Outline

• IoT interoperability at present
  (what companies showed at CES 2012)

• A vision of the future of IoT
  (interoperable and supporting innovation)

• Web of Things
  (reliance on the Web architecture as one enabler of interoperability on IoT)

• Semantic technology
  (for interoperability and openness on the information-level)
Interoperability at present

(in IoT-related systems)
UPnP / DLNA

- UPnP: “Universal Plug and Play”
- DLNA: “Digital Living Network Alliance”

- Not a new thing: An industry standard.

- At home, we have a Windows PC, a Sony PS3, two Samsung Android smartphones.

- All these devices can share their media (images/videos/audio) over WiFi, so that the can be played on any other of the devices.

- Windows Media Player on PC can also be remotely controlled. So, I can push a video from my smartphone to be played there.
CES 2012: “Screen-to-screen” sharing is everywhere

- Some solutions are based on video-streaming
- Some are based on data exchange
- Focus on ease: just swipe a finger.

@ Panasonic
CES 2012: “Screen-to-screen” sharing is everywhere (2)

- Not just image/video, but a set of popular applications like Email, Facebook, Google Maps, PowerPoint, etc.

- Special client app for different devices.

@ Eureka park
CES 2012: “Smart” fridges

- Can check your Facebook
- Can send a picture from smartphone
- Fridge can push to a smartphone alarms about expiring goods
CES 2012: “Smart” washing machines
CES 2012: Fridge communicating with an oven

- Fridge can have food recipes stored
- A recipe can be sent to the oven
- Oven will set the cooking temperature and other settings according to the recipe
CES 2012: Phones are connected to absolutely everything
CES 2012: Home Automation focus is on energy saving

- All is about:
  - Gateway “boxes” to access data from and control devices.
  - Interfaces for users
CES 2012: Home Automation: Lights, locks, alarms

- Can define rules, e.g.:
  - WHEN: front door is locked from outside (leaving house)
  - THEN: all light off, blinds close, etc.
One of the most visible trends at CES:

*Smartphone as a “remote control for life”* (mobile apps used for controlling home automation, home appliances, etc.)
So, interoperability?

- Well, some devices are shown interoperating.

- But, except for UPnP/DLNA-based media sharing, it is one of:
  - Buy all the devices from one vendor,
  - Connect “smart” devices (phones, TVs) from different vendors through installing a particular software client (from one vendor) on each of them (limited list of supported platforms),
  - Use a particular gateway box, then can connect devices from different vendors (from a limited list of supported by the gateway).

- In all three cases, a single vendor is responsible for all of the “interoperability”.
CES 2012: IoT-related panels

3 panels on “smart home”:
1. Mostly appliance manufacturers (Panasonic, GE, Whirlpool, etc.)
2. Home automation companies (Control4, Clare Controls, etc.)
3. Mostly networking companies and operators (Netgear, Verizon, etc.)

Opkeness?
• 3rd panel (especially Netgear representative) explicitly spoke of a need for the openness: cross-vendor interoperability, open APIs.
• In both 1st and 2nd, this topic was raised during Q&A. The answer was “yes, probably” in both cases. But IMHO: I am not sure how much they actually support this.

Connectivity gateways? (special boxes resolving interoperability issues).
The 3rd panel:
• Opinion was that it is an important first step.
• The second step would be do it gateway-free, based on intelligence spread in devices and applications.
A vision of the future of IoT

(interoperable and supporting innovation)
Vision

- Ability to have gradually growing IoT environments, contrasted to a need to install and interconnect all IoT devices and software at once.

- Ability to interconnect devices from different vendors.

- Ability of 3rd parties to develop software applications for IoT environments, contrasted to applications coming only from the devices’ vendors.

- Ability to develop applications that are generic in the sense of being able to run on various IoT device sets (different vendors, same purpose), contrasted to developing applications for a very particular configurations of devices.

In summary: “App store for a smart home”

A simple (yet difficult) scenario

1. Mary has at her home a remotely-controlled heater, an indoor temperature sensor in the living room, and a control software that adjusts the heater’s power based on the temperature and Mary’s preference settings. All devices and software are from some Vendor A.

2. Mary buys another temperature sensor, from a Vendor B, and it is installed in the bedroom. Let’s assume that:
   - Control software is able to work with more than one sensor (e.g. to keep the temperature in both rooms close to Mary’s preferences).
   - There is no common standard on how temperature measurements are encoded and transmitted. Vendor A and Vendor B use different approaches.

Problem: How to enable “Plug-n-Play”, i.e. that software of Vendor A is able to utilize the sensor of Vendor B (mostly) automatically?
A simple (yet difficult) scenario cntd.

3. Mary learns that there exists a software application from some Vendor C that is capable of energy-saving predictive heating control that takes into account not only the indoor temperature but also the changes in the outdoor temperature:
   • Mary wants to use this application instead of the control software she has in place. She downloads it from a Web-based store, or runs a service instance in the Cloud (the latter is probably easier to realize).
   • The application discovers the heater and the two indoor temperature sensors Mary has at home.
   • It then fails to find an outdoor temperature sensor in Mary’s home network and opts to use an appropriate Web-service (current data or a forecast) for that.

Problem: How such a decoupling of hardware (and other data providers) from software applications can be possible, in particular, that generic applications are able to configure themselves to work with various sensors / devices / services?
IoT supporting innovation

Variety of companies and individuals able to develop and publish IoT software applications (as they develop smartphone applications at present)

A plethora of innovative applications and ideas for future evolution of hardware

(not stuff like checking Facebook on a fridge)

Safety and security become even more important issues then…

(outside the scope here)
Additional wish: 3 IoT worlds interoperable

- **Attached devices**: Identifiers such as RFID tags or barcodes are attached to things to enable their automatic identification and tracking. Based on a thing’s identifier, the information about the thing is then retrieved from a database or from the Web.

- **Sensing and Actuating devices**: These devices are placed in the close vicinity of ‘things’ and provide a “second-hand” access (from outside) to their properties or functions. Examples are temperature and other sensors, cameras for cars’ register plate recognition, and actuators like remotely-controlled door locks or window blind controls.

- **Embedded devices**: Some ‘things’, such as industrial machinery, home electronics, smart phones, wearable devices, have embedded processors, data storages, sensors and actuators, enabling “first hand” access (from inside) to them, often over IP and without special gateways.
Two approaches

• **Standardize everything**
  (difficult to reach agreements, difficult to cover everything in an evolving world)

• **Some level of standardization + some intelligence**
  • E.g. Web protocols and formats (W3C recommendations)
  • Plus intelligence through semantic technology
    • Requiring formats/languages for encoding metadata and ontologies (W3C recommendations)
Web of Things
Web of Things

• A trend in IoT area is to attempt to integrate ‘things’ seamlessly with the existing Web infrastructure and to expose connected ‘things’ uniformly as Web resources, resulting in what is called the Web of Things (WoT).

• The aim is to reuse the architectural principles of the Web and apply them to the connection with the real world, i.e. with (smart) entities e.g. smart fridges (with embedded computers), smart packages (with RFIDs), smart rooms (with sensors and actuators), thereby making them first-class citizens of the Web.
Integrating the ‘things’ into the Web

WS-* vs. REST

Developers’ opinion survey in Ph.D. Thesis of D. Guinard
REST requests

**REST:** “Representational state transfer”

**RESTful:** stateless, the server does not have to keep the history of past requests to process and answer a request.

- GET is used to retrieve the representation of a resource.
- POST creates a new resource (includes payload).
- PUT is used to update the state of an existing resource or to create a resource by providing its identifier (includes payload).
- DELETE is used to remove a resource.

GET http://<domain>:<port>/generic-nodes/1/sensors/temperature

PUT http://<domain>:<port>/generic-nodes/1/sensors/temperature/value

{…}

Ph.D. Thesis of D. Guinard
REST responses and payloads

- JSON (easiest to process from JavaScript, i.e. in client Web apps)
  
  ```json
  {"resource":
    {"content":
      {"name": "Temperature",
       "content":
         {"name": "Current Temperature",
          "description": "Ambient Temperature",
          "value": 24.0,
          "unit": "celsius"}
       }
    }
  }
  ```

- XML
- CSV
- RDF
Pachube.com

- Send your data to Pachube Cloud DB over simple REST API
- Access the data from other components of your application
  
  **Request**: GET http://api.pachube.com/v2/feeds/1977/datastreams/1
  
  **Response**: `{ "current_value":"100", "max_value":"10000.0", "at":"2010-07-02T10:16:19.270708Z", "min_value":"-10.0", "tags":[ "humidity" ], "id":"1" }`

- Utilize reusable Web widgets,
  - E.g. for visualization of data-streams
  - E.g. for wrapping user input as data-stream(s)

- Utilize other reusable components
  - E.g. for representing user’s skype, etc. online / offline status as a datastream
Globally accessible blackboard

- The blackboard approach like in Pachube is one approach to interoperability per se.

- While potentially having some issues with respect to scalability and performance, on the positive side it separates the data itself from such questions as data availability (where?) and transmission (when?).

- Such a blackboard approach has also been recently utilized outside the “Web of Things” work, e.g. in Tekes DIEM / EU SOFIA projects (Smart-M3 platform):
  - Proprietary protocol.
  - API libraries for various platforms, which is just slightly better than “particular software client” (from an earlier slide).
Web of Things Architecture

Ph.D. Thesis of D. Guinard
Semantic Interoperability
Is WoT enough?

• Applying Web architecture to Internet of Things is a great facilitator of interoperability.

• WoT, however, is mostly about the protocols and formats.

• WoT as such will not enable realization of our “simple (yet difficult) scenario” (from earlier slides):
  • E.g. two temperature sensors both delivering measurements over HTTP GET as JSON, but of different structure and with different object/property names.
  • E.g. two heater devices accepting commands over HTTP PUT as JSON, but of different structure and with different object/property names.

• For true interoperability, we need also the semantic interoperability, the ability of the devices to unambiguously convey the meaning of data they communicate over Web protocols.
One slide summary: What is semantic technology

- **Question:** How to make computers act in an intelligent way?
  - Approach 1: To make computers *so clever* that they will be able to process the information about the world in its *full complexity*, e.g. understand human language => traditional Artificial Intelligence (AI)
  - Approach 2: To *simplify* the description of the world to a level that even *stupid* computers will be able to act “intelligently” based on it => Semantic Technology.

- **The goals of semantic technology:**
  - to make the meaning of data is as *explicit* as possible: as *unambiguous* as possible and as *context-independent* as possible.
  - to *link* data sources globally: more meaning with same data (small messages, fetch the rest from Web).

- Achieved through “three pillars” of the semantic technology: *Semantic Network data model, URI, and Ontologies.*
Semantic reasoning for interoperability

**Query**

```
SELECT ?x WHERE { ?x a family:Mother }
```

**Data**

- `org:Mary a person:Woman.
- `org:Mary person:hasSon org:Jack.
- `org:Mary human:hasSex human:Female.
- `org:Mary human:hasChild org:Jack.
- `org:Mary a family:Mother.

**Domain ontology**

- `family:Mother` is a subclass of `human:Human` with the restriction that it must have a property `human:hasSex` with the value `human:Female` and must also have at least one property `human:hasChild`.

**Definition of family:**

```
person:Woman is a subclass of person:Person which is in turn a subclass of human:Human. person:Woman has a restriction to have a property human:hasSex with the value human:Female. Also, person:hasSon is a sub-property of human:hasChild.
```

**Upper ontology**

```
rdf: , rdfs: , owl: 
```

**RDF-S / OWL rules**

- `org:Mary human:hasSex human:Female.
- `org:Mary human:hasChild org:Jack.
- org:Mary hasSon org:Jack.

**Definition of person:**

- `person:Woman` is a subclass of `person:Person` which is in turn a subclass of `human:Human`. `person:Woman` has a restriction to have a property `human:hasSex` with the value `human:Female`. Also, `person:hasSon` is a sub-property of `human:hasChild`. 
Ontology linking process

Has unknown concepts?

Yes

Is a registered upper ontology?

No

Obtain the definition of own domain ontology in terms of the upper ontology

Success?

No

Attempt ontology alignment

Done

Success?

Yes

Respond: NOT UNDERSTOOD

Success?

No

Attempt one or both:
• Download online ontology definition
  - may have to ask the sender for URL
• Ask the sender for the ontology definition

Semantic reasoning for dynamic coordination

Agent 1: “I plan to \textit{x:Send} a book \textit{y:HarryPotter} to \textit{org:AgPS4e}”

Agent 2:

- “I now \textit{z:Scan} a document on \textit{org:AgPS4e}”
- “Does Agent1’s intention concern me?”
- “I do not know what \textit{x:Send} means..”

Domain ontology

Definition of \textit{DO1}

Definition of \textit{DO2}

Upper ontology: \textit{BDI}

Upper ontology rules

Coordination rules

Beliefs-Desires-Intentions model

e.g. of Tamma et al. (2005)

To interpret action intentions

To resolve conflicts

Semantic reasoning for dynamic coordination

Intention

Data

Domain ontology

Domain ontology
Semantic reasoning for dynamic coordination (2)

1. I plan to x:Send y:HarryPotter to org:AgPS4e
2. I now z:Scan a document on org:AgPS4e
   • Does Agent1’s intention concern me?
3. AgPS4e is a multi-function device =>
   Agent 1 needs only the printing part of it
4. Agent 2’s domain ontology
5. I use only the scanning part of AgPS4e =>
   No conflict
6. Go on
7. Scan utilizes only the scanning component of the device if it is a multi-function printer.

Performing the behavior “Send” on a digital document and a printer means Print activity. Print utilizes only the printing component of the device if it is a multi-function printing device.

Definition of Agent1’s domain ontology
Definition of Agent2’s domain ontology

Interoperability on functional interfaces

Note: This demonstrative simple example is about programmatic integration of Application Programming Interfaces (APIs). The described issue, however, equally applies also to other kinds of functional interfaces, e.g. Web services.

Considering an application “Find a Contact, then Compose a message to him/her”:

- APIs are semantically compatible.
- Syntactically, there is nothing indicating this compatibility.
- A human has to possess the knowledge of the compatibility and of the “adapter”:
  
  ```cpp
  ((QContactPhoneNumber)contact.detail(QContactPhoneNumber::DefinitionName)).number()
  ```
Semantic support

• Annotate “FindContact” function as producing an output that is semantically belongs to the class `nco:Contact` and is syntactically represented as `qtm:Contact` (meaning QContact)

• Annotate ”ComposeSMS” function as taking an input of the class `nco:MessagingNumber` and the format `qtm:PhoneNumberString`.

• State also that:
  
  `nco:CellPhoneNumber rdfs:subClassOf nco:MessagingNumber.`
  
  `nco:CellPhoneNumber inter:partOf nco:Contact.`

• Provide description of an “adapter” - provides transformation from one syntactic form to another - `qtm:Contact` to `qtm:PhoneNumberString` in this case (the line of code of the previous slide).

• Now we have enough “semantic glue” so that a computer can automatically: understand compatibility of the two functions and integrate them through the adapter.
Supported composition / integration

• Compatible functions can be found automatically.
• Adapters can be found and incorporated automatically.
• *It is like software documentation that can be automatically put into use.*

• In our work on “Smart Modeller” tool, the goal was semi-automated *mash-up* of applications (“offline” mode).
• Exactly the same logic can also be used for run-time compatibility resolution.

Summary

Our vision of interoperability on IoT is:
• Ability to have gradually growing IoT environments.
• Ability to interconnect devices from different vendors.
• Ability of 3rd parties to develop software applications for IoT environments.
• Ability to develop applications that are generic in the sense of running on various IoT device sets (different vendors, same purpose).

We believe that:
• Web of Things approach (RESTful HTTP) is the first enabler.
• Semantic technologies is the second one.