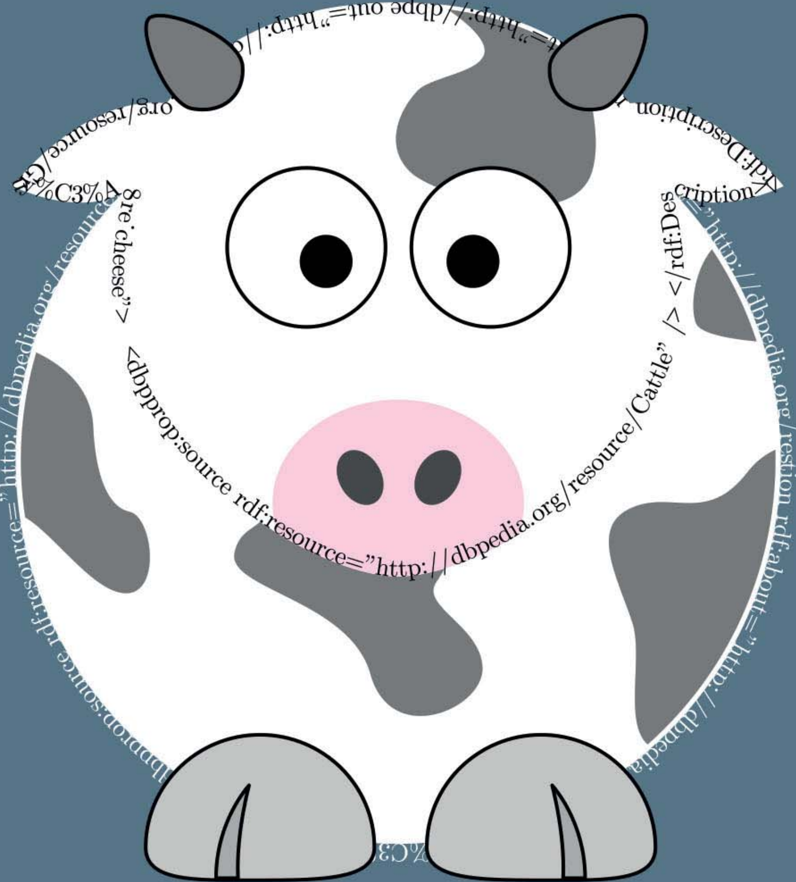
Henning Hasemann

Embedding Knowledge: Fusing the Internet of Things with the Semantic Web



Cuvillier Verlag Göttingen
Internationaler wissenschaftlicher Fachverlag

Embedding Knowledge: Fusing the Internet of Things with the Semantic Web

Von der Carl-Friedrich-Gauß-Fakultät
der Technischen Universität Carolo-Wilhelmina zu Braunschweig

zur Erlangung des Grades
Doktor der Naturwissenschaften (Dr. rer. nat.)
genehmigte Dissertation
von

Dipl.Inf. Henning Hasemann
geboren am 07.11.1983 in Bückeburg.

Eingereicht am: 07.08.2014
Mündliche Prüfung am: 15.12.2014
Referent: Prof. Dr. Alexander Kröller
Korreferent: Prof. Dr. Manfred Hauswirth

(2015)

Embedding Knowledge: Fusing the Internet of Things with the Semantic Web

Henning Hasemann

March 14, 2015

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

1. Aufl. - Göttingen: Cuvillier, 2015
Zugl.: (TU) Braunschweig, Univ., Diss., 2015

Nonnenstieg 8, 37075 Göttingen
Telefon: 0551-54724-0
Telefax: 0551-54724-21
www.cuvillier.de

1. Auflage, 2015
Gedruckt auf umweltfreundlichem, säurefreiem Papier aus nachhaltiger Forstwirtschaft

ISBN 978-3-95404-941-7
eISBN 978-3-7369-4941-6

Abstract

In the past years, much scientific interest has targeted the *Internet of Things* (IoT)—that is, the idea of creating a better connectivity between embedded devices and the Internet. This is a challenging endeavor: The upcoming multitude of heterogeneous devices, deployment scenarios and applications demands integration on different layers in a way that is extensible and future-proof. Standardization work is in progress to achieve interoperability of various device classes up to the application layer. However, so far the representation of data about applications, devices, and their surroundings is standardized only within enclosed knowledge domains. Thus, for data exchange with future applications or devices that report information from different knowledge domains, explicit adaptation is necessary which is a major hindrance for improvement of connectivity.

