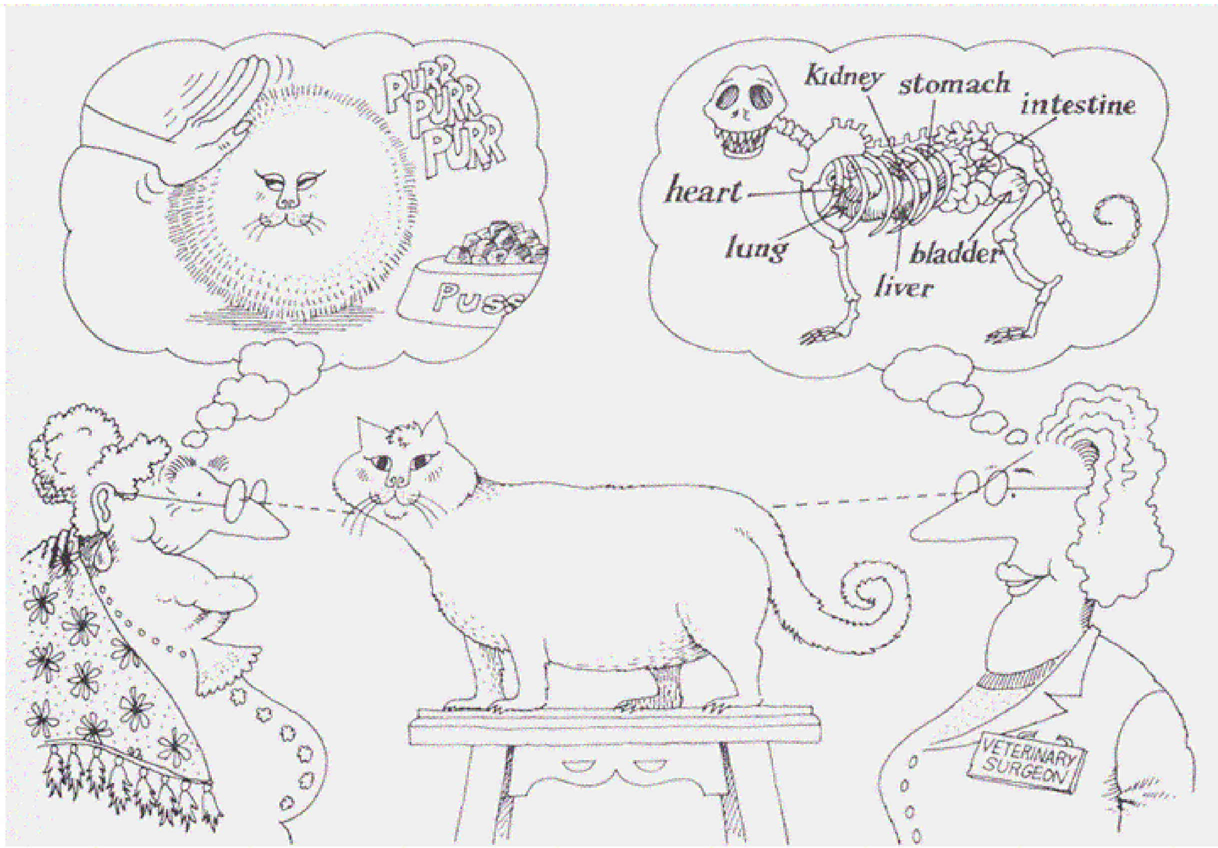
Example


Class Hierarchy
Hierarchy
person
cat liker
dog liker
driver
bus driver
haulage truck driver
lorry driver
van driver
white van man
grownup
man
white van man
old lady
woman
kid
boy
girl
pet owner
animal lover
cat owner
old lady
dog owner
vegetarian
bone
brain
Supers
pet owner
Done
Oiled 3.5.3
File Log Reasoner Help Export
Classes Properties Individuals Axioms Container Namespaces Imports
Classes
bus company
bus driver
car
cat
cat liker
cat owner
colour
company
cow
dog
dog liker
dog owner
driver
giraffe
girl
grass
grownup
haulage company
haulage truck driver
haulage worker
Find
Name
dog owner
Properties
SubclassOf
SameClassAs
Documentation
Classes
person
Restrictions
type property filler
has-class has pet dog
Inherited Restrictions
type property filler
has-class has pet animal
has-class eats thing
D:\Program Files\Oiled3-5-3\ontologies\mad_cows

What is the usefulness of an ontology?

