Natural Language for an Interoperable Internet of Simple Things

Grenot Thierry
Le Peuple Habile SAS
Paris, France
thierry.grenot@lepeuplehabile.com

Callejas Zoraida
Dept. of Computer Languages and Systems
Universidad de Granada
Granada, Spain
zoraida@ugr.es

Griol David
Dept. of Computer Science
Universidad Carlos III de Madrid
Leganes, Spain
dgriol@inf.uc3m.es

McTear Michael
Computer Science Research Institute
Ulster University
Belfast, United Kingdom
mf.mctear@ulster.ac.uk

Bandyopadhyay Debopam
Alumnus Software Ltd
Kolkata, India
debopam@alumnux.com

***Abstract*— The universal deployment of the Internet of Things requires interoperability among people, devices and applications. On the other hand, most devices are actually fairly *simple things*, able to provide only a few specialized pieces of data or to carry out limited actions. We explore here the opportunity and conditions to use natural language as a simple and straightforward vehicle to achieve an interoperability scheme that is universal, support very long lifecycles, break silos and embrace people, devices and applications all together. We identify and compare the different characteristics of natural language processing applied to human-to-machine and to machine-to-machine situations. Finally we demonstrate that a quite simple implementation could match the identified criteria and deliver interoperability in a simple things environment.**

Keywords—Internet of Things, interoperability, natural language, semantics, simple things.

I. Introduction

"Simple things" as we named them here, are connected devices with limited capabilities, specialized in narrow, well defined tasks. This is the case for most devices that are currently deployed in the Internet of Things (IoT) market; to name a few:

- Environmental sensors (humidity, temperature, noise, air quality, etc.)
- Security sensors (smoke detector, presence and opening detectors, monitoring camera, etc.)
- Specialized sensors (weather station, smart meter, parking lot occupation detector, traffic meter, etc.)
- Specialized actuators (thermostat, smart lock, smart light, smart shutter, etc.)

IoT is currently deployed in well-defined vertical markets (smart city, smart home, smart industry, smart agriculture, etc.) by homogeneous providers (technology vendors or platform providers). However, business and personal users will progressively require more open approaches where heterogeneous services can combine connected objects and services of a very different nature, providers and technologies. Imagine we could easily…

- Combine similar objects from different vendors in the same service (e.g. smart lights of different brands and models in the same building);
- Mix connected devices from several vendors into personalized and flexible services (e.g. user-defined home automation, mobile services over heterogeneous infrastructures like cities, harbors, warehouses…);
- Use the same sensors and actuators within multiple services (e.g. same air quality captors within a city supporting public digital signage, health care and traffic control services);
- Guarantee long-term compatibility and extended life time of expensive IoT infrastructures that take a long time to deploy (e.g. city lighting, active parking sensors, rain detectors, CCTV…).

To make simple things interoperable in an equally simple way, we propose here to use natural language. Chatbots and smart speakers already use it for human-to-applications or human-to-devices dialogs; extending this model to device-to-device and application-to-device situations will unify communications in a holistic manner, helping to achieve the expected global interoperability.

After briefly presenting related work in Section II, we describe different aspects of global IoT interoperability in Section III. The role and implications of using natural language as support for semantics interoperability is highlighted in Section IV. Section V describes an implementation of a smart service that combines disparate simple things and connected services based on natural language. We briefly report directions for future work in Section VI and finally present the conclusions of this paper in Section VII.

II. Related Work

The Social Internet of Things (SIoT) is introduced in [1], where the authors identify policies for the establishment and

978-1-5386-4980-0/19/$31.00 ©2019 IEEE

the management of social relationships between objects and describe a possible architecture that includes the functionalities required to integrate things into a social network of things.

An inter-thing relationships programming framework is proposed in [2], as a basis for a distributed programming ecosystem as in the Social Internet of Things. The framework extends the limited set of relationships with a set of service-level relationships that describe how thing services can be combined to build meaningful IoT applications.

The Virtual User for the IoT discussed in [3] proposes to augment (human) user capabilities with virtualization functions so as to make easier and faster involvement in applications, facilitating their integration into the technological digital world.

Human-machine interactions based on natural language are now deployed in everyday life (for example with virtual personal assistants and smart speakers). A comprehensive introduction to Conversational User Interfaces (CUI) between humans and smart devices is provided in [4].