Paralleling the recent IoT advances, we could witness the dawn of the *Semantic Web*. This term describes a series of standards and conventions which allow a universal representation of knowledge distributed in the Web. This knowledge is extensible to any knowledge domain and vast open databases of various domains are available online as of today.

In this work we address the question on whether the problem of a common and universal data layer for the Internet of Things can be solved using the technologies of the Semantic Web. In particular, we focus on the question to what extent it is possible to store, process and query semantic data in line with the ideas of the Semantic Web directly on a variety of resource-constrained embedded devices. We provide a complete reference implementation to assess feasibility of our approach and its cost in terms of resources such as energy and storage space. We show that by consequent semantic abstraction it is possible to provide a system which describes observed real-world objects rather than individual sensing devices. This eliminates the need for an application to address specifics of the sensor deployment or other involved data sources.

This newly acquired universality accompanies a certain overhead in terms of resources due to a necessary increase in verbosity. We show however, that processing knowledge of high abstraction levels directly on the devices also allows for semantically-informed sleep scheduling approaches that can conserve energy.

This work is structured as follows:

In Chapter 2 we discuss the annotation of resource-constrained embedded devices in accordance with the ideas of the Semantic Web and compare our approach to existing knowledge representation standards in terms of fitness for the interoperability demands posed by the Internet of Things. We show, that—in contrast to existing approaches—expressing knowledge on embedded devices using the technologies of the Semantic Web can address the concerns of interoperability of devices in the IoT and connect it to a vast existing distributed knowledge database in the current Internet.

In Chapter 3 we address the technical feasibility of this approach for resource-constrained devices. We introduce the *Wiselib RDF Provider*, a modular and lightweight storage- and communication system for RDF annotations on embedded devices. We show that the Wiselib RDF Provider indeed makes it possible to store and process semantic information directly on a wide range of resource-constrained devices and analyze its efficiency in terms of compression ratio and energy consumption experimentally.

In Chapter 4 we consider the problem of abstracting over specific deployment scenarios in order to make applications reusable in different contexts. For this, we extend the semantic descriptions of devices to automatically generated descriptions of observed objects, called *Semantic Entities*. We illustrate how Semantic Entities ease application development as they make the application independent from the deployment. Furthermore, we show that by allowing the Semantic Entity abstraction to take place within the embedded network itself, we can utilize high-level abstract knowledge to facilitate a self-stabilizing, energy-conserving sleep scheduling scheme. We provide a reference implementation of the system and evaluate its workings in a realistic environment.

In Chapter 5 we examine how transfers of annotations to a central base station can be reduced to a minimum by processing queries to information in a distributed way in the embedded network. We introduce a query processing scheme and provide a reference implementation. Our system allows to process complex queries that access data sources from the embedded network as well as the Semantic Web with little energy drain to the embedded devices. We verify experimentally, that in a realistic environment our approach can answer complex semantic queries that span the embedded network and the Semantic Web with comparatively little energy consumption on the embedded devices.

Zusammenfassung

Das *Internet der Dinge* (engl. *Internet of Things*, IoT), also die Vision einer verbesserten Konnektivität von eingebetteten Systemen und dem Internet, ist in den vergangenen Jahren verstärkt in das wissenschaftliche Interesse gerückt. Diese motiviert ein anspruchsvolles Unterfangen: Durch die aufkommende Vielzahl an heterogenen Endgeräten, Installationsszenarien und Anwendungsanforderungen ist es notwendig, die Integration auf mehreren Schichten voranzutreiben, so dass Erweiterbarkeit und Zukunftssicherheit gewährleistet sind. Aktuell laufende Standardisierungsbemühungen streben eine Integration verschiedener Geräteklassen bis hinauf zur Anwendungsschicht an; die Darstellung von Anwendungsdaten, Geräte-Metadaten und Daten über die Außenwelt der Geräte ist allerdings nur innerhalb geschlossener Wissensdomänen standardisiert. Um Datenaustausch mit zukünftigen Anwendungen und Geräten aus verschiedenen Wissensdomänen sicherstellen zu können, ist eine explizite Adaptierung zwischen den einzelnen Diensten notwendig. Dieser Umstand stellt derzeit eine wesentliche Hürde für die Verbesserung der Konnektivität im IoT dar.