- To make domain assumptions explicit
  - Ontological analysis
    - clarifies the structure of knowledge
    - allow domain knowledge to be explicitly defined and described
- Enrich software applications with the additional semantics
- To facilitate communications among systems with out semantic ambiguity. i,e to achieve inter-operability
- Thus, practically, improving: computer-computer, computer-human, and human-human communication
- To provide foundations to build other ontologies (reuse)
- To save time and effort in building similar knowledge systems (sharing)

World without ontology = Ambiguity



Ambiguity for humans

Cat

The Vet and Grandma associate different view for the concept cat.

Application Areas of Ontologies

- Information Retrieval
  - As a tool **for intelligent search** through inference mechanism instead of keyword matching
  - **Easy retrievability** of information without using complicated Boolean logic
  - **Cross Language Information Retrieval**
  - Improve **recall** by query expansion through the synonymy relations
  - Improve **precision** through Word Sense Disambiguation (identification of the relevant meaning of a word in a given context among all its possible meanings)
- Digital Libraries
  - Building **dynamical catalogues** from machine readable meta data
  - Automatic indexing and annotation of web pages or documents with meaning
  - To give context based organisation (semantic clustering) of information resources
  - Site organization and navigational support
- Information Integration
  - Seamless integration of information from different websites and databases
- Knowledge Engineering and Management
  - As a knowledge management tools for selective **semantic access** (meaning oriented access)
  - Guided discovery of knowledge
- Natural Language Processing
  - Better machine **translation**
  - Queries using natural language
- Artificial intelligence and intelligent agents

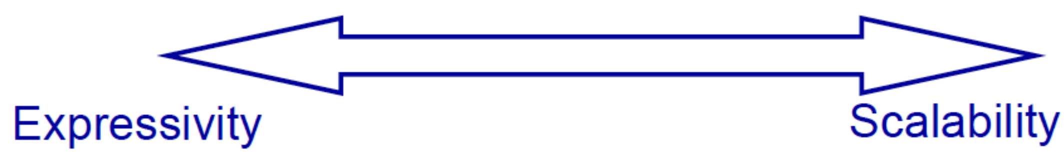
Tools and Services

- Design and maintain high quality ontologies, e.g.:
  - Meaningful — all named classes can have instances
  - Correct — captured intuitions of domain experts
  - Minimally redundant — no unintended synonyms
  - Richly axiomatised — (sufficiently) detailed descriptions
- Store (large numbers) of instances of ontology classes, e.g.:
  - Annotations from web pages
- Answer queries over ontology classes and instances, e.g.:
  - Find more general/specific classes
  - Retrieve annotations/pages matching a given description
- Integrate and align multiple ontologies

But be careful !!!!

- Ontologies are fancy, but don't prescribe it immediately, because

"Scalability is a challenge"

