

# Semantic Days 2010 Tutorial

## Semantic Web Technologies

### Lecture 1: Introduction

Martin Giese

31st May 2010



DEPARTMENT OF  
INFORMATICS



UNIVERSITY OF  
OSLO

# Today's Plan

- 1 Practicalities
- 2 Introduction to Semantic Technologies

# Outline

- 1 Practicalities
- 2 Introduction to Semantic Technologies

# Who?

## Teachers:



Martin Giese  
([martingi@ifi.uio.no](mailto:martingi@ifi.uio.no))



Martin Georg Skjæveland  
([martige@ifi.uio.no](mailto:martige@ifi.uio.no))

# Tutorial Structure

- Introduction
  - Lecture, 45 mins
  - Break, install software, 15 mins
- RDF – representing information
  - Lecture, 30 mins
  - Break, solve exercises, 15 mins
  - Solutions to exercises, 15 mins
- SPARQL – querying
  - 30 mins lecture, 15 mins break/solving, 15 mins solutions
- OWL – modeling
  - 30 mins lecture, 15 mins break/solving, 15 mins solutions
- D2R – mapping databases
  - 30 mins lecture, 15 mins break/solving, 15 mins solutions

# Used Software, Tutorial Web Page

- You will need:
  - A text editor for writing RDF
  - A web browser to play with some web services
  - Protégé for modelling
  - Some small Java apps we built for you
- Download
  - Required Software
  - Source code
  - Exercises
  - Solutions
- ...from the tutorial web page:

`http://sws.ifi.uio.no/semdays2010/`

# Outline

- 1 Practicalities
- 2 Introduction to Semantic Technologies

# The Vision of a Semantic Web

## A vision

I have a dream for the Web [in which computers] become capable of analyzing all the data on the Web—the content, links, and transactions between people and computers. A ‘Semantic Web’, which should make this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines. The ‘intelligent agents’ people have touted for ages will finally materialize.



Tim Berners-Lee

Quoted from: *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web*. Tim Berners-Lee with Mark Fischetti. Harper San Francisco, 1999.



# Let's go to the cinema!

- Kringsjå studentby, 20:00. . .

# Let's go to the cinema!

- Kringsjå studentby, 20:00. . .
- "Let's go to see *Avatar* now!"



# Let's go to the cinema!

- Kringsjå studentby, 20:00...
- "Let's go to see *Avatar* now!"
- Need to find out which cinema plays the movie tonight, e.g. on <http://www.google.no/movies>



# Let's go to the cinema!

- Kringsjå studentby, 20:00...
- "Let's go to see *Avatar* now!"
- Need to find out which cinema plays the movie tonight, e.g. on <http://www.google.no/movies>
- Need to find out where those cinemas are



# Let's go to the cinema!

- Kringsjå studentby, 20:00. . .
- “Let's go to see *Avatar* now!”
- Need to find out which cinema plays the movie tonight, e.g. on <http://www.google.no/movies>
- Need to find out where those cinemas are
- Need to find out which of those cinemas we can reach on time using public transport, e.g. on <http://www.trafikanten.no/>



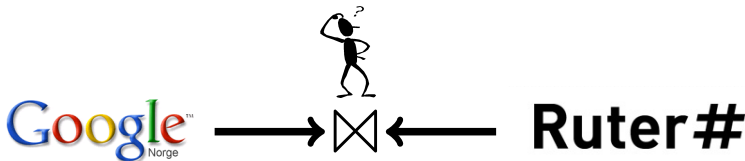
# Let's go to the cinema!

- Kringsjå studentby, 20:00. . .
- “Let's go to see *Avatar* now!”
- Need to find out which cinema plays the movie tonight, e.g. on <http://www.google.no/movies>
- Need to find out where those cinemas are
- Need to find out which of those cinemas we can reach on time using public transport, e.g. on <http://www.trafikanten.no/>
- Web user needs to combine information from different sites



# Let's go to the cinema!

- Kringsjå studentby, 20:00...
- "Let's go to see *Avatar* now!"
- Need to find out which cinema plays the movie tonight, e.g. on <http://www.google.no/movies>
- Need to find out where those cinemas are
- Need to find out which of those cinemas we can reach on time using public transport, e.g. on <http://www.trafikanten.no/>
- Web user needs to combine information from different sites
- Essentially a database join!



# The Solution?

- Wait for Google to produce a Cinema+Public Transport mashup?

**Ruter#**



# The Solution?

- Wait for Google to produce a Cinema+Public Transport mashup?

**Ruter#**

- But what about

# The Solution?

- Wait for Google to produce a Cinema+Public Transport mashup?

Ruter#

- But what about
  - Real estate + public transport?

# The Solution?

- Wait for Google to produce a Cinema+Public Transport mashup?

Ruter#

- But what about
  - Real estate + public transport?
  - Plane schedules and pricing + weather information?

# The Solution?

- Wait for Google to produce a Cinema+Public Transport mashup?

Ruter#

- But what about
  - Real estate + public transport?
  - Plane schedules and pricing + weather information?
  - Car rental + tourism?

