Welcome to INFO216: **Knowledge Graphs**

Andreas L. Opdahl
<Andreas.Opdahl@uib.no>

About me

- Background:
  - siv.ing (1988), dr.ing (1992) from NTH/NTNU
  - Univ. of Bergen since the early 1990-ies
  - part-time programmer for industry
  - consulting (enterprise and semantic modelling)
- Central research interest:
  - modelling of information systems and enterprises
  - several Forskningsråd projects and networks
- Semantic technologies:
  - semantics of modelling languages
  - Interop Network of Excellence (EU)
  - start-up on social semantic tagging (Lexitags)

Recent project: UBIMOB

Ubiquitous Data-Driven Urban Mobility

WESTERN NORWAY RESEARCH INSTITUTE
VESTLANDSFORSKING

NTNU
Norwegian University of Science and Technology

UNIVERSITAS BERGENSIS

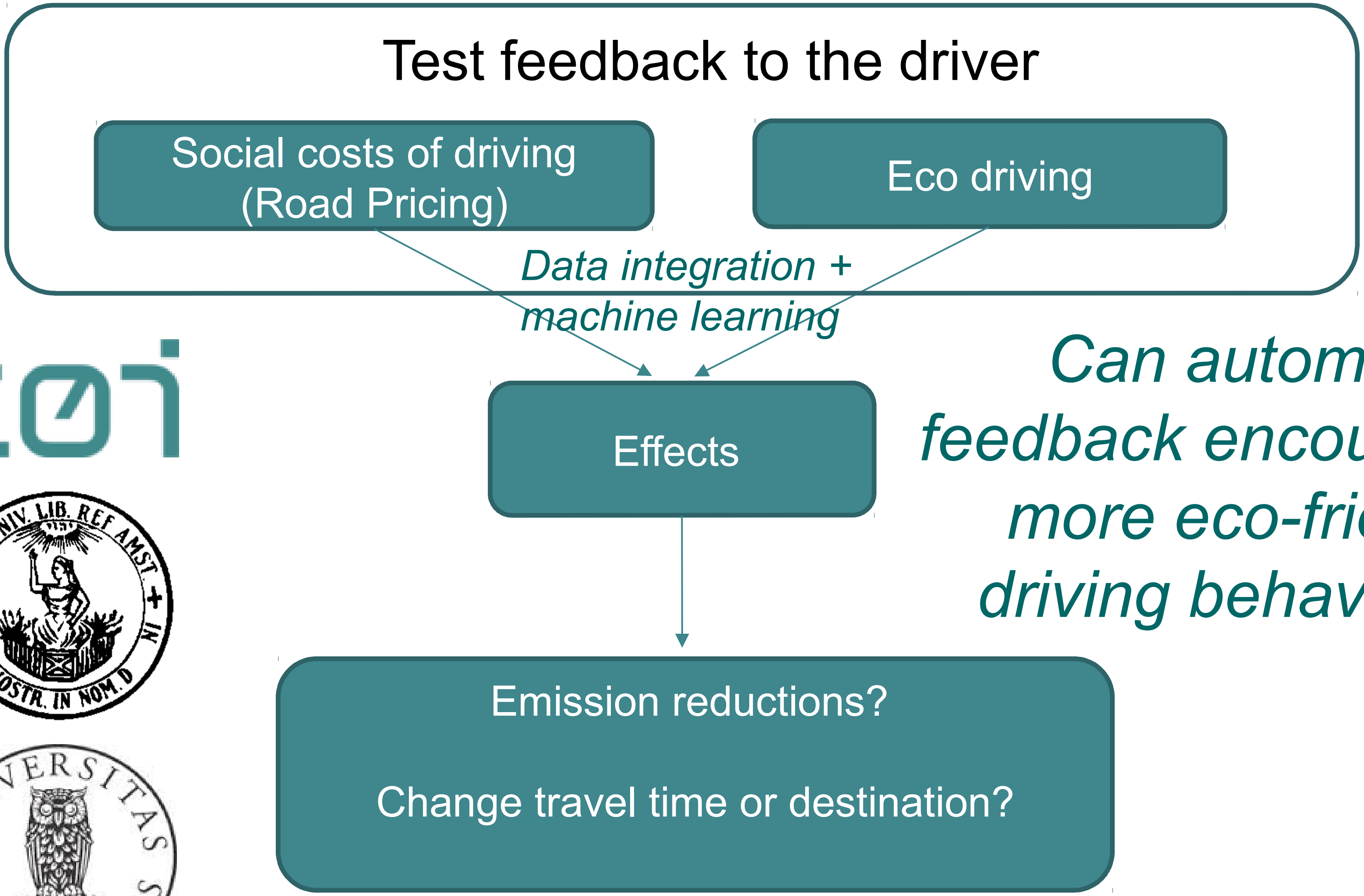
tøi
Transportøkonomisk institutt
Stiftelsen Norsk senter for samferdselsforskning

