Semantic Web of Things (SWoT)
An introduction

Gérald Rocher, 02/02/2016
Internet of Things (IoT)
Internet of Things (1/3)

▸ **Physical** things connected to **Devices**

▸ A Device provides access to **resources**
  ▸ **Sensors**,  
  ▸ **Actuators**.

▸ Devices can be...
  ▸ **Physically coupled** sensor & actuators  
    ▸ Heater (resistor + temp sensor).
Internet of Things (2/3)

- An hardware layer to connect devices to the internet through *communication protocols*...
  - Devices *accessibility*,
  - Devices *unique identification*.

- As...
  - Network of networks (gateways between heterogeneous protocols),
  - Things over internet (gateways toward a common protocol).

- Devices can also be...
  - *Logically coupled* sensor & actuators
    - Switch connected to a light.

Internet of Things (3/3)

▶ First Interoperability issue...

▶ Technological heterogeneity

▶ Internet is a common network and transport protocol to interconnect devices but still numerous ways to control devices and retrieve their data... that derives from the vast amounts of heterogeneous objects)

▶ Does not ensure interoperability between devices...
Web of Things (WoT)
Introduction

- Approaches, software architectural styles and programming patterns that allow devices to be part of the World Wide Web...

- Dominique Guinard -- PhD thesis (2011)
  - Four layers IoT common application architecture
    1. Accessibility,
    2. Findability,
    3. Sharing,
    4. Composition.

A Web of Things Application Architecture - Integrating the Real-World into the Web.
PhD thesis No. 19891, ETH Zurich, Zurich, Switzerland, August 2011
Accessibility Layer

- **Resource Oriented Architecture (ROA)**
  - RESTful services & description (RSDL)
  - data centric: over HTTP (GET, POST, PUT, DELETE, etc.)
- **Service Oriented Architecture (SOA)**
  - SOAP services → WS-* (functional control),
  - API descriptions (WSDL).

- A common & generic way to access devices data and API through gateways
  - Devices technological abstraction,
  - Fixing the IoT Technological Heterogeneity issue.

- Once devices are accessible to the web, web applications and tools can be used...
Provides a way to **find** and **locate** relevant services (devices) on the Web

- Search engines,
- Crawlers,
- Etc...

**Metadata model** for describing smart things and their services...

- Integration/indexation of smart things to existing search engines,
- **Semantic annotations** based on microformat/RDFa to describe static properties (product, service) and dynamic properties (Location, Quality of Service).

https://fr.wikipedia.org/wiki/Microformat
Sharing Layer

- Ensure data generated by devices are shared in an efficient and secure manner.
  - Security (encryption, authentication)
  - JWT, API tokens
  - Etc...
Integrate the services and data offered by the devices into higher level Web tools:

- Analytics software,
- Physical mashups, composite applications,
- Etc...
What’s the problem?

- Second interoperability issue...
  - Semantic heterogeneity
    - Numerous devices, manufacturers, ways to describe Devices, services and data...
    - Numerous data sources and types...
  
- How to describe data to get it **machine understandable** and establish **collaboration** among devices (full interoperability) far beyond microformat, RDFa, ...?
Problem illustration (1/2)

Sensor data

37.8°C

Sensor data

160.0°C

Sensor data

73.0°F

How machine can interpret data semantics?
Problem illustration (2/2)

Looking for a display...

How machine can interpret devices functionality?

WSDL, RSDL, etc... are syntactic descriptions; microformat/RDFa semantics descriptions are limited...
Semantic Web of Things (SWoT)

Reuse **Semantic Web standards** to explicitly (formally) describe things, devices, their services and the data they publish... with semantically enriched annotations

**In a nutshell:**
- Knowledge formal description (RDF, RDFS, OWL),
- Enables Machine to Machine (M2M) communication.
- Reasoning (Inference Rules),
- Querying (SPARQL).

[https://www.w3.org/standards/semanticweb/](https://www.w3.org/standards/semanticweb/)
Resource Description Framework (RDF)

RDF triple (subject, predicate, object)
- Labelled connection between two resources,
- s, p are URI (unambiguous),
- o is URI or literal,
- p states the relationship between s and o.

RDF triples are directed labelled graph.

Minimum vocabulary to describe the knowledge (taxonomy)

Without restrictions / conditions...

RDFS & OWL family languages allows more expressivity needed to build ontologies...
Ontology (aka vocabulary)

- Formal knowledge description of a domain:
  - Class (concept), Class hierarchy,
  - Properties, instances,
  - Restrictions, etc...

- A consensual knowledge to be reused and shared across applications and people.

Knowledge expressivity depends on the description language used (RDFS, OWL-LITE, OWL-DL, OWL2, etc...).
Reuse existing ontologies as much as possible

SSN-XG (Semantic Sensor Network ontology)
https://www.w3.org/2005/Incubator/ssn/ssnx/ssn

Part of the Linked Open Vocabularies (LOV) effort
http://lov.okfn.org/dataset/lov/

Up to 533 vocabularies to date...
Knowledge base (KB)

- Stores intentional (Tbox ➔ ontology) and assertional knowledge (Abox ➔ instances),
- An Inference engine that can reason about the knowledge and use rules and logic to deduce new knowledge or detect inconsistencies.
- A query engine (SPARQL) to retrieve, add, remove RDF data from the KB.
Ontology modeling layers

**Meta meta model**
(Ontology modeling languages)

**Meta model**
(Specific domain ontology, aka upper ontology)

**Model (Device annotation)**
(Assertions about the world)

---

Subject \( \rightarrow \) Predicate \( \rightarrow \) Object

rdfs:Resource

rdfs:domain

rdfs:domain

rdfs:domain

rdfs:domain

Appliance

Oven

IQ700

manufacturer

has_manufacturer

rdfs:domain

rdfs:domain

rdfs:domain

rdfs:domain

rdf:type

rdf:type

rdf:type

rdf:type

rdf:type

rdf:type

rdf:type

rdf:type

rdf:type

http://example.org/my_ontology

http://example.org/my_ontology

http://example.org/my_ontology

http://example.org/my_ontology

http://example.org/my_ontology

http://example.org/my_ontology

http://example.org/my_ontology

http://example.org/my_ontology
Ontology modeling (main) approaches in SWoT

**Global approach**: each device relies on a common upper ontology to describe its instance. The KB is enriched over time and contains only the needed knowledge.

**Heterogeneous approach**: each device relies on its own ontology to describe its instance. The KB is enriched over time with new meta models.

**Fragmented upper ontology approach**: each device relies on fragments of a common upper ontology to describe its instance. The KB is enriched over time and contains only the needed knowledge.
Semantic Web standards basics...

**Online resources**

- [https://www.w3.org/standards/semanticweb/](https://www.w3.org/standards/semanticweb/)
- [http://www.inria.fr/centre/sophia/actualites/mooc-web-semantique-et-web-de-donnees](http://www.inria.fr/centre/sophia/actualites/mooc-web-semantique-et-web-de-donnees)
- [http://www.dcs.bbk.ac.uk/~michael/sw/sw.html](http://www.dcs.bbk.ac.uk/~michael/sw/sw.html)
- Etc...
SWoT Applications
SWoT Application domains...

- Smart Home,
- Smart Office,
- Smart Cities,
- Building automation,
- Assisted living,
- Smart Energy,
- Farming,
- Sensor networks,
- Etc...

The Internet of Things vision: Key features, applications and open issues, Eleonora Borgia Institute of Informatics and Telematics (IIT), Italian National Research Council (CNR), via G. Moruzzi 1, 56124 Pisa, Italy Computer Communications 54 (2014) 1-31, paper
Interconnecting devices and gather data from them allows to create **context aware** (aka situation) applications:

- Better understanding their surrounding environment,
- Making intelligent decisions and better **reacting** to the **dynamics** of their environment.

---

SWoT for context awareness (2/4)

**Application objectives**

- Display movie in the livingroom
- Display movie for Bob
SWoT for context awareness (3/4)

Upper ontology

- Device
  - Display
    - TV
    - Tablet

- Location
  - Kitchen
  - Livingroom

- User
  - Bob

- Entity
  - Domain
    - has_location

- Device
  - has_location

- Display
  - rdf:subClassOf

- Livingroom
  - rdf:type

- Kitchen
  - rdf:type

- TV
  - rdf:type

- Tablet
  - rdf:type
SWoT for context awareness (4/4)

Application objectives
- Display movie in the livingroom
- Display video for Bob

System

Appearence...

Disappearence...

Semantic annotation
SWoT for services composition

- Services functionalities & context understanding enables applications to be composed using relevant services.
  - Self-adaptive systems,
  - Continuity of service.

- Solutions have been developed, allowing to semantically describe web services and ease their discovery, composition and invocation:
  - OWL-S (Semantic Markup for Web Services)
    http://www.daml.org/services/owl-s/1.0/owl-s.html
  - SAWSDL (Semantic Annotations for WSDL)
    https://www.w3.org/TR/sawsdl/
  - WSDL-S, WSMO, and more...

Standard vocabulary to semantically describe services

Service profile
- Describes what the service does, its name, description, the quality of service (QoS), etc...
- Primary meant for human reading...

Process Model
- Describes how a client can interact with the service.
- Input, Output, Pre-conditions, Effects/Results (IOPE/IOPR).

Service Grounding
- Information needed to interact with and instantiate the service,
- Communication protocol, Message format, Port number, Etc.
SAWSDL

- OWL-S ontology leaves out domain specific objects, and also Groundings for other service technologies like UPnP...

- SAWSDL...
  - Semantic annotations for WSDL components (Web Service Description Language),
  - Allows references to domain specific ontologies.
Semantic services composition approaches

► **Planning techniques**
  ► Problem of finding and aggregating a series of services with compatible IOPR/Ess allowing to reach the desired goal.
  ► Multi-objective Quality of Service (QoS)

► **Model-driven**
  ► Defining a high level abstraction (model of the target application),
  ► **Matching/service selection** approach to iteratively check each service, trying to match with the required functionality.
Looks good! What’s wrong???
That (will) most likely develop their own ontologies to describe their devices...

The development of a comprehensive ontology describing the world is unlikely to happen...
Numerous ontologies available, targeting different domains...

http://www.sensormeasurement.appspot.com/?p=ontologies references up to 296 different ontologies...
Actually, most of the current SWoT applications rely on ad-hoc ontologies that cannot be reused → It works but is application/domain specific.

Why it is a problem?

Knowledge base enrichment over time (1/4)

- SWoT applications have to integrate heterogeneous models & meta models unknown @design-time...

- Ontology merging
Ontology alignment & Mapping

Several algorithms available...

- Syntactic alignment,
- Structural alignment

Alignments are not perfect and depend on a pre-defined threshold value...


http://www.sensormeasurement.appspot.com/?p=ontoMappingTool
Knowledge base enrichment over time (3/4)

▶ Summary

- **Validity over time?**
- **Inconsistencies management?**

- **New facts, integrity checking, etc...**
- **Alignment (syntactic, structural) fixed threshold value**
- **Ontology A**
- **Ontology B**
- **Correspondences**


- **Meta model**
- **Discovery**
- **Reasoning**
- **Merging & Integration**
- **Alignment & Matching**
- **Knowledge Base**
- **!!! KB content management !!!**

- **REMOTE**
- **SWITCH**
- **MERGE**
- **INTEGRATE**
- **DISCOVER**
- **RUNTIME**
### Knowledge base enrichment over time (4/4)

#### Example

<table>
<thead>
<tr>
<th>Location</th>
<th>Device</th>
<th>Classes</th>
<th>Axioms</th>
<th>Degradation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home</td>
<td>Boiler</td>
<td>100</td>
<td>453</td>
<td>0%</td>
</tr>
<tr>
<td>Home</td>
<td>Clock</td>
<td>13</td>
<td>69</td>
<td>43.44%</td>
</tr>
<tr>
<td>Home</td>
<td>Computer</td>
<td>24</td>
<td>124</td>
<td>0%</td>
</tr>
<tr>
<td>Home</td>
<td>Cooker</td>
<td>48</td>
<td>109</td>
<td>73.28%</td>
</tr>
<tr>
<td>Home</td>
<td>DeepFreezer</td>
<td>48</td>
<td>105</td>
<td>76.87%</td>
</tr>
<tr>
<td>Home</td>
<td>DishWasher</td>
<td>38</td>
<td>110</td>
<td>75.22%</td>
</tr>
<tr>
<td>Home</td>
<td>Fan</td>
<td>24</td>
<td>124</td>
<td>0%</td>
</tr>
<tr>
<td>Home</td>
<td>Oven</td>
<td>109</td>
<td>489</td>
<td>0%</td>
</tr>
<tr>
<td>Home</td>
<td>Printer</td>
<td>24</td>
<td>124</td>
<td>0%</td>
</tr>
<tr>
<td>Shop</td>
<td>CoffeeMaker</td>
<td>24</td>
<td>124</td>
<td>0%</td>
</tr>
<tr>
<td>Shop</td>
<td>Computer</td>
<td>13</td>
<td>58</td>
<td>53.22%</td>
</tr>
<tr>
<td>Shop</td>
<td>DeepFreezer</td>
<td>100</td>
<td>454</td>
<td>0%</td>
</tr>
<tr>
<td>Shop</td>
<td>Entertainment</td>
<td>11</td>
<td>30</td>
<td>75.80%</td>
</tr>
<tr>
<td>Shop</td>
<td>Fan</td>
<td>2</td>
<td>4</td>
<td>96.77%</td>
</tr>
<tr>
<td>Shop</td>
<td>Fridge</td>
<td>44</td>
<td>73</td>
<td>85.45%</td>
</tr>
<tr>
<td>Shop</td>
<td>Printer</td>
<td>11</td>
<td>49</td>
<td>60.48%</td>
</tr>
<tr>
<td>Friend</td>
<td>Clock</td>
<td>24</td>
<td>122</td>
<td>0%</td>
</tr>
<tr>
<td>Friend</td>
<td>Computer</td>
<td>2</td>
<td>4</td>
<td>96.77%</td>
</tr>
<tr>
<td>Friend</td>
<td>Cooker</td>
<td>88</td>
<td>408</td>
<td>0%</td>
</tr>
<tr>
<td>Friend</td>
<td>DishWasher</td>
<td>97</td>
<td>444</td>
<td>0%</td>
</tr>
<tr>
<td>Friend</td>
<td>Entertainment</td>
<td>24</td>
<td>124</td>
<td>0%</td>
</tr>
<tr>
<td>Friend</td>
<td>Fridge</td>
<td>109</td>
<td>502</td>
<td>0%</td>
</tr>
<tr>
<td>Friend</td>
<td>Oven</td>
<td>26</td>
<td>67</td>
<td>86.29%</td>
</tr>
<tr>
<td>Friend</td>
<td>WashingMachine</td>
<td>110</td>
<td>490</td>
<td>0%</td>
</tr>
</tbody>
</table>
Some tools...
Some tools (1/2)...

- Ontology engineering
  - Protégé ([http://protege.stanford.edu/](http://protege.stanford.edu/))

- Java framework
  - Knowledge base

- Reasoners
  - Pellet ([https://github.com/Complexible/pellet](https://github.com/Complexible/pellet))

- Alignment
Some tools (2/2)...

- **Ontology search engines**
  - Swoogle ([http://swoogle.umbc.edu/](http://swoogle.umbc.edu/))

- **Ontology online validators**
  - W3C ([https://www.w3.org/2001/sw/wiki/SWValidators](https://www.w3.org/2001/sw/wiki/SWValidators))

More tools are listed here:
Is SWoT good enough?
Is SWoT enough?

- SWoT seen as **Cyber-Physical systems**
  - Actions in physical environment...
    - Non-deterministic,
    - Dynamic,
    - Implicit interactions,
    - Etc...
  - One cannot ensure the application’s functionality is **satisfied and maintained over time**...
  - ...Even with semantic descriptions.
One cannot ensure the intention is satisfied and maintained over time...
Feedforward approach

\[d_{s_i} \in \mathbb{R}\]

\(d_{s_i}\) depends only on \(Model_i\), gives \(\varepsilon_{\text{sem}_i}\) \(\Rightarrow\) (Semantic gap measure)

Formal description
- Homogeneous
- Heterogeneous

Physical phenomenon
- (No model available)
- Static
- Dynamic
- Predictable
- Unpredictable

Observation
- Fully observable
- Partially observable
- Perfect
- Noisy

Intention \(_i\)

\(\varepsilon_{\text{abs}_i}\)

\(\varepsilon_{\text{sem}_i}\)

\(\varepsilon_{\text{exec}_i}\)

\(\varepsilon_{\text{meas}_i}\)
Feedback approach: -- Cybernetics --

**Intention**

\[ d_{s_i} \in \mathbb{R} \]

\[ d_{s_i} \text{ is } \text{Obs}_i \text{ dependent, gives } \varepsilon_{\text{sem}_i} + \varepsilon_{\text{exec}_i} \text{ (Takes into account physical interactions)} \]

**Model**

\[ \{\text{Composite service}\} \]

\[ \{\text{Observer}\} \]

**Process**

\[ F \]

**Selection**

**Structural composition**

**Observer**

\[ \text{Composite service} = \varphi(\text{Model}_i, k - \text{combination } \{SD\}) \]

**Infrastructure**

**Formal description**

- **Homogeneous**
- **Heterogeneous**

**Physical phenomenon**

- **Static**
- **Dynamic**
- **Predictable**
- **Unpredictable**

**Observation**

- **Fully observable**
- **Partially observable**
- **Perfect**
- **Noisy**

**Disturbance**

**Measure error** \( \varepsilon_{\text{meas}_i} \)

**Intention**

\[ \text{Absorption error} \ varepsilon_{\text{abs}_i} \]

**Model**

\[ \text{Concrete CSD}_i \]

**Concrete CSD**

\[ \varphi(\text{Model}_i, k - \text{combination } \{SD\}) \]

**Concrete CSD**

**Physical phenomenon**

\[ \text{Perfect} \]

\[ \text{Noisy} \]
Thank you!