To address interoperability issues, the technical community has organized itself around associations that group vendors as within the Open Connectivity Foundation [5]. Standardization bodies are also active with projects to counter the fragmentation of the IoT, for example the W3C Web of Things (WoT) [6] and oneM2M, a joint effort from the world's preeminent standards development organizations [7]. Their approach is to define architectures, data structures and protocols that provide access to formal ontologies describing the organization and capabilities of devices. By parsing these ontologies, applications and platforms can discover what can be done with devices, how to activate their features and to get data from them.

However, this approach is complex and requires significant efforts by the standardization bodies themselves (including agreeing on vocabulary and standard ontologies sources), as well as by the device manufacturers, platforms editors and service providers that will need to understand such concepts and develop related solutions, with a still unclear transition path. Additionally, these documents maintain a disconnection between the world of people and the world of things that will make difficult to reinstall *humans in the loop*.

To address these limitations, we propose here an alternative currently limited to *simple things*: directly using natural language to support interoperability among devices, applications and people.

III. The Need for Global Interoperability

We briefly review here the need for interoperability in the IoT environment and the necessity to find long-term solutions that also take users into account.

A. The Need to Interoperate

At the pre-internet age, networks were isolated and a plethora of standards blossomed and faded nearly every day. Thanks to the universality of IP and HTTP (and a few others), digital services can now be combined in a seamless and secure manner, unleashing a world of innovation with considerable value, that structure the life of people, businesses and society in general.

Similarly, still in its infancy, the IoT ecosystem sustains the unavoidable Darwinian trial-and-error selection among a moving set of ad-hoc deployments, evolving standards, vendors' associations and standardization bodies. They all contribute in their own ways to the maturation of technologies and markets towards IoT global interoperability [8].

Studies have estimated the IoT total potential impact between \$3.9 to \$11.1 trillion per year in 2025; nearly half of this business value should depend on the ability of solutions to interoperate [9]. There is no doubt that IoT stakeholders will find ways to deliver such a considerable value sooner or later.

B. Life-cycle Clash

IoT means nearly literally "Software over Hardware." Intangible software and tangible hardware assets have a discordant lifestyle, with a lifetime of typically 2 to 5 years for software and 20 to 50 years for heavy infrastructure like buildings, transport, hospitals, cities, etc.

Obsolescence might be (consciously or not) planned by hardware vendors and software editors who will find it simpler (and perhaps more profitable) to force the early replacement of systems. This life-cycle discrepancy will lead to a severe underestimation of the Total Cost of Ownership (TCO) for IoT infrastructures: complicated support, early end-of-life, vendors lock-in, etc.

There is a strong need for solutions that bring stability over a long period of time and can support progressive evolutions without putting at risk the delicate technical and economical balance of complex infrastructures.

C. Of Devices and Humans

Interoperability can be defined as the ability of products made by different vendors at different periods of time to exchange information and to collaborate to jointly deliver combined services. On a larger standpoint, we should also add to this landscape the users of such services, which can be human beings and digital applications.

People interactions with their connected devices is quite cumbersome. Users are discouraged by the "one thing one app" paradigm that antagonizes the benefits of having new features or a better interface. There is even a decline in the number of applications that people use on their smartphones, the daily estimated average being between 8 and 12 [10]. Alternatives are proposed, for example chatbots, smart speakers or Progressive Web Apps (PWA), to promote easier interactions between people and digital services.

At the same time, the number of public and private APIs is growing fast [11]. Each has its own life-cycle and is updated and deprecated on an irregular basis, forcing client applications to be updated as well in a coordinated manner. This creates complex interrelations that lead digital integrators and service providers into considerable effort to maintain their systems in a good operating condition.

In summary, an ideal IoT interoperability scheme should be universal, support very long lifecycles and embrace people, devices and applications all together.

IV. Natural Language for the Internet of Simple Things

After positioning semantics as the preferred level for interoperability, we introduce the case of *simple things*. Then the pros and cons of using natural language are presented, followed by a comparison of human-to-machine with machine-to-machine requirements.

A. Semantics is the Right IoT Interoperability Level

Diversity is in the nature of things, whether connected or not. We can easily find a tenth of vendors and hundreds of models of each type: smart thermostats, lights, washers, fridge, camera, vehicle detection sensor, trackers, etc. This diversity is growing as IoT penetrates homes, offices, cities, hospitals and all sort of businesses. Each device is different for good reasons: environment, local standards, connectivity, performance, size, price, design, distribution channels, mode of installation, etc. Variety will stay and even grow as use cases proliferate.

All these devices should be able to interoperate in an easy manner with applications, people and among themselves for Machine-to-Machine (M2M) services. As low level device interface standardization is intractable and counterproductive, the alternative is to build interoperability at a semantic level, where the meaning of commands and responses is at the core of the communication, on top of a flexible lexicon and syntax.

B. Simple Things Require Simple Solutions

As already stated, “simple things” are connected devices with limited capabilities, specialized in some narrow, well defined tasks.

Each device exists in many versions from many vendors and they are reached by applications through different interfaces (API), even though they carry out similar and limited tasks: switch on/off, open/close, use a few parameters, provide a few metrics, etc. For example, a smart light is only able to switch on, switch off, change its color and set its luminosity; a connected washer can be remotely started, monitor its status, provide the time remaining and send an alert when the cycle has finished. Some vendors provide additional fancy features (blinking, periodic effects, etc.), however they are rarely used.

To make simple things semantically interoperable, we need a common language among them. We could imagine defining a new ‘Esperanto of things’ in the form of a structured language that everyone would accept. However, this will not only require defining it, but also obtaining the agreement of every stakeholder, which is practically impossible. So, we propose that devices (or their gateways) directly use the pre-existing (human) natural language to communicate.

Chatbots and smart speakers already do this for human-to-applications or human-to-devices dialogs in what is called Conversational User Interfaces [4]. Extending this model to device-to-device and application-to-device situations would provide a good step towards global interoperability.

C. Natural Language for the Internet of Simple Things

Collectively defined by everyone, belonging to no-one, with an inherent capability to convey meanings of queries and replies, natural language has some characteristics that we must use (pros) or mitigate (cons); among which are:

Pros:

- Free and readily available as a natural means of communication for humans;
- Stable over a period of time much longer than any IT technical standard;
- Ability to use existing text-based communication services.

Cons:

- Sometimes ambiguous, so obtaining a unique interpretation might be difficult;
- May involve complex processing;
- More difficult to process automatically compared with machine languages.

Whilst benefits speak for themselves, natural language drawbacks should be accommodated in a straightforward manner. The narrow context of simple things allows the implementation of simple and fast Natural Language Processing (NLP) techniques that can combine quality, rapidity and cost efficiency.

D. Natural Language Processing Requirements

Using natural language to support machine-to-machine (or application-to-machine) communication will imply significant changes in NLP, compared to human-to-machine conversational user interface.

TABLE I. NLP Requirements Comparison

Criteria	***Human-to-Machine***	***Machine-to-Machine Application-to-Machine***
Quality	Few false positive and good user experience: Precision and Recall are equally important	No false positive: Precision is more important than Recall
Flexibility	NL queries can be very different, ill-formed, include typos…	NL queries are “reasonable” and well-formed
Compactness	Multi-turns dialogs may be required to help completing or disambiguating queries	Dialogs must be concluded in one-turn only
Predictability	Not a requirement as long as Quality is good	Results must be repeatable (same utterance provides same output)
Rapidity	Delays in the chain must be acceptable by people (~1 second)	No significant delay in the M2M communication chain (~1 ms)
Load	CPU and memory footprint do not really matter (cloud services)	No significant CPU or memory impact for scalability and integration
Economics	Reasonable additional costs are acceptable (< 10.000$ per million queries)	No significant costs as the economic equation of simple things is restricted (<10$ per million queries)

Table I compares the necessary characteristics of NLP applied to the interoperability of simple things in human-to-machine situations (e.g. chatbots) with machine-to-machine and application-to-machine situations.

1) Quality: Let's define true positive as the case when a device accurately executes a correct natural language instruction, false positive when a correct instruction is misinterpreted so a wrong action is induced, and finally false negative when a correct instruction is rejected. Quality is generally expressed as Precision and Recall, two competing metrics that span from 0 (bad) to 1 (good). Precision is the ratio of true positives divided by the sum of true positives plus false positives, and Recall is the ratio of true positives divided by the sum of true positives plus false negatives. To provide Accuracy, Precision and Recall are combined into the F1 score, their harmonic mean. In the machine-to-machine case the ideal situation is to have a one-turn dialog, so we are looking for very high Precision (ideally equal to 1, i.e. no action should be falsely undertaken), above the usual target of human-to-machine situations, and so even at the price of a relatively low Recall.

2) Predictability: For machine-to-machine, NLP should be able to provide repeatable results where the same query ("switch on the light in dark green and maximum luminosity") should always lead to the same results. It will then be possible to anticipate results and replay situations for test, validation, qualification, support and forensic purposes.

3) Flexibility: Natural language queries ("intents") should be accepted in undocumented different forms: "turn on the light…", "the lamp must be switched on…", "set this bulb on…". In the machine-to-machine case, expressions will be generated by software and as such will be well-formed and less variable than in the human-to-machine situation. Unreasonable, tricky, rare expressions should be rejected by the NLP to ensure the requested high precision without impacting the actual flexibility of the interaction.

4) Compactness: In machine-to-machine situations there is no human in the loop, therefore no trivial possibility to correct erroneous interpretations, to request confirmation, to enter a completion or repair phase ("are you sure …? "can you confirm…?" "which color would you prefer?"). Hence dialogs must be concluded in one turn in most (if not all) cases.

5) Rapidity: Fluid human-to-device interactions based on natural language (like chatbots) require a speed equivalent to real-time human-to-human communication, approximately 1 second. Similarly, application-to-machine and machine-to-machine interactions based on natural language should appear as fast as in the usual machine language situation, i.e. a few milliseconds only.

6) Computing load: Machine-to-machine NLP computing workload and memory footprint should be kept low to prevent significant impact on the production and operation cost of the devices, gateways or cloud services running it. At the same time, it should also permit highly scalable deployments for centralized or cloud-based services.

7) Economic impact: Simple things are to be deployed *en masse*, with a low unit price that does not give rise to many additional costs. The introduction of natural language must not significantly impact the global economic equation for these simple devices and the associated services that they enable.

To match machine-to-machine requirements, NLP software might have to implement different principles from current chatbots and conversational assistants. In particular, it is yet to be confirmed whether (or how) deep-learning and probabilistic parsers, that provide good performance in human-to-machine context, could provide all the necessary characteristics in the simple things environment [12].

V. Example of Implementation

To illustrate these concepts, we have developed a smart gatekeeper scenario that is briefly explained in this section.

A. Set-up and scenario

A connected camera with face recognition looks at people and decides to let them in (authorized person) or to block them (unknown or excluded person). Moreover, when people are authorized, their short biography is extracted from the Web and published on a social platform for information traceability. Access authorization or denial is indicated by a change of color of a connected smart light (green or red).

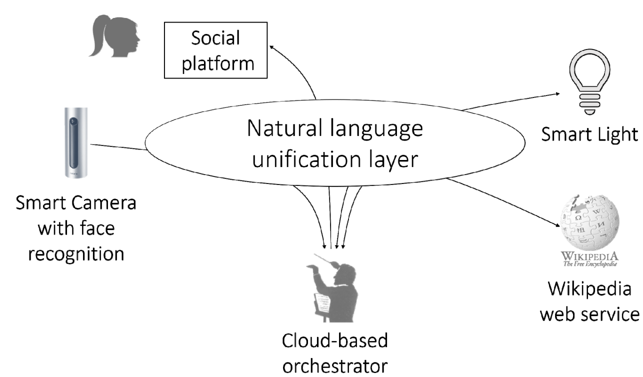

Fig. 1. Demonstrator set-up

Fig 1 shows the demonstrator set-up, which was made from off-the-shelf devices and services from different vendors, with no straightforward compatibility between them:

- A smart camera able to recognize pre-trained faces and to send webhook events (recognized person with her identity, unrecognized person);
- A smart light with programmable color and luminosity;
- An on-line biography service (Wikipedia);
- A cloud-based orchestrator to sequence the scenario;
- A collaboration platform to print out the biography of authorized persons and provide the transcripts of the smart gatekeeper service (Slack).

In order to interconnect these devices and services in natural language, we developed two applications, again cloud-based:

- A translating application (natural language unification layer) that turns natural language into the corresponding machine language (API on each of the vendors' gateway, webhook), and vice-versa;
- A management application (not represented here) that monitors the translating application and gathers metrics like errors, intents and delays (average and standard deviations).

To provide additional flexibility and facilitate testing, we have also implemented a direct natural language access from the collaboration platform to the smart light, so that a member of the workgroup can "write" to the light just as she would post a direct message to another member.

B. Demonstrator NLP

English was the chosen natural language. The smart light intents were: "switch on" (optional attributes: color, brightness), "switch off", "toggle on/off", "set color" (required attribute: color), "set brightness" (required attribute: brightness level). The camera events were: "known person seen" (with her name), "unknown person seen". The Wikipedia intent was: "provide short bio" (required attribute: name of the person). In addition, we developed two generic intents for each device: "Who are you?" and "What you do for me?" Color and brightness attributes can be expressed in various manners, including figures (10%...) and natural expressions (sky blue…).

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 [ˈmɔnna ˈliːza] or La Gioconda [la dʒoˈkonda], French: La Joconde [la ʒɔkɔ̃d]) 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: [ˈʃɔt̪ːodʒit̪ ˈrai̯] (listen); 2 May 1921 – 23 April 1992) was an Indian filmmaker, screenwriter, graphic artist, music composer and author, widely regarded as one of the greatest...

Fig. 2. Example of a transcript of the smart keeper scenario

Fig. 2 shows an extract of the smart keeper transcript when two people (Mona Lisa and Satyajit Ray) were recognized. Smart Camera, Orchestrator, Smart Light and Wikipedia are exchanging information that is published on the collaboration platform.

The Natural Language Understanding (NLU) pipeline was constructed using a suite of tokenizer, stop words remover, stemmer, lexical analyzer, entity solver, deterministic intent parser, and finalizer (Fig. 3). It was trained off-line from a small number of utterances (approximately 30 utterances per intent). It provided its output (intents, entities, values) to the dialog manager (not represented) that managed slots (optional or required parameters) and values in different formats.

The dialog management involved a single turn: either the intent was recognized with at least its required attributes ("change the color to light green") or it was refused ("change the color") with an explanation message in natural language.

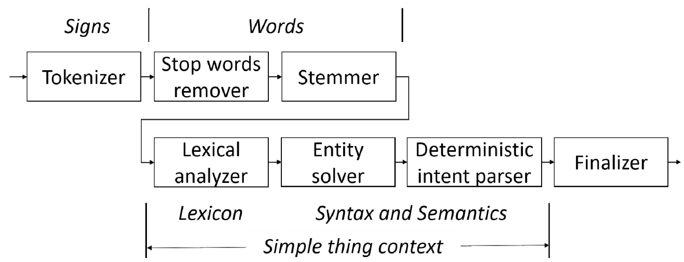

Fig. 3. NLU pipeline architecture

C. Measured Performance

We built a validation set for the smart gatekeeper context (including the natural language access from the collaboration platform) of 500 utterances by combining and deduplicating several sources: 10 people in our professional environment (not involved in the project), the Amazon Mechanical Turk and sentences related to handling lights that we have extracted from the NOW (News On the Web) corpus [12].

We obtained the following NLP performance results:

- A high precision (1.0);
- An acceptable recall (0.66) ;
- A reasonable accuracy – F1 score (0.79).

We also measured the natural language to machine language conversion time on a set of 5.000 queries that went through the translating application, including communication interfaces, NLU pipeline and API interfaces. Not specifically optimized, the application was coded in NodeJS, ran over an Ubuntu distribution of the Linux operating system and was deployed on low-cost, entry-level economical virtual machines within the Microsoft Azure cloud platform. We found:

- An average delay of 3.9 milliseconds;
- A standard deviation of 3.7 milliseconds.

Because of the multiplicity of situations and the quasi impossibility to use a common validation set, comparing NLU performances is always difficult. However it is generally accepted that state-of-the-art performance for human-to-machine solutions is an F1 accuracy between 0.65 and 0.95 ([13]). Response time is usually between 0.2 and 2 seconds.

In summary, our implementation achieved a good quality level with a very high precision, a very fast response time and a low CPU/memory impact, compatible with the machine-to-machine targeted use case.

We believe that the narrow context of simple things is instrumental to the overall performance, as it permits achieving good precision, predictability and speed all together, with a relatively simple implementation.

VI. Future Work

This approach to interoperability has provided promising results, however improvements are yet to be achieved, mainly in architecture and NLP levels.

1) Architecture: To accommodate a large variety of deployment situations, it should be necessary to integrate the translation application not only in cloud servers, but also within devices, gateways and IoT platforms. This will require improving the modularity of agents and developing easy-to-use software production and testing tools.

2) Platform: The native interoperability of applications, people and connected devices around natural language will raise the need for flexible exchange platforms able to support any-to-any communication mesh. It will also be useful to add a voice interface to complete the social platform access for human-to-machine communication.

3) Natural Language Processing: While English will likely be sufficient and handy to support machine-to-machine and application-to-machine communications, it will be necessary to accommodate more natural languages to support human-to-machine conversations at a worldwide level. Also, the current recall performance needs to be enhanced and to be reasonably more permissive - although not at the price of a lower precision.

4) Security Aspects: This point is still to be analyzed. If the underlying communication security between translation applications and smart devices (or their proxy gateway) is correctly implemented (e.g. with Transport Layer Security TLS and proper certificate management), it is our belief that the introduction of natural language as semantic support should not negatively impact the overall security of the global system.

Moreover, using natural language as the core of communication among all stakeholders (humans, machines and applications) should facilitate the traceability of operations. For example, a global thesaurus could be registered for further usages like support, forensic, search for intrusion, etc., where such thesaurus should be easily accessible to human operators or specialized applications.

VII. Conclusions and Future Directions

Humans have progressively and collectively developed natural languages (as a crowdsourcing exercise, as we would say today) not only to communicate information but also to support the elaboration of progressively more complex concepts [14].

The natural language approach of interoperability for simple things should be able to anticipate, then complement semantics-based standards (like W3C and oneM2M) and speed up the deployment of semantic IoT thanks to its inherent simplicity. Moreover, natural language evolves much more slowly than technical standards, and should provide extended lifetime and lower total cost of ownership for many IoT infrastructures.

We analyzed the characteristics of Natural Language Processing applied to machine-to-machine situations, and compared them to the human-to-machine case of chatbots. We demonstrated, with the smart gatekeeper example, that it is possible to obtain this performance (interoperability, quality, speed, light footprint) with a reasonably simple implementation.

Finally, it is likely that natural language interoperability, natively gathering people, connected devices and applications around the same unifying principle, would open new application fields and pave the way to still unexpected additional services. We would certainly need a more thorough analysis of verticals and use cases to discover what they might be in the future.

References

[1] L. Atzori, A. Iera, G. Morabito and M. Nitti, "The Social Internet of Things (SIoT) – When social networks meet the Internet of Things: Concept, architecture and network characterization." In Computer Networks, Volume 56, Issue 16, 14 November 2012, Pages 3594-3608

[2] Ahmed E. Khaled and Sumi Helal, "A framework for inter-thing relationships for programming the social IoT." In 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), Pages 670-675

[3] R. Girau, V. Pilloni and L. Atzori, "The Virtual User: the Holistic Manager of our IoT Applications." In 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), Pages 149-154

[4] M. McTear, Z. Callejas and D. Griol, "The Conversational Interface – Talking to Smart Devices." Springer, 2016, ISBN 978-3-319-32965-9 DOI 10.1007/978-3-319-32967-3

[5] The Open Connectivity Foundation - https://openconnectivity.org/

[6] The Web of Things - https://www.w3.org/WoT/

[7] oneM2M - http://www.onem2m.org/

[8] A. Čolaković and M. Hadzialic "Internet of Things (IoT): A Review of Enabling Technologies, Challenges, and Open Research Issues." In Computer Networks 144, July 2018

[9] McKinsey Global Institute, 2015 - The Internet of Things: Mapping the value beyond the hype.

[10] https://techcrunch.com/2017/05/04/report-smartphone-owners-are-using-9-apps-per-day-30-per-month/

[11] https://www.programmableweb.com/news/research-shows-interest-providing-apis-still-high/research/2018/02/23

[12] Davies, Mark. (2013) Corpus of News on the Web (NOW): 3+ billion words from 20 countries, updated every day. Available online at https://corpus.byu.edu/now/

[13] D. Braun, A.Hernandez Mendez, F. Matthes, M.Langen: "Evaluating Natural Language Understanding Services for Conversational Question Answering Systems" In Proceedings of the SIGDIAL 2017 Conference, pages 174–185

[14] F. de Saussure, Cours de linguistique générale, 1915, Introduction, Chapitre III, section 1.