# The Solution?

- Wait for Google to produce a Cinema+Public Transport mashup?

Ruter#

- But what about
  - Real estate + public transport?
  - Plane schedules and pricing + weather information?
  - Car rental + tourism?
  - Public information + private information (preferences, calendar, location, etc.)

# The Solution?

- Wait for Google to produce a Cinema+Public Transport mashup?

Ruter#

- But what about
  - Real estate + public transport?
  - Plane schedules and pricing + weather information?
  - Car rental + tourism?
  - Public information + private information (preferences, calendar, location, etc.)
- Can hardly wait for a separate mashup for each useful combination!

# A Web of Data!

Imagine. . .

- All those websites publish their information in a machine-readable format.

# A Web of Data!

Imagine. . .

- All those websites publish their information in a machine-readable format.
- The data published by different sources is *linked*



# A Web of Data!

Imagine. . .

- All those websites publish their information in a machine-readable format.
- The data published by different sources is *linked*
- Enough domain knowledge is available to machines to make use of the information

# A Web of Data!

Imagine. . .

- All those websites publish their information in a machine-readable format.
- The data published by different sources is *linked*
- Enough domain knowledge is available to machines to make use of the information
- User-agents can find and combine published information in appropriate ways to answer the user's information needs.

# But How?

- This sounds like a nice idea, but how can it work?

# But How?

- This sounds like a nice idea, but how can it work?
- There has been a lot of hype around the Semantic Web!

# But How?

- This sounds like a nice idea, but how can it work?
- There has been a lot of hype around the Semantic Web!
- Visions instantly transformed to promises (and \$\$\$)

# But How?

- This sounds like a nice idea, but how can it work?
- There has been a lot of hype around the Semantic Web!
- Visions instantly transformed to promises (and \$\$\$)
- Most of this simply does not work (yet?)

# But How?

- This sounds like a nice idea, but how can it work?
- There has been a lot of hype around the Semantic Web!
- Visions instantly transformed to promises (and \$\$\$)
- Most of this simply does not work (yet?)
- But then, a lot does!

# But How?

- This sounds like a nice idea, but how can it work?
- There has been a lot of hype around the Semantic Web!
- Visions instantly transformed to promises (and \$\$\$)
- Most of this simply does not work (yet?)
- But then, a lot does!
- Current *partial* solutions build on traditions of



# But How?

- This sounds like a nice idea, but how can it work?
- There has been a lot of hype around the Semantic Web!
- Visions instantly transformed to promises (and \$\$\$)
- Most of this simply does not work (yet?)
- But then, a lot does!
- Current *partial* solutions build on traditions of
  - Modelling

# But How?

- This sounds like a nice idea, but how can it work?
- There has been a lot of hype around the Semantic Web!
- Visions instantly transformed to promises (and \$\$\$)
- Most of this simply does not work (yet?)
- But then, a lot does!
- Current *partial* solutions build on traditions of
  - Modelling
  - Calculating with Knowledge

# But How?

- This sounds like a nice idea, but how can it work?
- There has been a lot of hype around the Semantic Web!
- Visions instantly transformed to promises (and \$\$\$)
- Most of this simply does not work (yet?)
- But then, a lot does!
- Current *partial* solutions build on traditions of
  - Modelling
  - Calculating with Knowledge
  - Information Exchange (XML, HTTP, etc.)

# Building Models

- A *model* is a simplified representation of certain aspects of the real world.

# Building Models

- A *model* is a simplified representation of certain aspects of the real world.
- Made for

# Building Models

- A *model* is a simplified representation of certain aspects of the real world.
- Made for
  - understanding

# Building Models

- A *model* is a simplified representation of certain aspects of the real world.
- Made for
  - understanding
  - structuring

# Building Models

- A *model* is a simplified representation of certain aspects of the real world.
- Made for
  - understanding
  - structuring
  - predicting



# Building Models

- A *model* is a simplified representation of certain aspects of the real world.
- Made for
  - understanding
  - structuring
  - predicting
  - communicating

# Building Models

- A *model* is a simplified representation of certain aspects of the real world.
- Made for
  - understanding
  - structuring
  - predicting
  - communicating
- Can be

# Building Models

- A *model* is a simplified representation of certain aspects of the real world.
- Made for
  - understanding
  - structuring
  - predicting
  - communicating
- Can be
  - Taxonomies (e.g. species, genus, family, etc. in biology)