Parallel zu den Entwicklungen im IoT konnten wir die Entstehung des *semantischen Webs* (engl. *Semantic Web*) beobachten. Dieser Begriff beschreibt eine Reihe von Standards und Konventionen, die es erlauben, Wissen im Web auf universelle und verteilte Weise darzustellen. Wissen im Semantic Web ist nicht auf eine Wissensdomäne beschränkt, zudem existieren bereits enorme Wissensdatenbanken aus verschiedenen Anwendungsbereichen.

In dieser Arbeit untersuchen wir die Frage, ob das Problem einer gemeinsamen und universellen Wissensdarstellung für das IoT mit den Technologien des Semantic Web gelöst werden kann. Insbesondere erforschen wir, inwieweit es möglich ist, semantisches Wissen im Sinne des Semantic Web auf verschiedenen ressourcenbeschränkten Systemen zu speichern, zu verarbeiten und abzufragen. Um die Kosten unseres Ansatzes im Bezug auf Energie- und Speicherverbrauch hin untersuchen zu können, stellen wir eine komplette Referenzimplementierung zur Verfügung. Wir zeigen, dass es durch konsequente semantische Abstraktion möglich ist, über Beschreibungen von eingebetteten Systemen hinaus Beschreibungen von Objekten der realen Welt zur Verfügung zu stellen. Durch diese Abstraktion entfällt die

Notwendigkeit für Anwendungen, die Einzelheiten des Installationszenarios der eingebetteten Geräte oder anderen Datenquellen zu adressieren.

Diese neugewonnene Universalität bringt einen zusätzlichen Ressourcenverbrauch mit sich, dem eine Erhöhung des Datenaufkommens zugrunde liegt. Wir zeigen, dass es durch die Verarbeitung von semantischem Wissen auf den eingebetteten Geräten selbst möglich ist, semantisch informiertes Sleep-Scheduling zur Verfügung zu stellen, welches seinerseits den Energieverbrauch reduzieren kann.

Diese Arbeit ist wie folgt gegliedert:

In Kapitel 2 betrachten wir die Annotation von eingebetteten Geräten mit Technologien des Semantic Web. Wir stellen vorhandene Datenformate für eingebettete Geräte vor und vergleichen diese mit der semantischen Annotation im Bezug auf die Anforderungen des IoT. Wir zeigen, dass die Verwendung von Semantic-Web-Technologien auf eingebetteten Geräten diese Anforderungen lösen kann.

In Kapitel 3 adressieren wir die technische Umsetzbarkeit von semantischen Annotationen für Geräte mit beschränkten Ressourcen. Wir stellen den *Wiselib RDF Provider* vor, ein modulares, leichtgewichtiges semantisches Datenbank- und Kommunikationssystem für eingebettete Systeme. Wir zeigen, dass der Wiselib RDF Provider das Verarbeiten von semantischen Informationen auf einer Vielzahl von ressourcenbeschränkten Geräten ermöglicht und untersuchen seine Effizienz im Bezug auf Speicher- und Energieverbrauch.

In Kapitel 4 betrachten wir die Abstraktion von Beschreibungen konkreter Geräte-Installationen hin zu Beschreibungen beobachteter Objekte, den *Semantic Entities*. Wir zeigen auf, wie die Semantic-Entities-Abstraktion die Anwendungsentwicklung vereinfacht, indem es Anwendungen von dem konkreten Installationsszenario entkoppelt. Des Weiteren zeigen wir, dass es möglich ist, durch Verlagerung der Semantic-Entity-Erstellung in die eingebetteten Geräte semantisches Wissen für energiesparendes Sleep-Scheduling zu nutzen. Wir führen ein semantisch gestütztes, selbststabilisierendes, energiesparendes Verfahren ein, stellen eine Referenzimplementierung zur Verfügung und werten unseren Ansatz in einem realistischen Szenario aus.