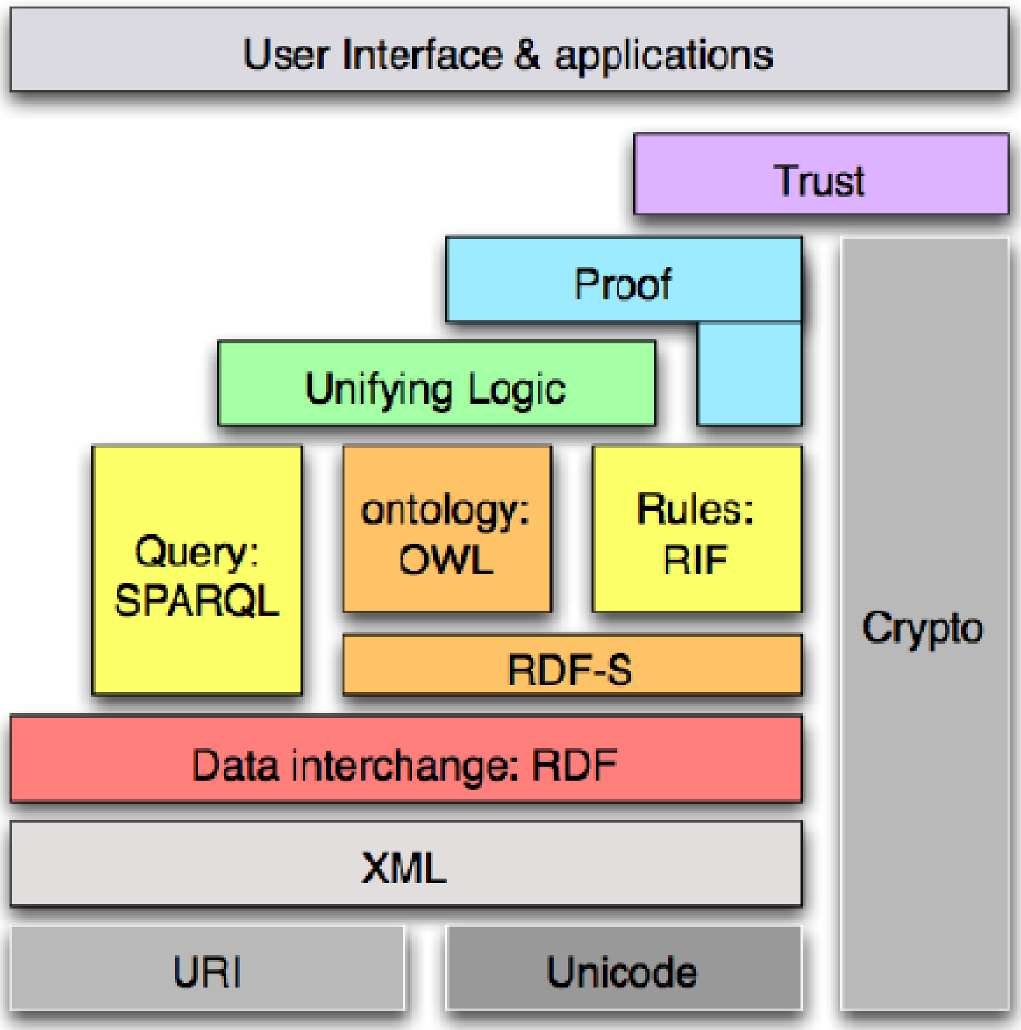


Still There are Challenges …

- The challenge:
  - Ontologies are tricky
    - People do it too easily; People are not logicians
      - Intuitions hard to formalise

"The challenge of the Semantic Web is to find a representation language powerful enough to support automated reasoning but simple enough to be usable" [AKT 2003]

Ontology Languages: the Wedding Cake …



HTML → XML

HTML:

```
<H1>Semantic Web</H1>
        <UL>    <LI>Teacher: Sætre
                <LI>Students: one, two, three
                <LI>Requirements: none
        </UL>
```

XML:

```
<?xml version="1.0"?>
<Course id="TDT44"
        xmlns="http://idi.ntnu.no/emner/tdt44">
            <title>Semantic Web</title>
            <teacher>Sætre</teacher>
            <students>one, two, three, …</students>
            <req>none</req>
</course>
```

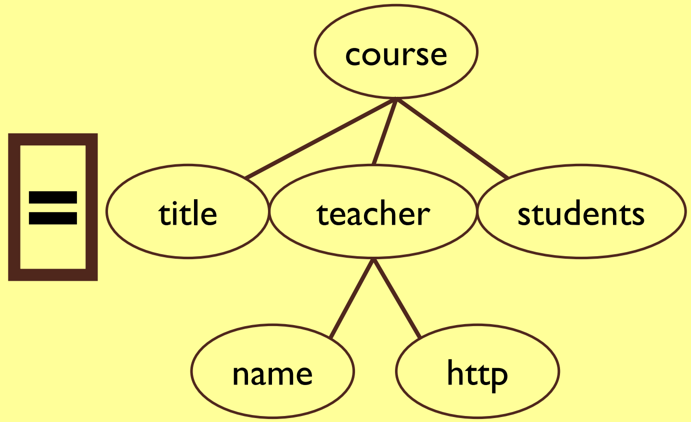
XML: document = labeled tree

- node = label + attr/values + contents

```
<course date="…">
        <title>…</title>
        <teacher>…</teacher>

        <name>…</name>

        <http>…</http>
        <students>…</student
s>
</course>
```



- XML Schema: grammars for describing legal trees and datatypes

- So:
  why XML is not good enough for the Semantic Web?