# Building Models

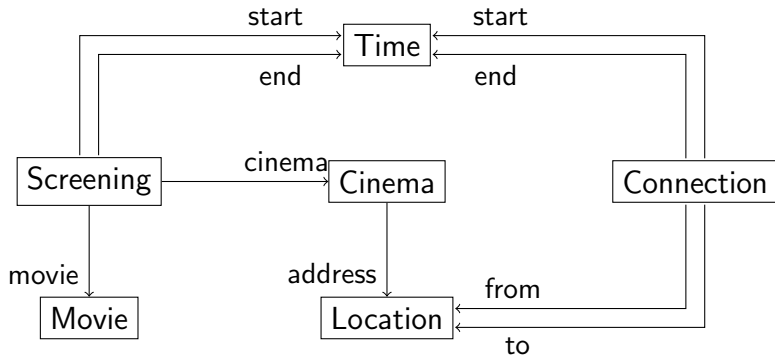
- A *model* is a simplified representation of certain aspects of the real world.
- Made for
  - understanding
  - structuring
  - predicting
  - communicating
- Can be
  - Taxonomies (e.g. species, genus, family, etc. in biology)
  - Domain models, e.g. in UML

# Building Models

- A *model* is a simplified representation of certain aspects of the real world.
- Made for
  - understanding
  - structuring
  - predicting
  - communicating
- Can be
  - Taxonomies (e.g. species, genus, family, etc. in biology)
  - Domain models, e.g. in UML
  - Numerical Models (Newtonian mechanics, Quantum mechanics)

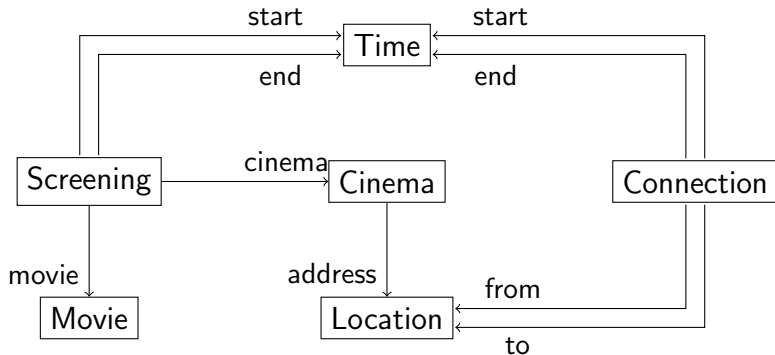
# A Cinema Transport Model

An example of a UML domain model:



# A Cinema Transport Model

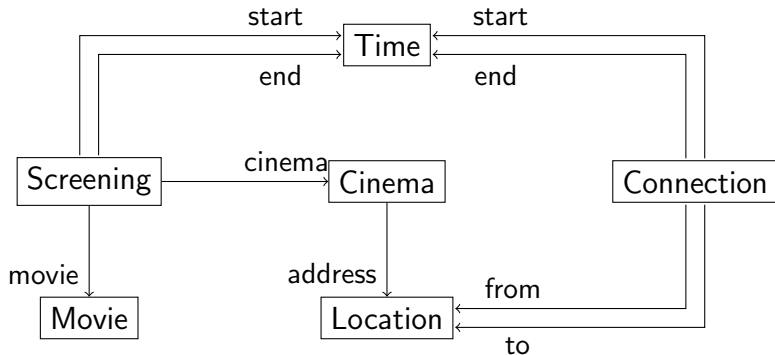
An example of a UML domain model:



- What is the vocabulary?

# A Cinema Transport Model

An example of a UML domain model:



- What is the vocabulary?
- How is it connected?



# A Query

What is it we want?

- Screening(s), movie(s, AVATAR)

# A Query

What is it we want?

- Screening(**s**), movie(**s**, AVATAR)
- cinema(**s**, **k**), address(**k**, l)

# A Query

What is it we want?

- Screening(**s**), movie(**s**, AVATAR)
- cinema(**s**, **k**), address(**k**, **l**)
- Connection(**c**), from(**c**, KRINGSJÅ), to(**c**, **l**)

# A Query

What is it we want?

- Screening(**s**), movie(**s**, AVATAR)
- cinema(**s**, **k**), address(**k**, **l**)
- Connection(**c**), from(**c**, KRINGSJÅ), to(**c**, **l**)
- start(**c**, **cStart**), *before*(20:00, **cStart**)

# A Query

What is it we want?

- Screening(*s*), movie(*s*, AVATAR)
- cinema(*s*, *k*), address(*k*, *l*)
- Connection(*c*), from(*c*, KRINGSJÅ), to(*c*, *l*)
- start(*c*, *cStart*), before(20:00, *cStart*)
- end(*c*, *cEnd*), start(*s*, *sStart*), before(*cEnd*, *sStart*)

# A Query

What is it we want?

- Screening(*s*), movie(*s*, AVATAR)
- cinema(*s*, *k*), address(*k*, *l*)
- Connection(*c*), from(*c*, KRINGSJÅ), to(*c*, *l*)
- start(*c*, *cStart*), before(20:00, *cStart*)
- end(*c*, *cEnd*), start(*s*, *sStart*), before(*cEnd*, *sStart*)

Find *s*, *k*, *l*, *c*, *cStart*, *cEnd*, *sStart* satisfying this and we have the answer!

# A Query

What is it we want?