Forschungszentrum · Research Center
L3S

The University Of Sheffield.

telenor

Ongoing project: Transfeed



Test feedback to the driver

Social costs of driving (Road Pricing)

Eco driving

Data integration + machine learning

Effects

Emission reductions?

Change travel time or destination?

Can automated feedback encourage more eco-friendly driving behaviour?

Ongoing project: BDEM

- Leveraging *Big Data for Emergency Management*
  - how can semantic technologies play a part?
  - developing a new Master course: INFO319

NII 大学共同利用機関法人 情報・システム研究機構
国立情報学研究所
National Institute of Informatics

SAN DIEGO STATE UNIVERSITY

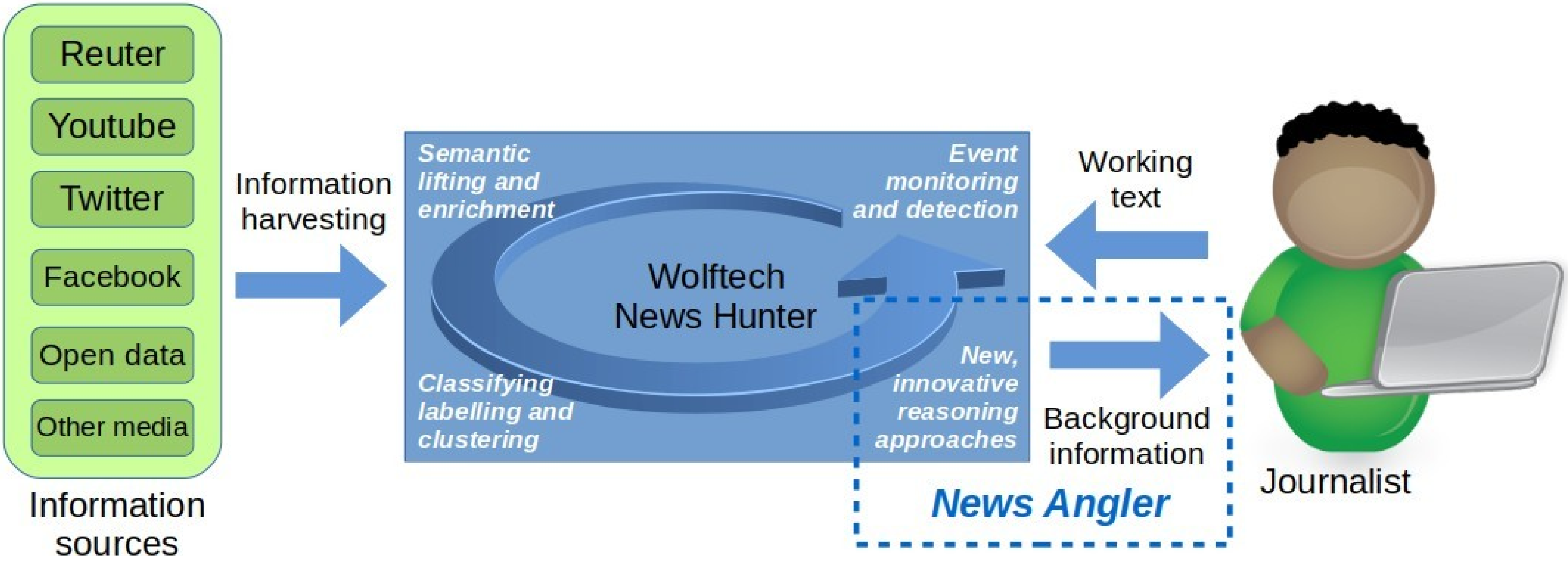
GEORGE MASON UNIVERSITY

to Advance Knowledge for Humanity · 1993 ·
THE UNIVERSITY OF AIZU

UNIVERSITAS BERGENSIS

VESTLANDSFORSKING

Ongoing project: News Angler



"Wolftech News supports and improves the workflows in a newsroom through mobile solutions for field work that are integrated with central systems for news monitoring, resource management, news editing, and multi-platform publishing"