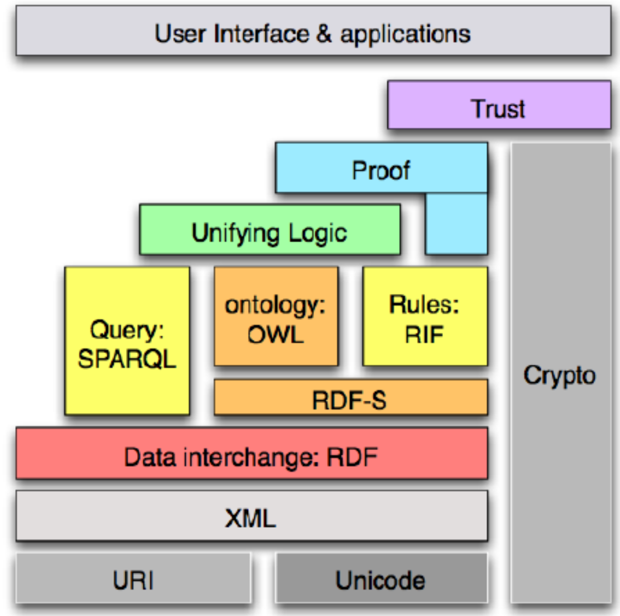
Syntax versus Semantics

- Syntax → the structure of your data
- Semantics → the meaning of your data
- Two conditions necessary for interoperability:
  - Adopt a common syntax: this enables applications to parse the data
  - Adopt a means for understanding the semantics: this enables applications to use the data

- XML makes no commitment on
  - Domain-specific ontological vocabulary
  - Ontological modeling primitives

- XML Requires pre-arranged agreement on these two
  - Only feasible for closed collaboration
    - agents in a small & stable community
    - pages on a small & stable intranet
  - Not suited for sharing Web-resources

Stack of languages



- XML
  - Surface syntax, no semantics
- XML Schema
  - Describes structure of XML documents
- RDF
  - Datamodel for "relations" between "things"
- RDF Schema
  - RDF Vocabulary Definition Language
- OWL
  - A more expressive Vocabulary Definition Language

RDF (Resource Description Framework)

- RDF is a standard way of specifying data "about" something

- RDF is a data model
  - domain-neutral, application-neutral and ready for internationalization
  - abstract, conceptual layer independent of XML
  - consequently, XML is a transfer syntax for RDF, not a component of RDF

- The details of RDF will be given in the next session, but why we should bother about the RDF?????

XML → RDF

Modify the following XML document so that it is also a valid RDF document:

XML:

TDT44.xml

```
<?xml version="1.0"?>
<Course id="TDT44"
        xmlns="http://idi.ntnu.no/emner/tdt44">
          <title>Semantic Web</title>
          <teacher>Sætre</teacher>
          <students>one, two, three, …</students>
          <req>none</req>
</course>
```

↓ "convert to"

TDT44.rdf

RDF:

```
<?xml version="1.0"?>
<Course rdf:ID="TDT44"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns="http://idi.ntnu.no/emner/tdt44#">
          <title>Semantic Web</title>
          <teacher>Sætre</teacher>
          <students>one, two, three, …</students>
          <req>none</req>
</course>
```

The RDF Format

① RDF provides an ID attribute for identifying the **resource** being described.

② The ID attribute is in the RDF namespace.

```
<?xml version="1.0"?>
<Course rdf:ID="TDT44"
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns="http://idi.ntnu.no/emner/tdt44#">
            <title>Semantic Web</title>
            <teacher>Sætre</teacher>
            <students>one, two, three, …</students>
            <req>none</req>
</course>
```

③ Add the "fragment identifier symbol" to the namespace.

Still why should I bother about the RDF????