- Screening(*s*), movie(*s*, AVATAR)
- cinema(*s*, *k*), address(*k*, *l*)
- Connection(*c*), from(*c*, KRINGSJÅ), to(*c*, *l*)
- start(*c*, *cStart*), before(20:00, *cStart*)
- end(*c*, *cEnd*), start(*s*, *sStart*), before(*cEnd*, *sStart*)

Find *s*, *k*, *l*, *c*, *cStart*, *cEnd*, *sStart* satisfying this and we have the answer!

- Maybe not the easiest way to ask, but it's a start.
- Models are an important part of a Web of Data!
- Need to connect models from different domains.

# Calculating

- What is calculation?



# Calculating

- What is calculation?

$$\begin{array}{l} A \text{ owns } x \text{ } Bs \\ A \text{ gets another } y \text{ } Bs \\ \hline A \text{ now owns } (x + y) \text{ } Bs \end{array}$$

# Calculating

- What is calculation?

$$\begin{array}{l}
 A \text{ owns } x \text{ } Bs \\
 A \text{ gets another } y \text{ } Bs \\
 \hline
 A \text{ now owns } (x + y) \text{ } Bs
 \end{array}$$

e.g.

$$\begin{array}{l}
 \text{Peter} \text{ owns } 2 \text{ apples} \\
 \text{Peter} \text{ gets another } 3 \text{ apples} \\
 \hline
 \text{Peter} \text{ now owns } 5 \text{ apples}
 \end{array}$$

# Calculating

- What is calculation?

$$\begin{array}{l} A \text{ owns } x \text{ } Bs \\ A \text{ gets another } y \text{ } Bs \\ \hline A \text{ now owns } (x + y) \text{ } Bs \end{array}$$

e.g.

$$\begin{array}{l} \text{Peter} \text{ owns } 2 \text{ apples} \\ \text{Peter} \text{ gets another } 3 \text{ apples} \\ \hline \text{Peter} \text{ now owns } 5 \text{ apples} \end{array}$$

- Calculation is algorithmic manipulation of numbers. . .

# Calculating

- What is calculation?

$$\begin{array}{l} A \text{ owns } x \text{ } Bs \\ A \text{ gets another } y \text{ } Bs \\ \hline A \text{ now owns } (x + y) \text{ } Bs \end{array}$$

e.g.

$$\begin{array}{l} \text{Peter owns } 2 \text{ apples} \\ \text{Peter gets another } 3 \text{ apples} \\ \hline \text{Peter now owns } 5 \text{ apples} \end{array}$$

- Calculation is algorithmic manipulation of numbers. . .
- . . . where the *meaning* of the numbers is not needed

# Calculating

- What is calculation?

$$\begin{array}{l}
 A \text{ owns } x \text{ } Bs \\
 A \text{ gets another } y \text{ } Bs \\
 \hline
 A \text{ now owns } (x + y) \text{ } Bs
 \end{array}$$

e.g.

$$\begin{array}{l}
 \text{Peter owns } 2 \text{ apples} \\
 \text{Peter gets another } 3 \text{ apples} \\
 \hline
 \text{Peter now owns } 5 \text{ apples}
 \end{array}$$

- Calculation is algorithmic manipulation of numbers. . .
- . . . where the *meaning* of the numbers is not needed
- Can calculate  $2 + 3 = 5$  without knowing what is counted

# Calculating

- What is calculation?

$$\begin{array}{l} A \text{ owns } x \text{ } Bs \\ A \text{ gets another } y \text{ } Bs \\ \hline A \text{ now owns } (x + y) \text{ } Bs \end{array}$$

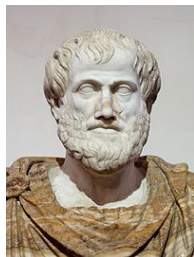
e.g.

$$\begin{array}{l} \text{Peter owns } 2 \text{ apples} \\ \text{Peter gets another } 3 \text{ apples} \\ \hline \text{Peter now owns } 5 \text{ apples} \end{array}$$

- Calculation is algorithmic manipulation of numbers. . .
- . . . where the *meaning* of the numbers is not needed
- Can calculate  $2 + 3 = 5$  without knowing what is counted
- Abstraction!

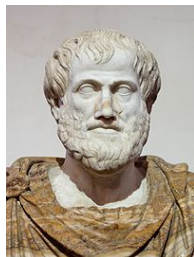
# Calculating with Knowledge

- Can be traced back to Aristotle (384–322 BC)



# Calculating with Knowledge

- Can be traced back to Aristotle (384–322 BC)
- Modus Barbara:

$$\begin{array}{l} \text{All } A \text{ are } B \\ \text{All } B \text{ are } C \\ \hline \text{All } A \text{ are } C \end{array}$$


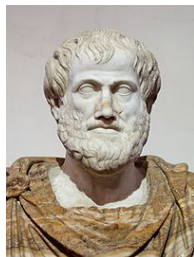


# Calculating with Knowledge

- Can be traced back to Aristotle (384–322 BC)
- Modus Barbara:

$$\begin{array}{l} \text{All } A \text{ are } B \\ \text{All } B \text{ are } C \\ \hline \text{All } A \text{ are } C \end{array}$$

e.g.