1) Harvesting and analysing messages
2) Growing a semantic news graph
   - concepts, named entities, context…
3) Analysing working texts (stories)
4) Identifying background information
5) Prioritising and preparing
6) Journalistic and editorial preferences

Research: graph, searches, preparation, preferences, language, scaling

Lecture 1

- Themes:
  - *what are knowledge graphs?*
    - and what are semantic technologies?
    - ...picking up the thread from INFO116
  - *introduction to INFO216*
    - organisation of the course
    - practical information
  - *a little about programming langauge*
    - from Java to Python
    - a little about RDFLib
  - *(if we have more time: the programming project)*

Readings

- Sources:
  - Allemang & Hendler (2011):
    Semantic Web for the Working Ontologist
    chapters 1-2
- Detailed readings at http://wiki.uib.no/info216:
  - Tim Berners-Lee talks about the semantic web
  - stuff about RDFLib (for lab 1)
    - or Jena as an alternative

Knowledge Graphs

Transition

- From a *Web of Documents*
  - – today, the “plain old web” (PoW)
  - – document-centric
  - – document-to-document links
  - – for humans
- to a *Web of Data*
  - – the future “semantic web”, “Web 3.0”, “Linked Data”, “Web of Knowledge”
  - – document- *and data-centric*
  - – doc-to-doc *and data-to-data links*
  - – for humans *and machines*
- *AAA = Anyone can say Anything about Any topic*

Challenge

- There's an enormous amount of data on the web
  - ...but the data are mostly not linked
    (think of a world wide web without document links!)
  - availability, accessibility does not go all the way
  - *what if we had standard ways of representing data so that linkable data could always be automatically linked?*
  - *enormous potential to solve, simplify, speed up... many critical information handling problems*
- This is the purpose of *semantic technologies*
- This is the vision that lead to *knowledge graphs*

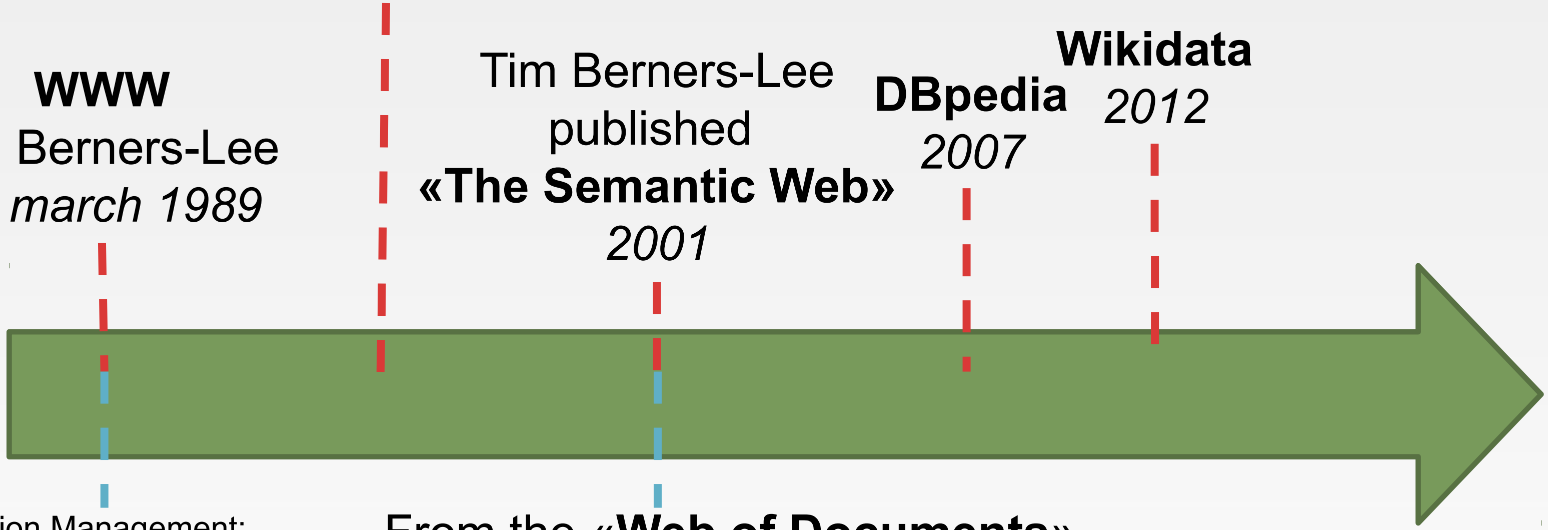
Tim Berners-Lee: <http://www.youtube.com/watch?v=HeUrEh-nqtU>