In Kapitel 5 untersuchen wir die Möglichkeit, Datentransfers von eingebetteten Geräten zur Basisstation auf ein Minimum zu reduzieren. Dazu werden Anfragen an das System nicht mehr zentral beantwortet, sondern in Teilanfragen zerlegt und verteilt im Netzwerk verarbeitet. Wir stellen solch ein System zur Anfrageverarbeitung inklusive Referenzimplementierung vor. Unser System kann komplexe Anfragen verarbeiten, die sich gleichzeitig auf Datenquellen im eingebetteten Netzwerk und im Semantic Web beziehen können und hält dabei den Energieverbrauch der eingebetteten Geräte gering. Wir weisen experimentell nach, dass unser Ansatz in einem realistischen Szenario praktisch anwendbar ist und im Vergleich zu einer zentralisierten Datenverarbeitung deutlich weniger Energie verbraucht.

Acknowledgments

Although this work deceivingly carries a single author's name, it would not have been possible without the support of several other people.

First in line is undoubtedly my PhD adviser and mentor, **Alexander Kröller** who understood for the past four years to see my strengths and steer my view into the right direction. In addition to valuable scientific discussions, I drew a great part of my inspiration and motivation from you, thanks!

I do not even know where to start to thank **Tobias Baumgartner** and **Max Pagel**, who (literally) opened the door for me into the working group by welcoming me in their midst on a professional and personal level right from the start. Thanks for trustfully handing me over the Wiselib, for fruitful collaboration, being supportive and reliable colleagues, for a good time and of course for the proofreading work. Thanks go to the whole Algorithms working group for providing a cozy yet productive working atmosphere even in tense situations. In particular, **Sabine Anthony** and **Ute Marchot**, both of which not only sometimes compensated for my lack of skills of handling paper work, but also were a source of constant encouragement. **Christiane Schmidt**, who provided valuable help for theoretical problems, thanks for the discussions and of course the proofreading. Thanks to **Stephan Friedrichs** for scientific discussions, the music, the revitalizing coffee breaks, and the proofreading.

This work was conducted in the context of the SPITFIRE project, in which I could not have been productive without the support of a number of persons such as **Christian Tille**, who implemented several features for Coalesenses iSense devices that were necessary to make many components presented in this work possible and usable and also **Oliver Kleine**, the main author and maintainer of the Smart Service Proxy which constitutes another vital precondition for this work. Thanks to both of you for the many hours of hacking we spent on balconies together getting things working! Neither shall I fail to mention **Myriam Leggieri**, who helped me getting a better understanding of the wonderful world of the Semantic Web—thanks for the helpful discussions and the proofreading!

Thanks also to **Rebecca Finster**, **Marc Jandt** and **Stefan Winterfeldt** for proofreading. Last but not least, thanks to my family and friends for their constant support during the past years in more ways than I can count.

You rock.

I wish I never met you doctor,
I was much better off as a coward.

Captain Jack Harkness, Doctor Who

Contents

Chapter 1

Introduction

Just looked like a "thing", didn't it? People don't question "things". They just say, "oo... it's a thing."

The Doctor, Doctor Who

In the past years we witnessed a development leading from the data-collecting *Wireless Sensor Networks* (WSNs) towards the *Internet of Things* (IoT). This term, which is believed to be first publicly introduced in a presentation of Kevin Ashton in 1991 [Ash], nowadays stands for different ideas, depending on the source of its definition. Ashton's definition strives to connect real-world objects to the Internet, much in the spirit of earlier term *Ubiquituos Computing* [Wei91]. The "things" in his definition do not refer to embedded computing devices, but to actual real-world objects they observe and control. Another accepted definition [HTM+14] stresses a new level of connectivity between devices, exceeding that of existing *Machine-to-Machine* (M2M) communication approaches and allowing machines and humans to exchange data. Both definitions of the term are inherently related to embedded and mobile technologies and devices such as wireless sensor nodes, smart phones, smart watches, wearables, smart TVs, home appliances and a wide array of new applications and devices focusing on observation of and interaction with the physical world. Thomas Liesner quotes in his article "The Internet of Things – next Revolution or Smooth Transition?" [Lie] data from Gartner, Inc. [Gar] and others predicting a rapid growth of in the number of connected devices of this type (see Figure 1.1).