$$\begin{array}{l} \text{All } \text{Greeks} \text{ are } \text{men} \\ \text{All } \text{men} \text{ are } \text{mortal} \\ \hline \text{All } \text{Greeks} \text{ are } \text{mortal} \end{array}$$


# Calculating with Knowledge

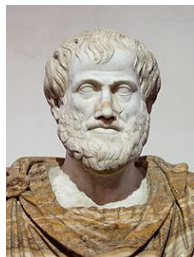
- Can be traced back to Aristotle (384–322 BC)
- Modus Barbara:

$$\begin{array}{l} \text{All } A \text{ are } B \\ \text{All } B \text{ are } C \\ \hline \text{All } A \text{ are } C \end{array}$$

e.g.

$$\begin{array}{l} \text{All } \text{Greeks} \text{ are } \text{men} \\ \text{All } \text{men} \text{ are } \text{mortal} \\ \hline \text{All } \text{Greeks} \text{ are } \text{mortal} \end{array}$$

- Algorithmic manipulation of *knowledge*...



# Calculating with Knowledge

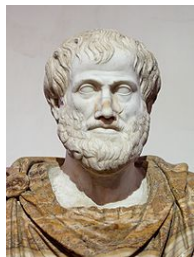
- Can be traced back to Aristotle (384–322 BC)
- Modus Barbara:

$$\begin{array}{l} \text{All } A \text{ are } B \\ \text{All } B \text{ are } C \\ \hline \text{All } A \text{ are } C \end{array}$$

e.g.

$$\begin{array}{l} \text{All } \text{Greeks} \text{ are } \text{men} \\ \text{All } \text{men} \text{ are } \text{mortal} \\ \hline \text{All } \text{Greeks} \text{ are } \text{mortal} \end{array}$$

- Algorithmic manipulation of *knowledge*...
- ...where the *meaning* of the words is not needed!



# Calculating with Knowledge

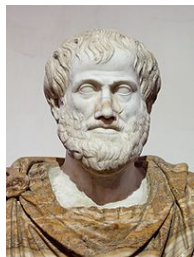
- Can be traced back to Aristotle (384–322 BC)
- Modus Barbara:

$$\begin{array}{l} \text{All } A \text{ are } B \\ \text{All } B \text{ are } C \\ \hline \text{All } A \text{ are } C \end{array}$$

e.g.

$$\begin{array}{l} \text{All } \text{Greeks} \text{ are } \text{men} \\ \text{All } \text{men} \text{ are } \text{mortal} \\ \hline \text{All } \text{Greeks} \text{ are } \text{mortal} \end{array}$$

- Algorithmic manipulation of *knowledge*...
- ...where the *meaning* of the words is not needed!
- Also an abstraction!



# Calculating with Knowledge

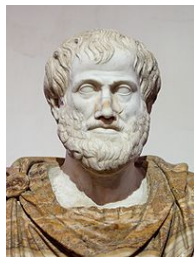
- Can be traced back to Aristotle (384–322 BC)
- Modus Barbara:

$$\begin{array}{l} \text{All } A \text{ are } B \\ \text{All } B \text{ are } C \\ \hline \text{All } A \text{ are } C \end{array}$$

e.g.

$$\begin{array}{l} \text{All } \text{Greeks} \text{ are } \text{men} \\ \text{All } \text{men} \text{ are } \text{mortal} \\ \hline \text{All } \text{Greeks} \text{ are } \text{mortal} \end{array}$$

- Algorithmic manipulation of *knowledge*...
- ...where the *meaning* of the words is not needed!
- Also an abstraction!
- The topic of *formal logic* and *computational logic*



# Computing with Knowledge About Movies

- Query: find a *fun event* we can reach by public transport

# Computing with Knowledge About Movies

- Query: find a *fun event* we can reach by public transport
- Knowledge base:

# Computing with Knowledge About Movies

- Query: find a *fun event* we can reach by public transport
- Knowledge base:
  - ① A movie screening is an event



# Computing with Knowledge About Movies

- Query: find a *fun event* we can reach by public transport
- Knowledge base:
  - ① A movie screening is an event
  - ② A movie screening is fun if the movie being shown is not a documentary

# Computing with Knowledge About Movies

- Query: find a *fun event* we can reach by public transport
- Knowledge base:
  - ① A movie screening is an event
  - ② A movie screening is fun if the movie being shown is not a documentary
  - ③ James Cameron does not direct documentaries

# Computing with Knowledge About Movies

- Query: find a *fun event* we can reach by public transport
- Knowledge base:
  - ① A movie screening is an event
  - ② A movie screening is fun if the movie being shown is not a documentary
  - ③ James Cameron does not direct documentaries
  - ④ James Cameron directed *Avatar*

# Computing with Knowledge About Movies

- Query: find a *fun event* we can reach by public transport
- Knowledge base:
  - ① A movie screening is an event
  - ② A movie screening is fun if the movie being shown is not a documentary
  - ③ James Cameron does not direct documentaries
  - ④ James Cameron directed *Avatar*
  - ⑤ There is a screening of *Avatar* at 19:00.