- Answer: there are numerous benefits:
  - More interoperability
    - Tools can instantly characterize the structure, "this element is a type (class), and here are its properties"
    - RDF promotes the use of standardized vocabularies ... standardized types (classes) and standardized properties
  - A structured approach to designing XML documents (it is a regular, recurring pattern)
  - Better understand the data
    - quickly identification of weaknesses and inconsistencies of non-RDF-compliant XML designs
  - Benefits of both worlds:
    - You can use standard XML editors and validators to create, edit, and validate your XML
    - You can use the RDF tools to apply inferencing to the data
  - It positions your data for the Semantic Web!

Interoperability

Network effect

Set Theory

- RDF is a language with *model-theoretic* semantics
  - Models supposed to be analogue of (part of) world
    - e.g., elements of model correspond to objects in world

↓

- Set-theoretic representation is a natural choice for this language
- The main utility → deep analysis of the nature of the things being described by the language

Introduction to Set Theory

Sets

- *Definitions*
  - A *Set* is any well defined collection of "objects"
  - The elements of a set are the objects in a set
- Set membership
  - $x \in \mathrm{A}$ means that *x* is a member of the set A
  - $x \notin \mathrm{A}$ means that x is not a member of the set A
- Ways of describing sets
  - List the elements $\mathrm{A}=\{1,2,3,4,5,6\}$
  - Give a verbal description
    - "A is the set of all integers from 1 to 6, inclusive
  - Give a mathematical inclusion rule $\mathrm{A}=\{\text{Integers } x \mid 1 \le x \le 6\}$
- Some special sets
  - The Null Set or Empty Set. This is a set with no elements, often symbolizedby $\varnothing$
  - The Universal Set is the set of all elements currently under consideration, and is often symbolized by Ω

- Membership relationships (subset)
  - $A \subseteq B$ "A is a subset of B"
  - We say "A is a subset of B" if $x \in A \Rightarrow x \in B$, i.e., all the members of A are also members of B
  - The notation for subset is very similar to the notation for "less than or equal to," and means, in terms of the sets, "included in or equal to"

- Proper Subset
  - $A \subset B$ "A is a proper subset of B"
  - We say "A is a proper subset of B" if all the members of A are also members of B, but in addition there exists at least one element *c* such that $c \in B$ but $c \notin A$
  - The notation for subset is very similar to the notation for "less than," and means, in terms of the sets, "included in but not equal to"

Operators

- Set *union*: $A \cup B$
  - "*A* union *B*" is the set of all elements that are in *A*, or *B*, or both
  - Similar to the logical "or" operator
- Set *intersection*: $A \cap B$
  - "*A* intersect *B*" is the set of all elements that are in *both* *A* and *B*
  - Similar to the logical "and"
- Set *complement*: $\overline{A}$
  - "*A* complement," or "not *A*" is the set of all elements not in *A*.
  - Similar to the logical not, and is reflexive, that is, $\overline{\overline{A}} = A$
- Set *difference*: $A - B$
  - The set difference "A minus B" is the set of elements that are in A, with those that are in B subtracted out
  - Or the set of elements that are in A, *and* not in B, so $A - B = A \cap \overline{B}$

- Cartesian product (*product* set) of two sets A and B: $A \times B$
  - All pairs such that the first component of which is an element of *A* and the second is an element of *B*

    $$A \times B = \{(a,b) \mid a \in A, b \in B\}$$

- *Power* of set: 2^A
  - A set that contains all subsets of *A* as elements
  - E.g., $2^{\{a,b\}} = \{\Phi, \{a\}, \{b\}, \{a,b\}\}$

- (binary) *relation* between *A* and *B*: $R \subseteq A \times B$
  - $(a,b) \in R$ or aRb
  - A subset of the Cartesian product of *A* and *B*
  - If *A*=*B* then we call it a Relation on *A*
  - *Properties*
    - *Reflexive: if xRx holds for all x*
    - *Symmetric: if xRy implies yRx for x,y*
    - *Transitive: if for all x, y, z from xRy and yRz follows xRz*