We see a large variety of examples of (consumer-targeted) IoT devices presented in the last years fitting Ashtons definition, some shown in Figure 1.2: The Netamto Weather Station allows to measure several environmental properties such as air temperature, humidity, CO_2 level and noise [PD]. The Belkin WeMo Insight Switch allows to remote-control the power supply to power-plugged devices and assess their energy consumption [Bel]. The Jawbone UP24 wrist band can measure

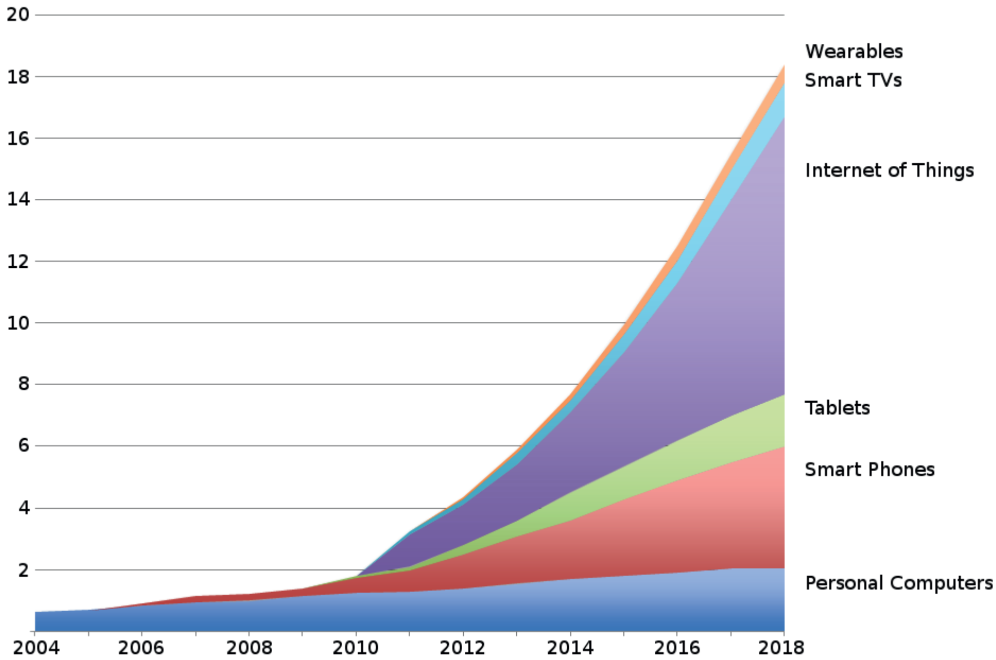


Image: *http://www.kaizen-factory.com/2013/10/26/the-internet-of-things-next-revolution-or-smooth-transition/*
(labels edited for readability)

Figure 1.1: Connected IoT devices (in billion). Values from 2013 onwards are predictions.

personal activity during night- and daytime and monitor sleeping behavior as well as aid in working out [Jaw]. All these devices come bundled with smart phone- or cloud-based applications that make them more useful to the end user: The Netatmo Weather Station application uses the measured values to provide warnings about air quality and a sophisticated user interface that allows to analyze weather history and forecast future weather conditions. The WeMo Insight application allows to track energy consumption of a connected appliance and can, for example, estimate the monthly energy cost for that appliance. Additionally, devices can be power-cycled remotely via smart phone or automatically by the time of day. The UP24 smart phone application is specifically designed to set personal goals for improving habits in terms of sleeping, diet or training.

Each of these devices are sophisticated, useful products on their own, some of which connect to the Internet to increase the value of the application by incorporating additional data sources or making use of centralized cloud storage. Vendors of consumer articles seem to agree on the use of established protocols like 3G, Bluetooth, Zigbee and WLAN on the low layers and thus usually integrate easily into existing home networks. To that end, there is a general trend of the usage of standardized communication protocols established in practice. However, on higher layers the formats of data exchange are usually vendor-specific, not always publicly documented and targeted at machine-to-user communication.

Image: *http://netatmo.com* Image: *https://belkin.com* Image: *https://jawbone.com*

Figure 1.2: Examples of current IoT products. From left to right: The Netatmo Weather Station, the Belkin WeMo Insight Switch, the Jawbone UP24 wrist band.