# Computing with Knowledge About Movies

- Query: find a *fun event* we can reach by public transport
- Knowledge base:
  - ① A movie screening is an event
  - ② A movie screening is fun if the movie being shown is not a documentary
  - ③ James Cameron does not direct documentaries
  - ④ James Cameron directed *Avatar*
  - ⑤ There is a screening of *Avatar* at 19:00.

...

# Computing with Knowledge About Movies

- Query: find a *fun event* we can reach by public transport
- Knowledge base:
  - ① A movie screening is an event
  - ② A movie screening is fun if the movie being shown is not a documentary
  - ③ James Cameron does not direct documentaries
  - ④ James Cameron directed *Avatar*
  - ⑤ There is a screening of *Avatar* at 19:00.

...
- Let's deduce...

# Computing with Knowledge About Movies

- Query: find a *fun event* we can reach by public transport
- Knowledge base:
  - ① A movie screening is an event
  - ② A movie screening is fun if the movie being shown is not a documentary
  - ③ James Cameron does not direct documentaries
  - ④ James Cameron directed *Avatar*
  - ⑤ There is a screening of *Avatar* at 19:00.
  - ...
- Let's deduce...
  - ⑥ From 3 and 4: *Avatar* is not a documentary

# Computing with Knowledge About Movies

- Query: find a *fun event* we can reach by public transport
- Knowledge base:
  - ① A movie screening is an event
  - ② A movie screening is fun if the movie being shown is not a documentary
  - ③ James Cameron does not direct documentaries
  - ④ James Cameron directed *Avatar*
  - ⑤ There is a screening of *Avatar* at 19:00.

...
- Let's deduce...
  - ⑥ From 3 and 4: *Avatar* is not a documentary
  - ⑦ From 6 and 2: A screening of *Avatar* is fun



# Computing with Knowledge About Movies

- Query: find a *fun event* we can reach by public transport
- Knowledge base:
  - ① A movie screening is an event
  - ② A movie screening is fun if the movie being shown is not a documentary
  - ③ James Cameron does not direct documentaries
  - ④ James Cameron directed *Avatar*
  - ⑤ There is a screening of *Avatar* at 19:00.
  - ...
- Let's deduce...
  - ⑥ From 3 and 4: *Avatar* is not a documentary
  - ⑦ From 6 and 2: A screening of *Avatar* is fun
  - ⑧ From 1, 5, 7: there is a fun event at 19:00

# Computing with Knowledge About Movies

- Query: find a *fun event* we can reach by public transport
- Knowledge base:
  - ① A movie screening is an event
  - ② A movie screening is fun if the movie being shown is not a documentary
  - ③ James Cameron does not direct documentaries
  - ④ James Cameron directed *Avatar*
  - ⑤ There is a screening of *Avatar* at 19:00....
- Let's deduce...
  - ⑥ From 3 and 4: *Avatar* is not a documentary
  - ⑦ From 6 and 2: A screening of *Avatar* is fun
  - ⑧ From 1, 5, 7: there is a fun event at 19:00...

# Computing with Knowledge About Movies

- Query: find a *fun event* we can reach by public transport
- Knowledge base:
  - ① A movie screening is an event
  - ② A movie screening is fun if the movie being shown is not a documentary
  - ③ James Cameron does not direct documentaries
  - ④ James Cameron directed *Avatar*
  - ⑤ There is a screening of *Avatar* at 19:00....
- Let's deduce...
  - ⑥ From 3 and 4: *Avatar* is not a documentary
  - ⑦ From 6 and 2: A screening of *Avatar* is fun
  - ⑧ From 1, 5, 7: there is a fun event at 19:00...
- Computing with Knowledge is an important part of a Web of Data!

# The “Home” of the Semantic Web

See the W3C pages for the Semantic Web effort:

<http://www.w3.org/2001/sw/>

For standards (RDF, OWL, SPARQL, etc.), see:

[http://www.w3.org/2001/sw/wiki/Main\\_Page](http://www.w3.org/2001/sw/wiki/Main_Page)



# Semantic technologies

- Tim Berners-Lee's vision of a Semantic Web is still far away!

# Semantic technologies

- Tim Berners-Lee's vision of a Semantic Web is still far away!
- Then what is this tutorial about?

# Semantic technologies

- Tim Berners-Lee's vision of a Semantic Web is still far away!
- Then what is this tutorial about?
- Let's have a look at what we do have:

# Semantic technologies

- Tim Berners-Lee's vision of a Semantic Web is still far away!
- Then what is this tutorial about?
- Let's have a look at what we do have:
  - W3C standards: RDF, SPARQL, OWL, some more



# Semantic technologies

- Tim Berners-Lee's vision of a Semantic Web is still far away!
- Then what is this tutorial about?
- Let's have a look at what we do have:
  - W3C standards: RDF, SPARQL, OWL, some more
  - Technology like reasoners, ontology editors

# Semantic technologies

- Tim Berners-Lee's vision of a Semantic Web is still far away!
- Then what is this tutorial about?
- Let's have a look at what we do have:
  - W3C standards: RDF, SPARQL, OWL, some more
  - Technology like reasoners, ontology editors
  - Interfacing to relational databases, etc.