Examples

$$\Omega = \{1,2,3,4,5,6\}$$

$$A = \{1,2,3\} \quad B = \{3,4,5,6\}$$

$$A \cap B = \{3\} \quad A \cup B = \{1,2,3,4,5,6\}$$

$$B - A = \{4,5,6\} \quad \overline{B} = \{1,2\}$$

RDF semantics

- Semantics can be given by RDF Model Theory (MT)
  - MT defines relationship between syntax and *interpretations*
    - Can be many interpretations (models) of one piece of syntax
    - Models supposed to be analogue of (part of) world
      - e.g., elements of model correspond to objects in world
    - Formal relationship between syntax and models
      - structure of models reflect relationships specified in syntax
    - Inference (e.g., subsumption) defined in terms of MT
      - By reasoning we mean deriving facts that are not expressed in ontology or in knowledge base explicitly

- Semantics can be given using on the basis of axioms
  - relating it to another well understood representation, e.g., by first-order logic, for which a semantic model exists
  - A benefit of this approach is that the axioms may provide the basis of an "executable semantics"

Introduction to First-order Predicate Logic

Propositional Logic

- Logic provides
  - A representation of knowledge &
  - Automation of the inferencing process

- Formal Logic
  - Propositional Logic
  - Predicate Logic

- Propositional logics
  - Propositional symbols denote propositions or statements about the world that may be either **true** or **false**

Propositional logic connectives			
Conjunction	AND		
Disjunction	OR		
Negation	NOT	A’	
Material implication	If-Then	$\rightarrow$	$\equiv$
Material equivalence	Equals		

Some terms

- Interpretation: the meaning or semantics of a sentence determines its

- Given the truth values of all symbols in a sentence, it can be "evaluated" to determine its truth value (True or False)

- A model for a KB (the "possible world")
  - Assignment of truth values to propositional symbols in which each sentence in the KB is True

More terms …

- Valid sentence or tautology
  - A sentence that is True under all interpretations, no matter what the world is actually like or what the semantics is
  - e.g., "It's raining or it's not raining"
- Inconsistent sentence or contradiction
  - a sentence that is False under all interpretations. The world is never like what it describes
  - e.g., "It's raining and it's not raining"
- P entails Q, P |= Q
  - whenever P is True, so is Q
  - all models of P are also models of Q

Predicate Logic

- Propositional logic drawbacks
  - can only deal with complete sentences
    - i.e. it can not examine the internal structure of a statement
  - too simple for complex domains
  - no support for inferencing
  - doesn't handle fuzzy concepts

- Predicate logic was developed in order to analyze more general cases
  - Propositional logic is a subset of predicate logic
  - Concerned with internal structure of sentences
  - Quantifiers – *for all*, *there exists some*, *there exists no* - make sentence more exact
  - Wider scope of expression

- Predicate logic
  - First-order logic
  - Second-order logic
  - Higher-order logics

FOL Syntax

- User defines these primitives:
  - **Constant symbols**
    - "individuals" in the world)
    - eg, Mary, 3, …
  - **Function symbols**
    - mapping individuals to individuals
    - e.g., father-of(Mary) = John, color-of(Sky) = Blue
  - **Predicate symbols**
    - mapping from individuals to truth values
    - e.g., greater(5,3), green(Grass), color(Grass, Green)
- FOL supplies these primitives:
  - **Variable symbols**
    - x,y,…
  - **Connectives**
    - Same as in PL: not (~), and (^), or (v), implies (=>), if and only if (<=>)
  - **Quantifiers**
    - Universal (∀) and Existential (∃)
-

Quantifiers

- Universal quantification
  - corresponds to conjunction ("and")
  - $(\forall x)P(x)$ means that P holds for all values of x in the domain associated with that variable
    - e.g., $(\forall x)\ dolphin(x) \Rightarrow mammal(x)$
- Existential quantification
  - corresponds to disjunction ("or")
  - $(\exists x)P(x)$ means that P holds for some value of x in the domain associated with that variable
    - e.g., $(\exists x)\ mammal(x) \wedge lays-eggs(x)$
- Universal quantifiers are usually used with "implies" to form "if-then rules"
  - e.g., $(\forall x)\ TDT44-student(x) \Rightarrow smart(x)$ means "All TDT44 students are smart" :D
  - You rarely use universal quantification to make blanket statements about every individual in the world: $(\forall x)\ TDT44-student(x) \wedge smart(x)$ meaning that everyone in the world is a TDT44 student and is smart!!!!