At the same time, there is a strong and growing interest for integration of these diverse IoT components with each other and with services and applications on the Web on a data layer. This trend is witnessed by recent practical integration approaches (Figure 1.3): The web service *"If this then that"* (IFTTT) [IFT], provides its users with the means to create simple rules that connect triggers from different *channels* to actions on other channels. At the time of this writing, IFTTT supports 104 distinct channels, including popular web services such as Facebook [Fac], YouTube [You] or GitHub [Git] and IoT appliances such as the Belkin WeMo Insight Switch, the Jawbone UP24 wrist band and Netatmo Weather Station introduced above. Example rules could switch off a WeMo plug whenever the UP24 detects sleep, or post a Facebook update when the user achieves its step goal as measured by UP24. Another practical integration approach is presented by the *Ninja Sphere* [Nin]: The Ninja Sphere can track objects and users and provide its users with location-based services. It allows, for example, to inform the user about a left-on heating device and give him/her (via smart phone) the possibility to turn it off remotely, or for a user to control the light in the room he/she currently is in via his/her smart watch. The Ninja Sphere supports a variety of IoT devices such as the Belkin WeMo, the Phillips Hue [Kon] and the Pebble smart watch [Peb].

In addition to this movement of integrating existing IoT appliances with each other we also witness a development towards more user-controlled, multi-purpose IoT sensing devices, targeted at user-customized installation, tinkering and even user application development. The *Ninja Blocks* [Nin] system provides a variety of wireless sensors that can sense temperature, motion or button presses. The vendor specifically encourages tinkering with the devices and installation of user-provided embedded or centralized applications and provides full compatibility with the pop-

Image:
http://www.kickstarter.com/projects/ninja/ninja-sphere-next-generation-control-of-your-envir

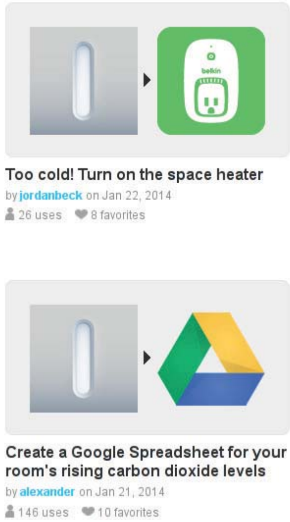


Image:
https://ifttt.com/netatmo

Figure 1.3: Integration of IoT components. Left: Prototype of the "Ninja Sphere". Right: Example IFTTT rules involving the Netatmo Weather Station.

ular Arduino [Ard] platform. The *VARIABLE NODE+* [Var] provides a wireless sensor platform with exchangeable sensor- and actuator plugs and can be used for a variety of applications such as reading bar codes, measuring light, air- or surface temperature and air composition, shown in Figure 1.4. Additionally, the vendor provides an *Input / Output* plug, that provides controllable I/O pins for custom user applications and extension. These approaches of empowering the end users in using their IoT devices and building custom applications pose new demands on integrability of devices and applications: Whereas in the formerly presented approaches devices and software applications were shipped as an atomic bundle, for this type of devices the application it not known a priori. What data will the user or application access? Where should it be communicated? What other devices is the device going to interact with? These questions stress the second meaning of the term "Internet of Things" we introduced: A high degree of connectivity between different devices and applications. The idea of carefully integrating the multitude of upcoming IoT appliances into platforms such as IFTTT or the Ninja Sphere one by one however does not scale in the long term: Code has to be added specific for the API of each new appliance and user-defined applications (such as IFTTT rules) are not reusable in that they still refer to specific products and vendors.

Consider a simple, user-defined home automation application: A contact sensor observes the open–closed state of a window. Whenever the window is being opened, the smart power plug should cut the power to the air conditioning device to avoid a waste of energy. A possible, IFTTT-like formulation could be *"IF sensors 42*

Images: *https://variableinc.com*

Figure 1.4: VARIABLE NODE+ with some of the available exchangeable sensor plugs.

measures contact loss THEN turn off power plug 23". Ideally, we would rather like to express our applications in the spirit of *"IF a window is open in any room $x THEN turn off all AC devices in room $x"*. Note how much more generic and reusable this second formulation is: It can work with any number of window sensors and any number of AC devices and matches them to the same room. More important, it does not include (a) any implicit assumptions about where sensors are installed and what they are observing (b) access to the raw sensor information ("contact loss"). This level of abstraction, considered at a general level, requires the following preconditions:

1. A description of the sensor and actuator devices: What are they observing? What does that mean for the application? Where are they located? How can they interact with the real world?