Semantic web and WWW history

Weaving the Web (1999)

The original design and ultimate destiny of the World Wide Web, by its inventor

https://www.w3.org/People/Berners-Lee/Weaving/Overview.html



WWW
Tim Berners-Lee
12 march 1989

Tim Berners-Lee
published
«The Semantic Web»
2001

DBpedia
2007

Wikidata
2012

Information Management:
A Proposal
Tim Berners-Lee, CERN

From the «**Web of Documents**»
to the «**Web of Data**»

Tim Berners-Lee: http://www.youtube.com/watch?v=HeUrEh-nqtU
Information Management: A Proposal: https://cds.cern.ch/record/369245/files/dd-89-001.pdf

Not a single coordinated effort...

- A *family of developments*:
  - Semantic Web, Web of Data, contextual web: making the web data-oriented, semantic, machine-processable (around 2000)
  - microformats, Microdata: weaving facts, semantics, and small RDF graphs into HTML pages
  - semantic technologies: reusable technologies and tools for handling semantic data: RDF, SPARQL, OWL...
  - social tagging: large-scale semantic tagging produced by social media (around 2005)

Not a single coordinated effort...

- A *family of developments*:
  - Linked Open Data (LOD) cloud:
    interlinking semantic datasets, making them openly available: DBpedia, Wikidata… (around 2007)
  - company-internal semantic data:
    linked open data and semantic technologies used inside an enterprise or cluster
  - *knowledge graph:*
    general term for semantic graph representations of (primarily) factual information, may not use RDF (from 2012)
  - Giant Global Graph (GGG):
    global interlinking of open knowledge graphs (≈ LOD)

Common themes

- Semantically tagged data
- Well-defined tags (terms)
  – defined in standard vocabularies
  – formal ontologies, description logic
- Graph representations of knowledge
  – RDF, RDFS
  – more recently: labelled-property graph databases
- Standard exchange formats
  – (re-)using the same APIs, etc.
- Open, community-based
  – ...(re-)using many of the *same technologies*

So what are semantic (-ally tagged) data?

- *Metadata* are data about other data (actually *information*)
  - e.g., data about the format and language of a web page
- *Semantic metadata* are data about the meaning of other data
  - e.g., data about the meaning of each table and column in a relational database
- *Semantic data* (or *semantically tagged data*) are data supported by semantic metadata
  - or: semantic data are data supported by metadata about their meaning
  - e.g., the above relational database along with the data about its meaning

How can we *represent* meaning?

- Only in part!
- Meaning a *complex concept* with several *levels*: semantics, pragmatics, social...

- *Vocabularies* can capture certain aspects of meaning:
  - standard IRIs for *types of resources*
  - standard IRIs for *properties*
  - standard types for *literals*
  - *rules* about how they combine
- Other *open semantic datasets* define:
  - standard IRIs for *individual resources*

How to represent semantic data?

- Can in principle be represented on many formats
- The Web of Data relies heavily on the *Resource Description Framework (RDF)*
  - a "normal form" for semantic data
  - can be used to represent "knowledge graphs"
  - used both for the data and their metadata
  - either *native/reified*, *embedded, or virtual*
- More expressive vocabularies are available using
  - *RDF Schema (RDFS), "RDFS Plus"*
  - *Web Ontology Language (OWL)*
  - (can be said to) *build on RDF*

