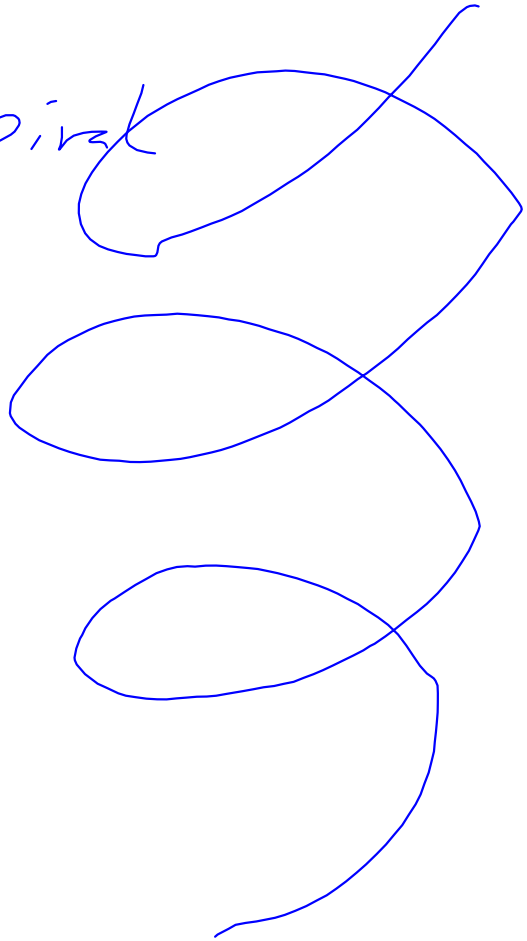
Final Exam 60%

- Presnt Group Work
- Questions to the Group work
- "Random learning outcomes"

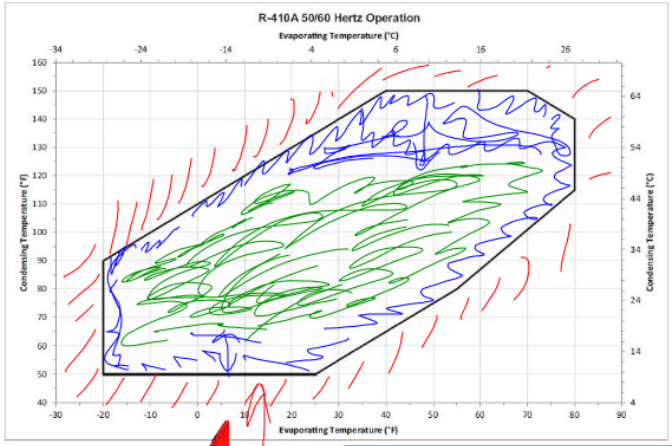
New date (?)
2 Jun 2016

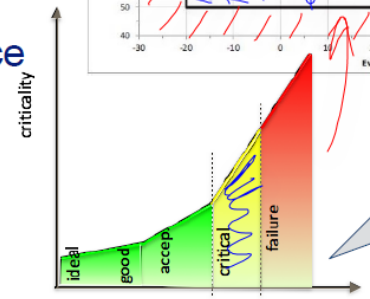
Security spirit



Recap:
Conversion and operating envelope

- Operating envelope: the operational parameters where our network can work “well”, depends on the technology and on the task
- For traffic estimation we need it in ”communication” QoS
  - ➡ Bandwidth, delay, jitter, (redundancy)
- Often can be done with simple arithmetic with a certain confidence level





1) How does he Operating Envelope look like applying criticality?
2) How can the criticality be applied for SPD?

Examples of
Security challenges in the IoT

- **System**: Intrusion awareness, fault-tolerance, data redundancy and diversity
- **Platform**: Auto start up on power failure, Auto reconfigurable on software failure, Auto synchronization on software failure, End-to-end secure communication, Mal-user detection, Access control for accessing sensor data
- **Middleware**: SPD Audit, Cryptographic Support, Identification and Authentication, Protection of the SPD functionalities, Security Management
- **Hardware**: SPD metrics, Self–recovery from hardware transient faults (through fault–injection), Auto–reconfiguration, Data encryption, Provision of security and privacy services, data encryption/decryption
- **Radio**: Threats tolerant transmission