# Semantic technologies

- Tim Berners-Lee's vision of a Semantic Web is still far away!
- Then what is this tutorial about?
- Let's have a look at what we do have:
  - W3C standards: RDF, SPARQL, OWL, some more
  - Technology like reasoners, ontology editors
  - Interfacing to relational databases, etc.
  - Existing ontologies for applications in medicine, industry, some of them with over 1M concepts

# Semantic technologies

- Tim Berners-Lee's vision of a Semantic Web is still far away!
- Then what is this tutorial about?
- Let's have a look at what we do have:
  - W3C standards: RDF, SPARQL, OWL, some more
  - Technology like reasoners, ontology editors
  - Interfacing to relational databases, etc.
  - Existing ontologies for applications in medicine, industry, some of them with over 1M concepts
- Possible, and a lot easier, to use Semantic Web technologies for more closed, controlled applications

# Semantic technologies

- Tim Berners-Lee's vision of a Semantic Web is still far away!
- Then what is this tutorial about?
- Let's have a look at what we do have:
  - W3C standards: RDF, SPARQL, OWL, some more
  - Technology like reasoners, ontology editors
  - Interfacing to relational databases, etc.
  - Existing ontologies for applications in medicine, industry, some of them with over 1M concepts
- Possible, and a lot easier, to use Semantic Web technologies for more closed, controlled applications
- We talk about “semantic technologies” since they make sense independent of the Web

# Data integration

- One of the foremost problems in industry today

# Data integration

- One of the foremost problems in industry today
  - within one organization

# Data integration

- One of the foremost problems in industry today
  - within one organization
  - between organizations



# Data integration

- One of the foremost problems in industry today
  - within one organization
  - between organizations
- Enormous amounts of data gathered over the last decades

# Data integration

- One of the foremost problems in industry today
  - within one organization
  - between organizations
- Enormous amounts of data gathered over the last decades
  - different formats, different data models

# Data integration

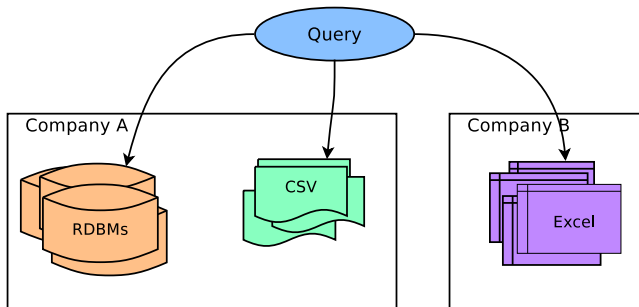
- One of the foremost problems in industry today
  - within one organization
  - between organizations
- Enormous amounts of data gathered over the last decades
  - different formats, different data models
  - specialists needed to find, access, convert data when it is needed

# Data integration

- One of the foremost problems in industry today
  - within one organization
  - between organizations
- Enormous amounts of data gathered over the last decades
  - different formats, different data models
  - specialists needed to find, access, convert data when it is needed
  - large need for automated, unified data access

# Data integration

- One of the foremost problems in industry today
  - within one organization
  - between organizations
- Enormous amounts of data gathered over the last decades
  - different formats, different data models
  - specialists needed to find, access, convert data when it is needed
  - large need for automated, unified data access



# Ontology-based data access

- Use ontology (= model) to define common vocabulary

# Ontology-based data access

- Use ontology (= model) to define common vocabulary
- Possibly by connecting ontologies for different sources using mediating ontologies

# Ontology-based data access

- Use ontology (= model) to define common vocabulary
- Possibly by connecting ontologies for different sources using mediating ontologies
- Create mappings between the common vocabulary and what is in the data sources.



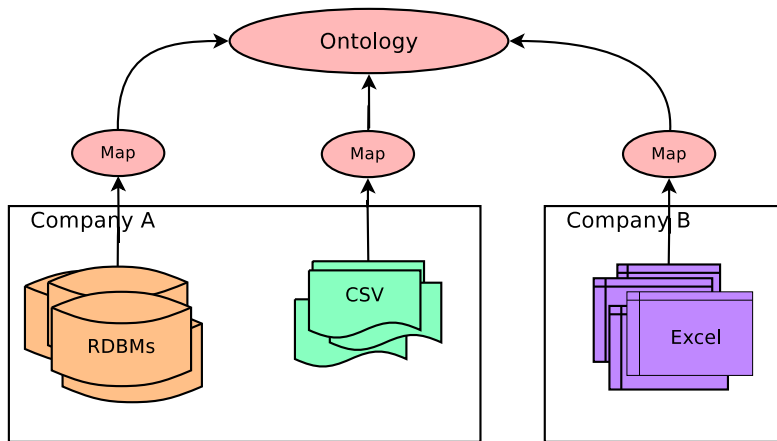
# Ontology-based data access

- Use ontology (= model) to define common vocabulary
- Possibly by connecting ontologies for different sources using mediating ontologies
- Create mappings between the common vocabulary and what is in the data sources.
- Access data using queries expressed using the common vocabulary

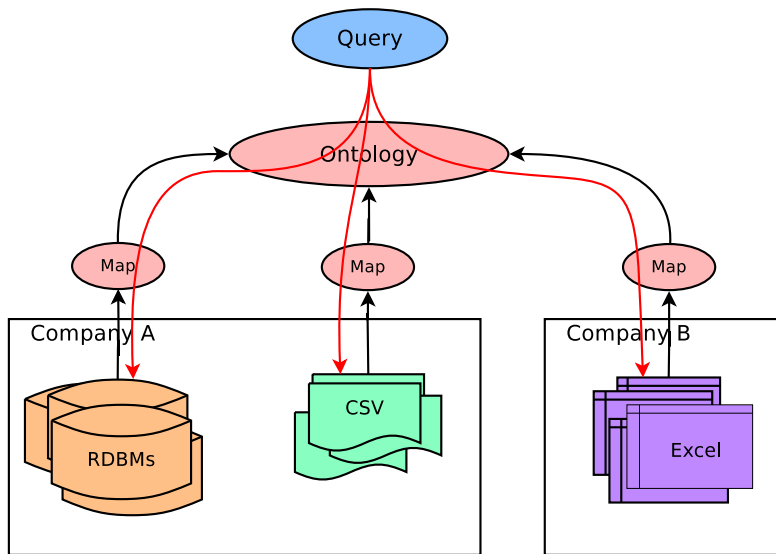
# Ontology-based data access

- Use ontology (= model) to define common vocabulary
- Possibly by connecting ontologies for different sources using mediating ontologies
- Create mappings between the common vocabulary and what is in the data sources.
- Access data using queries expressed using the common vocabulary
- Background machinery gives answers as if data had always been stored according to a common data model

## Ontology-based data access (cont.)



## Ontology-based data access (cont.)



# This Tutorial

The aim of this tutorial is to give you a glimpse of...

- some basic semantic web technologies (RDF, OWL)

# This Tutorial

The aim of this tutorial is to give you a glimpse of...

- some basic semantic web technologies (RDF, OWL)
- some underlying concepts like reasoning

# This Tutorial

The aim of this tutorial is to give you a glimpse of...

- some basic semantic web technologies (RDF, OWL)
- some underlying concepts like reasoning
- ... to give you some overview of the field

# This Tutorial

The aim of this tutorial is to give you a glimpse of...

- some basic semantic web technologies (RDF, OWL)
- some underlying concepts like reasoning
- ... to give you some overview of the field
- ... as an entry point for further exploration



# This Tutorial

The aim of this tutorial is to give you a glimpse of...

- some basic semantic web technologies (RDF, OWL)
- some underlying concepts like reasoning
- ... to give you some overview of the field
- ... as an entry point for further exploration

For more information:

# This Tutorial

The aim of this tutorial is to give you a glimpse of...

- some basic semantic web technologies (RDF, OWL)
- some underlying concepts like reasoning
- ... to give you some overview of the field
- ... as an entry point for further exploration

For more information:

- the W3C web pages contain the standards

# This Tutorial

The aim of this tutorial is to give you a glimpse of. . .

- some basic semantic web technologies (RDF, OWL)
- some underlying concepts like reasoning
- . . . to give you some overview of the field
- . . . as an entry point for further exploration

For more information:

- the W3C web pages contain the standards
- UiO lecture slides:

<http://www.uio.no/studier/emner/matnat/ifi/INF3580/v10/undervisningsplan.xml>

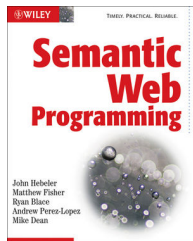
## Further Reading

- For practical aspects:

*Semantic Web Programming.*

Hebeler, Fisher, Blace, Perez-Lopez.

Wiley 2009



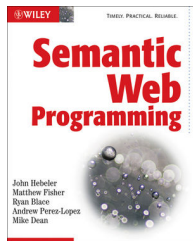
# Further Reading

- For practical aspects:

*Semantic Web Programming.*

Hebeler, Fisher, Blace, Perez-Lopez.

Wiley 2009

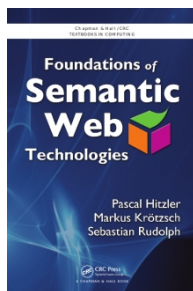


- For theoretical aspects:

*Foundations of Semantic Web Technologies.*

Hitzler, Krötzsch, Rudolph.

CRC Press 2009



# Downloads, Tutorial Web Page

Download and install software needed for the exercises:

- Protégé: ontology editor
- Java runtime environment
- D2R Server: “RDF server”
- D2RQueryEngine: Java program

Go to <http://sws.ifi.uio.no/semdays2010/> for more information.

Also contains:

- Required Software
- Source code
- Exercises
- Solutions