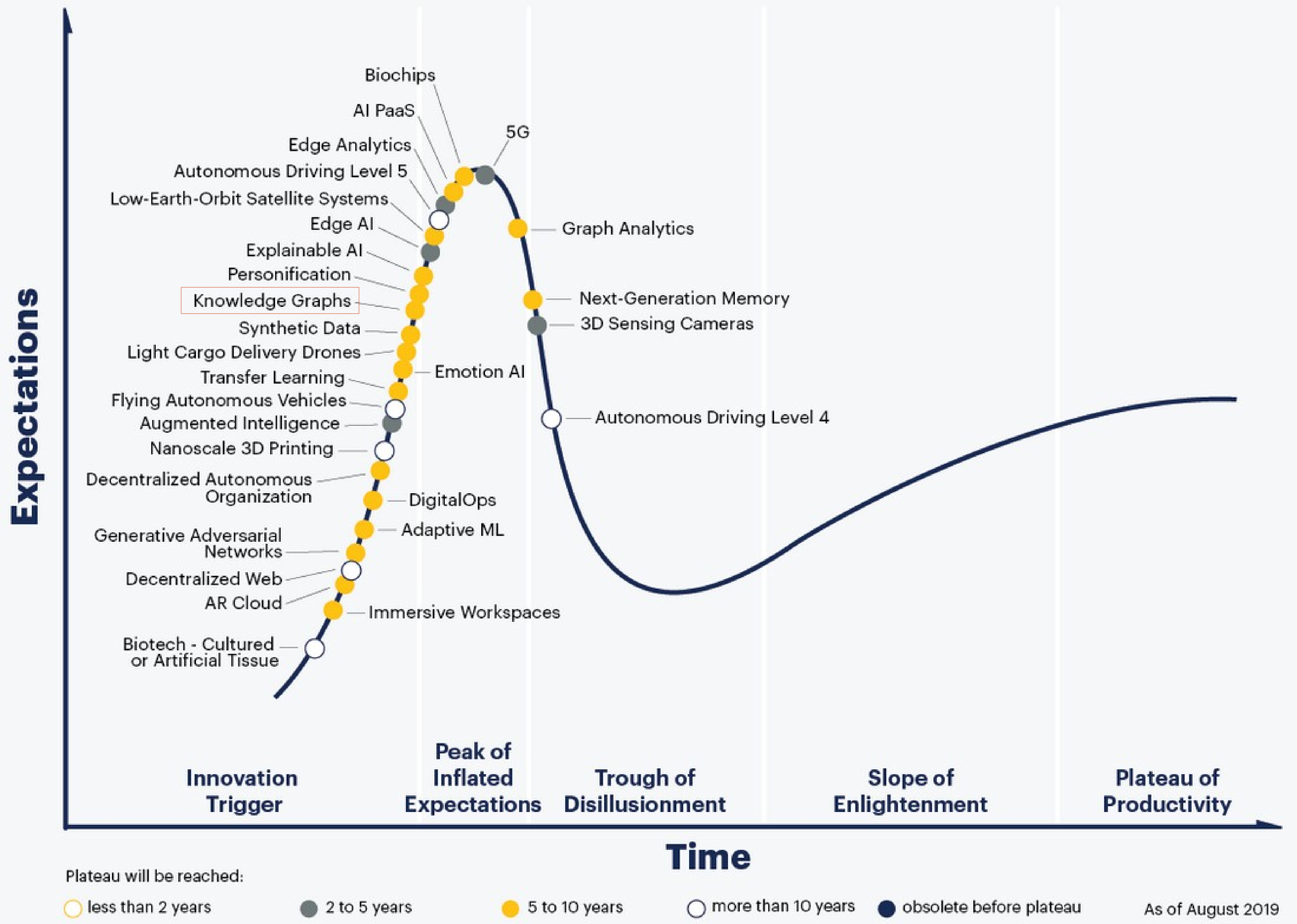
Resource Description Framework

- Represents data as triples (“statements”):
  - *(subject, predicate, object)*
  - *subject:*
    - represents what the statement is about
    - the IRI of a semantic resource
  - *predicate:*
    - represents a property of the subject resource
    - the IRI of a semantic property
  - *object:*
    - represents the value of a property for a subject
    - either: the IRI of a semantic resource
    - or: a literal (number, string, boolean...)