Quantifiers …

- Existential quantifiers are usually used with "and" to specify a list of properties or facts about an individual
  - e.g., $(\exists x)\ TDT44-student(x) \wedge smart(x)$ means "there is a TDT44 student who is smart"
  - A common mistake is to represent this English sentence as the FOL sentence: $(\exists x)\ TDT44-student(x) \Rightarrow smart(x)$

- Switching the order of universal quantifiers does not change the meaning
  - $(\forall x)(\forall y)\, P(x,y)$ is logically equivalent to $(\forall y)(\forall x)\, P(x,y)$
  - Similarly, you can switch the order of existential quantifiers

- Switching the order of universals and existentials *does* change meaning
  - Everyone likes someone: $(\forall x)(\exists y)\, likes(x,y)$
  - Someone is liked by everyone: $(\exists y)(\forall x)\, likes(x,y)$

First-Order Logic (FOL) *Syntax…*

- *Sentences* are built up of *terms* and *atoms*:
  - A term
    - denoting a real-world object
    - a constant symbol, a variable symbol, or a function
    - e.g., left-leg-of ( )
    - x and $f(x_1, ..., x_n)$ are terms, where each x_i is a term.
  - An atom
    - has value true or false
    - if **P** and **Q** are atoms, then **~P, P V Q, P ^ Q, P => Q, P <=> Q** are atoms
  - A sentence
    - an atom, or
    - if **P** is a sentence and **x** is a variable, then **(∀x)P** and **(∃ x)P** are sentences
  - A well-formed formula (wff)
    - a sentence containing no "free" variables. i.e., all variables are "bound" by universal or existential quantifiers
    - e.g., **(∀x)P(x,y)** has **x** bound as a universally quantified variable, but **y** is free

Translating English to FOL

- Every gardener likes the sun.

  `(Ax) gardener(x) => likes(x,Sun)`

- You can fool some of the people all of the time.

  `(Ex)(At) (person(x) ^ time(t)) => can-fool(x,t)`

- You can fool all of the people some of the time.

  `(Ax)(Et) (person(x) ^ time(t) => can-fool(x,t)`

- All purple mushrooms are poisonous.

  `(Ax) (mushroom(x) ^ purple(x)) => poisonous(x)`

Translating English to FOL…

- No purple mushroom is poisonous.
  `~(Ex) purple(x) ^ mushroom(x) ^ poisonous(x)`
  or, equivalently,
  `(Ax) (mushroom(x) ^ purple(x)) => ~poisonous(x)`

- There are exactly two purple mushrooms.
  `(Ex)(Ey) mushroom(x) ^ purple(x) ^ mushroom(y) ^ purple(y) ^ ~(x=y) ^`
  `(Az) (mushroom(z) ^ purple(z)) => ((x=z) v (y=z))`

- Deb is not tall.
  `~tall(Deb)`

- X is above Y if X is on directly on top of Y or else there is a pile of one or more other objects directly on top of one another starting with X and ending with Y.
  `(Ax)(Ay) above(x,y) <=> (on(x,y) v (Ez) (on(x,z) ^`
  `above(z,y)))`

Inference

- Inference in formal logic is the process of generating new wffs from existing wffs (KB) through the application of rules of inference
  - An inference rule is **sound** if
    - every sentence X produced by an inference rule operating on a KB, logically follows from the KB
    - the inference rule does not create any contradictions
  - An inference rule is **complete** if
    - it is able to produce every expression that logically follows from (is entailed by) the KB

- Inference rules for PL apply to FOL as well, e.g.,
  - Modus Ponens
  - And-Introduction
  - And-Elimination
  - …

Inference …

- New sound inference rules for use with quantifiers
  - Universal Elimination
    - If `(Ax)P(x)` is true, then `P(c)` is true
  - Existential Introduction
    - If `P(c)` is true, then `(Ex)P(x)` is inferred
  - Existential Elimination
    - From `(Ex)P(x)` infer `P(c)`
  - Paramodulation
    - From P(a) and a=b derive P(b)
  - Generalized Modus Ponens
    - from P(c), Q(c), and (Ax)(P(x) ^ Q(x)) => R(x), derive R(c)
  - …

Next Session

- September 16, x:00-xx:00
  - Chapter 1:
  - Chapter 2:
  - Chapter 3: