Natural language for an interoperable Internet of Simple Things
Users and businesses require connected devices to **interoperate** seamlessly – and that’s not yet easy

40% of IoT business value will depend on solutions’ ability to interoperate
(McKinsey Global Institute, 2015)
It is difficult for people to stay in the loop – Connected things, applications and humans should be members of the same club

People need an easy and natural access to:

• Control devices and services they use
• “See” what’s going on, replay/forensic

The “one device - one app” paradigm does not work well anymore:

• The number of apps really used on a smartphone is decreasing (down to ~10)
Many connected devices are simple things, able to execute only a few actions or to provide a few data.

Smart light:
- Turn on/off
- Toggle
- Set color
- Set luminosity

Connected washer*:
- Monitor cycle status & time remaining
- Extend the dryer cycle
- Receive an alert when the cycle has finished
- Remote start the cycle

(*) Here: GE’s GTW washer
Semantics is a convenient foundation for interoperability

Diversity is in the nature of things:
  • Low level standardization is intractable
  • **Semantics** interoperability can cope with it

Current approaches are two-fold:
  • **Implicit semantics for H2M**: Conversational User Interfaces (natural language)
  • **Explicit semantics for M2M**: formal ontologies (oneM2M, Web of Things)
Could natural language be used to unify H2M and M2M interactions?

**Pros**
- ‘Natural’ for people usage
- Crowdsourced, no need for new standard
- Stable (compared to IT life-cycles)
- Maturing natural language technology (e.g. chatbot and smart speakers)

**Cons**
- Not so ‘natural’ for machine communication
- Natural language is occasionally ambiguous
- Natural language processing can be complex
- Several natural languages should be considered
The 7 Elements Difference Game for Natural Language applied to interoperability

<table>
<thead>
<tr>
<th>Criteria</th>
<th>NL for Human to Machine</th>
<th>NL for Machine to Machine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality</td>
<td>Few false positive and good user experience: Precision and Recall are equally important</td>
<td>No false positive: Precision is more important than Recall</td>
</tr>
<tr>
<td>Flexibility</td>
<td>NL queries can be very different, ill-formed, include typos...</td>
<td>NL queries should be “reasonable” and well-formed</td>
</tr>
<tr>
<td>Compactness</td>
<td>H2M dialogs might be ‘multi-turns’ to help completing queries</td>
<td>M2M dialogs are concluded in one turn only</td>
</tr>
<tr>
<td>Predictability</td>
<td>Not a requirement as long as Quality is good enough</td>
<td>Results must be repeatable (same utterance provides same outcome) for replay/audit/forensic</td>
</tr>
<tr>
<td>Rapidity</td>
<td>Delay in the communication chain must be acceptable by people (~1 s)</td>
<td>No significant delay in the communication chain (~1 ms)</td>
</tr>
<tr>
<td>Load</td>
<td>CPU and memory footprint do not really matter in most cases (e.g. cloud based chatbots)</td>
<td>No significant CPU/memory footprint for scalability and possible integration into devices/gateways</td>
</tr>
<tr>
<td>Economics</td>
<td>Reasonable additional costs (&lt; 10.000€ per million queries) are acceptable</td>
<td>No significant additional cost (&lt; 10€ per million queries) (economics of IoT is tight)</td>
</tr>
</tbody>
</table>
Simple things allow combining natural language flexibility with machine language performance.

- **Narrow context** ("this thing", "this service") limits the scope of natural language understanding (*pragmatics*).
- **Well identified features** (switch-on, set color, open/close…) can be reliably mapped on ‘intents’ (*semantics*).
- **Low variability** (M2M) of software generated utterances simplifies natural language parsing (*syntax, morphology*).

### Graph

- **Machine language**
  - Speed of dialogs: 1,000/sec
  - Flexibility: Low
- **Machine-to-Machine NLP**
  - Speed of dialogs: 100 to 1,000 times faster, lighter, and cheaper
  - Flexibility: High
- **Human-to-Machine CUI**
  - Speed of dialogs: 1/sec
  - Flexibility: Low

**NLP**: Natural Language Processing  
**CUI**: Conversational User Interface
Example of a **smart gatekeeper** made of disparate devices and services using NL as unified communication layer.

- **Smart Camera with face recognition**
- **Slack** - Social platform
- **Wikipedia web service**
- **Cloud-based orchestrator**
- **Smart Light**
Smart gatekeeper’s simple scenario

1. The camera is pre-trained with the face of persons to be “accepted”
2. When the camera sees someone, it sends an event to the orchestrator with the recognized/unrecognized attribute
3. The orchestrator switches the light in green (“accepted”) or in red (“rejected”) and, if “accepted”, asks Wikipedia to provide an extract of the visitor’s bio
4. The verbatim - including the visitor’s bio - is published on the social platform
Smart gatekeeper in action

slackbot

Smart Camera ➔ Orchestrator: I have seen Mona Lisa
Orchestrator ➔ Smart Light: Turn on my Smart Light in green

Smart Light ➔ Orchestrator: Success
Orchestrator ➔ Wikipedia: Give me the bio summary of Mona Lisa
Wikipedia ➔ Orchestrator: The Mona Lisa (Italian: Monna Lisa ['monna ˈliːza] or La Gioconda [la dʒɔ'konda], French: La Joconde [la ʒɔkɔ̃dɔ]) is a half-length portrait painting by the Italian Renaissance artist Leonardo da Vinci ...

Smart Camera ➔ Orchestrator: I have seen Satyajit Ray
Orchestrator ➔ Smart Light: Turn on my Smart Light in green

Smart Light ➔ Orchestrator: Success
Orchestrator ➔ Wikipedia: Give me the bio summary of Satyajit Ray
Wikipedia ➔ Orchestrator: Satyajit Ray (Bengali: ['sɔt̪ajitˌraɪ] (listen); 2 May 1921 – 23 April 1992) was an Indian filmmaker, screenwriter, graphic artist, music composer and author, widely regarded as one of the great ...

Smart Camera ➔ Orchestrator: I’ve seen a person I can’t recognize
Orchestrator ➔ Smart Light: Turn on my Smart Light in red
Smart Light ➔ Orchestrator: Success
Natural language fits well for smart services that require an extended community of devices and people

Main relevance criteria for use cases:

- **People** matter (family, citizens, customers, travelers...)
- IoT is **fragmented** (vendors, services, phased deployments...)
- **Machine-to-machine** to be significant (distributed intelligence)

**Home**
- Home automation, lighting, audio/video, security...

**City**
- Digital signage, lighting, traffic, accessibility, safety...

**Building, office and hospitality**
- Access control, HVAC, workspaces, energy...

**Retail**
- Drive, contextual adds, signage, logistics...

**Utilities**
- Metering, grid, E-vehicle, pipeline, water, waste...
From H2M and M2M fragmentation to an integrated H+M IoT world

This is a no-standard, low profile approach of semantics interoperability, in the context of “simple things”.

It should be complementary to more ambitious oneM2M and W3C approaches.

Natively gathering people, machines and applications, could natural language open new applications fields and support still unexpected services?

You have more questions or want to discuss?
thierry.grenot@lepeuplehabile.com or www.linkedin.com/in/thierrygrenot/