Semantic graphs and data sets

- *Graph*:
  - a collection of *triples/statements* (possibly none)
  - *"knowledge graphs"*
- *Data set (or "Conjunctive graph")*:
  - a collection of graphs (at least one)
  - one of the graphs is *default/unnamed*
  - the others are *named*
  - from triples/statements:
    - *(subject, predicate, object)*
  - to quadruples *(quads):*
    - *(graph/"context", subject, predicate, object)*

Knowledge graphs are everywhere!


Expectations
Biochips
AI PaaS
5G
Edge Analytics
Autonomous Driving Level 5
Low-Earth-Orbit Satellite Systems
Edge AI
Graph Analytics
Explainable AI
Personification
Knowledge Graphs
Next-Generation Memory
Synthetic Data
3D Sensing Cameras
Light Cargo Delivery Drones
Emotion AI
Transfer Learning
Flying Autonomous Vehicles
Autonomous Driving Level 4
Augmented Intelligence
Nanoscale 3D Printing
Decentralized Autonomous Organization
DigitalOps
Adaptive ML
Generative Adversarial Networks
Decentralized Web
AR Cloud
Immersive Workspaces
Biotech - Cultured or Artificial Tissue
Innovation Trigger
Peak of Inflated Expectations
Trough of Disillusionment
Slope of Enlightenment
Plateau of Productivity
Time
Plateau will be reached:
○ less than 2 years
● 2 to 5 years
● 5 to 10 years
○ more than 10 years
● obsolete before plateau
As of August 2019

And many others...

- BBC’s content management, ontologies, BBC Things
- Google, Bing, Yahoo… (schema.org) (2011)
- Google’s Knowledge Graph (2012), Microsoft’s Satori
- Facebook’s Open Graph and Graph Search (2013)
- Thomson Reuters, Bloomberg...
- Amazon’s Product Graph (2017), Neptune
- Uber Eats’ food graph

Frank van Harmelen’s keynote at CAiSE 2018.

Organisation of the course

Curriculum

- Mandatory:
  - textbook:
    Allemang & Hendler: Semantic Web for the Working Ontologist, 2nd ed. 2011
  - lectures and lecture notes
  - electronic materials in the wiki (wiki.uib.no/info216)
    - introductions, tutorials
    - standards documents
    - academic papers
- Cursory:
  - further materials in the wiki (wiki.uib.no/info216)

Lectures and labs

- 15 lectures:
  - most Thursdays 1215-1400 ***(next week is special)***
  - mostly theory
    - maybe some workshop-style parts
- 15 lab sessions (Friday and Wednesday), ***starting tomorrow***:
  - lab leader: Martin Eidsvik Torvanger <Martin.Torvanger@student.uib.no>
  - 3 lab days used for project presentations/discussions:
    - weeks 7-8, 12-13, and 17-18
  - the rest are practical assignments
  - 80% mandatory, including *all the presentation days*
- Also seminar/question sessions (Mondays)

Theory lectures (tentative)

1. Knowledge graphs
2. RDF
3. SPARQL
4. Architecture
5. RDFS
6. RDFS Plus
7. Vocabularies 1
8. Vocabularies 2
9. Linked Open Data 1
10. Linked Open Data 2
11. Web APIs 1
12. Web APIs 2
13. OWL
14. OWL DL
15. Open / ontology development

You learn programming (mostly) through the lab exercises and project!

Lab exercises (tentative)

1. Getting started
2. RDF programming
3. SPARQL
4. Storing graphs
5. *Project presentations*
6. SPARQL programming
7. SPARQL Update
8. Client-side presentation
9. Web APIs & JSON-LD
10. *Project presentations*
11. Protege-OWL
12. OWL programming
13. Reasoners
14. (Open)
15. *Project presentations*

You learn programming (mostly) through the lab exercises and project!

Evaluation

- Two-part evaluation:
  - individual, written 3-hour exam (60%)
  - group assignment/programming project (40%)
- Exam requirements:
  - submitted programming project
  - participation in 80% of labs

Programming project

- Mandatory programming project submitted in May:
  - groups:
    - 3 people recommended, 1-2 ok, *never 4 or more*
  - counts 40% of course grade
    - idea meetings with me
      - week 5-6: January 31st and February 5th
    - mandatory presentations in the labs
      - week 7-8: February 14th and 19th
      - week 12-13: March 20th and 25th
      - week 17-18: April 24th and 29th
  - *the wiki outlines project types and criteria*

Programming project

- The programming project shall develop a semantic/linked data application. Development and run-time platform is free choice, as is programming language.
- The project should be carried out in groups of three and not more. Working individually or in pairs is possible, but not optimal. Groups of more than three will not be accepted.
- The application will be presented in the seminar groups, and each group member will describe their contribution to the finished product. The assignment must be done in the teaching semester.
- ***...more about that later!***

Programming RDF (and RDFS, SPARQL...) with Python

RDFLib

- A library and API (Application Programming Interface) for programming RDF and SPARQL in Python
  - simple, powerful and *pythonic*
  - parsers and serialisers for most RDF formats
  - a *Graph* interface
  - with multiple alternative *Stores*
  - SPARQL 1.1 Query and Update
- Other technologies later:
  - a triple store (RDF database), most likely *Blazegraph*
  - an OWL library for Python, most likely *owlready2*

RDFLib interfaces

- Graph:
  - an RDF model
  - a Python collection (set) of triples
  - adding, removing, listing, and searching for triples
  - combine with other graphs
  - writing to and reading from RDF files
  - responding to SPARQL queries and updates
  - backed by an in-memory or persistent store
- Dataset / ConjunctiveGraph:
  - multiple named RDF models in a dataset
  - a set of quads: triples with graph ids

RDFLib interfaces

- Triples or statements:
  - ordinary 3-item Python tuples
    - immutable sequences
    - >>> triple = (s, p, o)   # creates a triple
    - >>> s[0]   # returns the subject…
- URIRef:
  - a node with a IRI
- BNode:
  - a blank node with a Graph-internal identifier only
- Literal:
  - a typed or untyped value
  - untyped values (strings) can be language-tagged

RDFLib interfaces

- Namespaces:
  - predefined RDF, RDFS, OWL, XSD, FOAF, SKOS, DC, DCTERMS
  - user-defined namespaces, e.g.:

```
>>> i2s = Namespace(‘http://i2s.uib.no/’)
>>> i2s.MainAuthor
rdflib.term.URIRef(u'http://i2s.uib.no/MainAuthor’)
```

Programming project

Past projects

- Example projects:
  - *make your own muncipalities*
  - *map of party financing*
  - *reasoning over toll roads*
  - *social assessment network*
  - *LinkedMDB-portal*
  - *tracking IT infrastructure*
  - *music concert assistant*
  - *quiz generator*
  - *live semantic flight data*
  - *semantic security service*

Success factors

- Show that you can program with semantic technologies
  - *at least* RDF, *preferred* RDFS, SPARQL, ...
  - ...JSON-LD is an *emerging alternative*
- Use existing data sets (open semantic resources)
- Use existing vocabularies (and perhaps extend them)
- *Simple* presentation interface / dashboard
- Make the program run :-)
- *Shortcuts can be ok* (some manual steps, artificial data)
- *Try to have an original idea*

Example: combination projects

- Take two or more (semantic?) data sets
- Read them
- If necessary: lift them (i.e.: add semantic tags)
- Combine the data sets semantically
- Use them to derive new data/answer new queries
  - impossible to answer before
  - harder to answer before
- Mantainability:
  - what happens when the data sets change?
- *Dynamic data sets are more interesting that static ones!*

Example: lifting projects

- Take a data set or a Web API (web service)
- Read it / access it over the net
- Lift it (i.e.: add semantic tags)
  - using existing vocabularies as far as possible
- Show and implement use cases
  - that were impossible before
  - that were harder before
  - that were less flexible before
- *Focus on maintainability – making it easy run over time!*

Other projects are very possible!

- Combination and lifting projects are the most common
- Other types are very possible, e.g.:
  - semantic crawlers and spiders
  - presentation / visualisation of graphs
- *You are free to propose (almost) anything!*

- How big should my project be?
  - usually not a problem
  - always possible to narrow the scope
  - usually possible to expand the scope
  - a bit easier to start "too big" than "too small"

Expectations to first meeting (31/1, 5/2)

- Alone or in groups of 2-3
  - not plenary the first time
  - first talk to Martin in the labs, then with me (room 609)
- Which data sets will you use?
- Which vocabularies will you use?
- What will you use them for?
  - something that cannot be done today
  - something that is harder to do today
  - something that is harder to do flexibly today
- You may bring several alternatives
  - but make sure you have a clear favourite