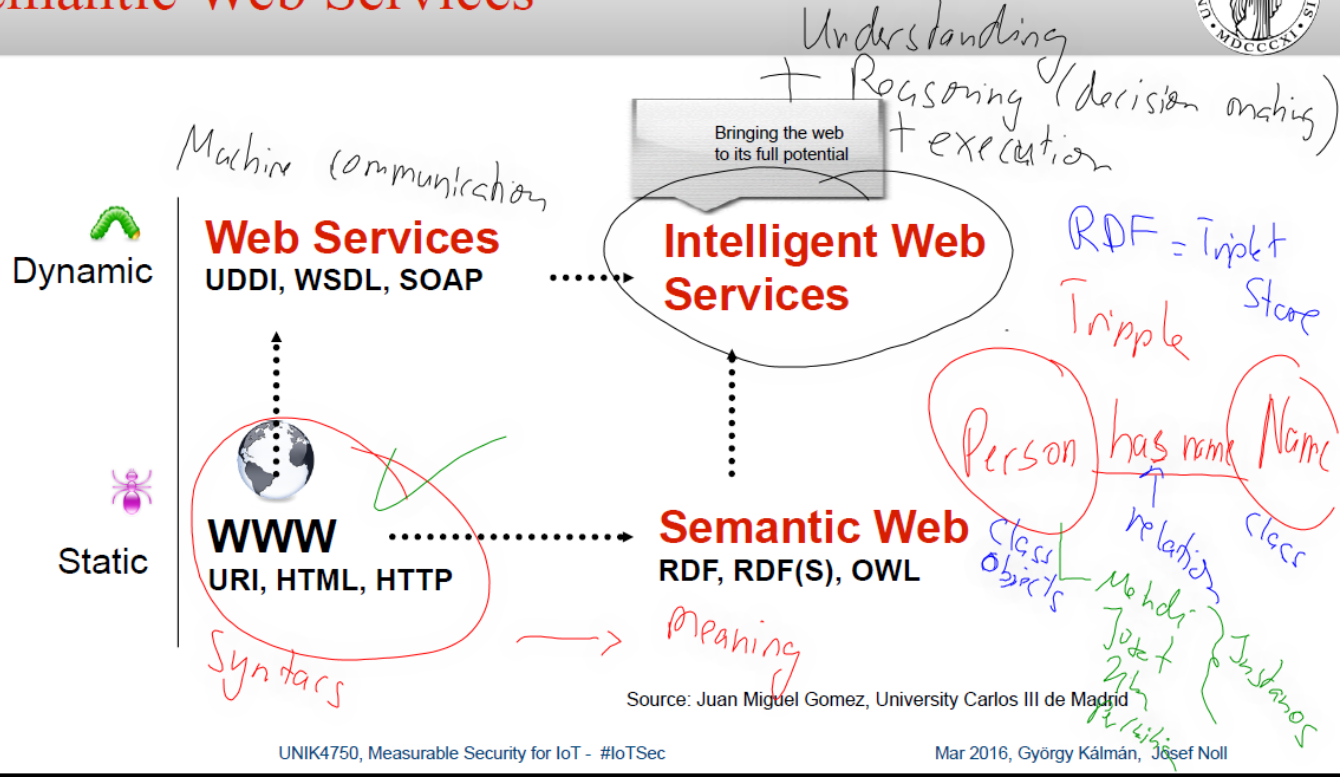
System components
classified after objective

- Functional components
  - input component (sensors, keyboard, mouse,..)
  - output component (alarm, screen, actuator,..)
  - processing component
  - Storing component (data base, files,)
  - Connection (wireless connection, wired connection)

- Security, Privacy, Dependability (SPD) components:
  - Encryption: Encryption algorithm, keys,..
  - Protocols
  - Authentication(mechanism (fingerprint, password, password complexity,.....) .
  - Authorization (privileges, ..) → functionality & component (door lock)

- Management components (OS, Web server, data server)
- Human component (admin, user, ..).
- Physical component, car being a component in a car factory. (if treated as “sub-system)

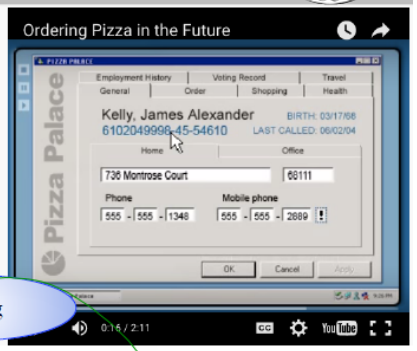
Semantic Web Services



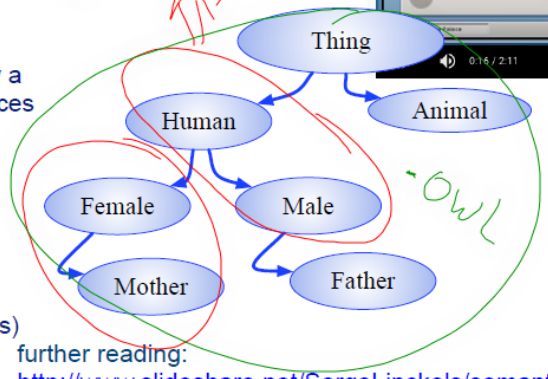
Source: Juan Miguel Gomez, University Carlos III de Madrid

Elements in Semantic Technologies

- Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format which is both human-readable and machine-readable. [Source: Wikipedia]
- RDF - Formal semantics is built upon a W3C XML standard for objects called the Resource Description Framework (RDF)
- OWL - The Web Ontology Language (OWL) is a family of knowledge representation languages for authoring ontologies.
- A semantic reasoner, reasoning engine, rules engine, or simply a reasoner, is a piece of software able to infer logical consequences from a set of asserted facts or axioms.
- Classes (concepts) are abstract groups, sets, or collection of objects (example: human, woman)
- Individuals (instances) are the specific objects, e.g. Josef is a Father
- Attributes (properties) describing objects (individual and classes) in the ontology. Example: Human hasName, Josef has name Josef Noll



TRIPLETS (triple store)



further reading:
http://www.slideshare.net/SergeLinckels/semantic-web-ontologies

```
{{User
|Prename=Bjørnar
|Name=Dragnes
|Email=bjdragnes@mil.no
|Partner=MIL,
}}
```

Semantics Info user

Semantics about Project

```
|Project leader=Terje Tjelta
|Project Participant=Bjørn Andersen, Bjørnar Dragnes, Herman Hansen, Jan Erik Håkegård, Jan Hetland, Jos
Rytir, Michael Cheffena, Michal Ciecko, Per Arne Grotthing, Terje Medby, Terje Mikal Mjelde, Trond Henning
```

Reason

who is project participant from Partner MIL

```
{{#ask: [[-Project Participant.Project::SAT]] [[Partner::MIL]] |intro=Project members of NDLO: &#10;  }}
```

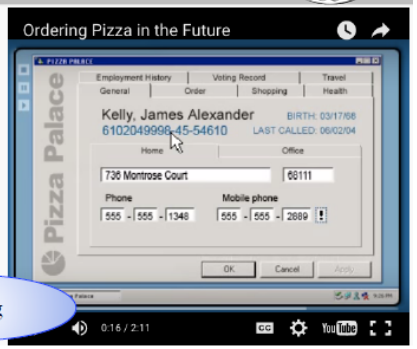
Result

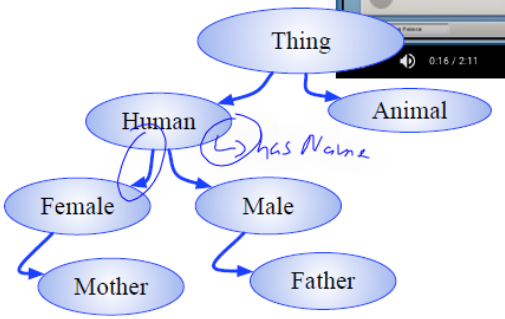
Project members of NDLO: Bjørn Andersen, Bjørnar Dragnes, Trond Henning Johansen

Elements in Semantic Technologies

- Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format which is both human-readable and machine-readable.
- RDF - Formal semantics is built upon a W3C XML standard for objects called the Resource Description Framework (RDF)
- OWL - The Web Ontology Language (OWL) is a family of knowledge representation languages for authoring ontologies.
- A semantic reasoner, reasoning engine, rules engine, or simply a reasoner, is a piece of software able to infer logical consequences from a set of asserted facts or axioms.
- Classes (concepts) are abstract groups, sets, or collection of objects (example: human, woman)
- Individuals (instances) are the specific objects, e.g. Josef is a Father
- Attributes (properties) describing objects (individual and classes) in the ontology. Example: Human hasName, Josef has name Josef Noll

[Source: Wikipedia]





further reading:
http://www.slideshare.net/SergeLinckels/semantic-web-ontologies