2. An infrastructure that allows to obtain the information asked for in the query: As the second query does not address sensors explicitly, other means of accessing this information need to be provided.

We pose the following questions: How can we express—on the data layer—such information in a way that any thinkable future application scenario remains possible, yet still have it be descriptive enough to enable reuse of knowledge across different applications and devices? The simple query shown above only accesses information about sensors, how can we incorporate knowledge from remote databases or data publicly available on the Web such as company time tables, large-scale location information or weather services? What is the cost of such a knowledge representation, and where can this knowledge reside? If queries do not address specific devices anymore, how can we still make this knowledge accessible?

A lot of research is currently in progress that aims to address these and similar questions from different perspectives: The European Union project *Internet of Things Architecture* (IoT-A) [BBB+12] was started in 2010 and concluded in November 2013. IoT-A's mission was to create architectural foundations for the Internet of Things for addressing questions like integration, self configuration and orchestration. Figure 1.5 gives an overview over a fraction of their outcomes,

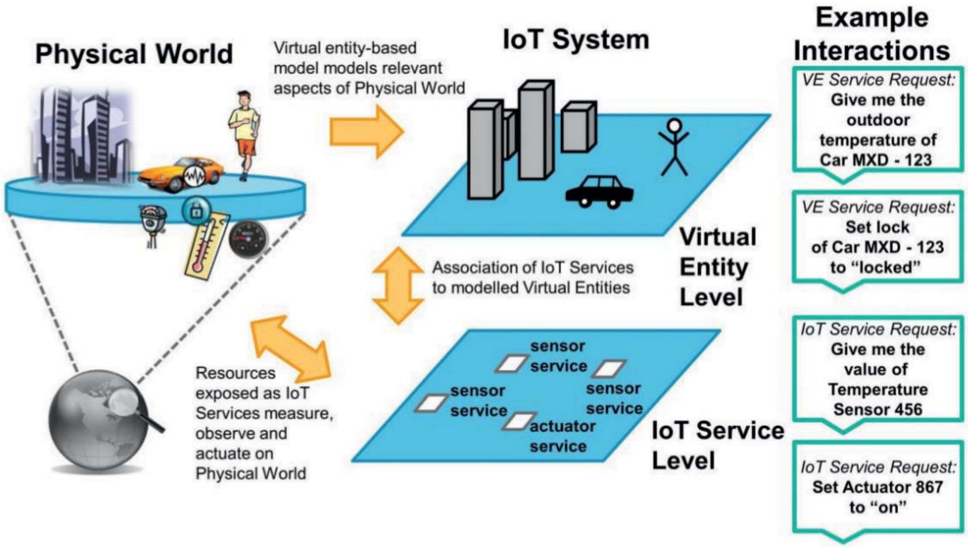


Image: [BBB+12]

Figure 1.5: IoT-A deliverable D1.5: IoT service and Virtual Entity abstraction levels.

namely the conceptual integration of the physical world, the several IoT services and the IoT system with its *Virtual Entities*—virtual representations of observed or controlled real-world objects. The EU project *iCore* [iCo] implements a realization of the IoT-A architecture, abstracting over device heterogeneity and individual devices towards a representation of the observed real-world objects.

In parallel, the European Union funded the project *Semantic Service Provisioning for the Internet of Things using Future Internet Research by Experimentation* (SPITFIRE) [SPIa]. SPITFIRE addresses this set of questions by consequently connecting the IoT to the *Semantic Web* [BLHL01] and emphasizing the elevation of embedded devices to self-describing first-class citizens of the future Internet of Things.

Thesis Outline. This thesis addresses the question of how Semantic Web technologies can be used to describe IoT devices efficiently to provide a better integration. In Chapter 2 we analyze existing standards for describing embedded devices with respect to their expressiveness and universality. We introduce the Semantic Web and show how the description methodology it offers can improve on these existing standards. Chapter 3 discusses the storage of semantic descriptions directly on the resource-constrained embedded devices with a focus on determining the overhead, especially in terms of energy consumption of such a data model. In Chapter 4 we address the question of how we can further abstract from the description of embedded devices discussed so far. In particular, we raise the question of how these device descriptions can be converted into knowledge about the