Human Empowerment in a Semantic Web of Things
Concept of a semantic platform for connected devices

Jens Meder
Abstract

Human Empowerment in a Semantic Web of Things

Jens Meder

The Internet of Things is a vision of autonomous devices that can gather, process, and transmit information without human intervention. The power of the Internet of Things lies in the connections between devices and internet resources. They allow humans to delegate automatable tasks to devices and internet services so that they can execute them. Yet, there is no common platform for connecting these resources today.

Recent studies show that human autonomy, agency, and social injustice are of concern to users when asked about Internet of Things scenarios. Yet, research focuses on technological advancements, making internet enabled devices more autonomous instead of addressing needs and concerns of users.

This thesis identifies requirements for a tool, that allows layman users to create connections between connected devices and internet resources. A Theory of Use incorporates these requirements and outlines qualities of a good solution which are then refined by sketches. The resulting hypothesis incorporates these qualities as well as the principles of designing for appropriation to support adaptation and adoption by users.

The outcome of this thesis is a framework that puts human needs first, supporting human autonomy, agency, and social justice. It includes concepts of the Semantic Web of Things for easy integration of devices and internet services and allows both humans and machines to make sense of the underlying information.
Table of contents

Acronyms  7
1. Introduction  9
  1.1 Objectives & Delimitations  10
2. Background  11
  2.1 Towards a Semantic Web of Things (SWoT)  11
    2.1.1 Internet of Things (IoT)  11
    2.1.2 Web of Things (WoT)  11
    2.1.3 Semantic Web (SW)  11
    2.1.4 Semantic Web of Things (SWoT)  12
  2.2 Ethical concerns  12
    2.2.1 Autonomy  12
    2.2.2 Human agency  13
    2.2.3 Correlation between autonomy and agency  14
  2.3 Empowering users  14
    2.3.1 The importance of semantics  15
    2.3.2 The road to human empowerment in the IoT  15
  2.4 Summary  16
3. Methods  17
  3.1 The problems with user-centered design  17
  3.2 Designing for appropriation  17
  3.3 Theory of Use Protocol  18
  3.4 Sketching and prototyping  19
  3.5 Summary  20
4. Results  21
  4.1 ToU  21
    4.1.1 Model  21
    4.1.2 Qualities of a good solution  22
  4.3 Hypothesis: The Butler Framework  22
    4.3.1 Resource Types  23
    4.3.2 Abilities & Interests  24
    4.3.3 Conversations  24
    4.3.4 Butlers  24
# Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>BLE</td>
<td>Bluetooth Low Energy</td>
</tr>
<tr>
<td>CASAGRAS</td>
<td>Coordination and support action for global RFID-related activities and standardisation</td>
</tr>
<tr>
<td>CSS</td>
<td>Cascading Style Sheets</td>
</tr>
<tr>
<td>HTML</td>
<td>Hypertext Markup Language</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>ISO</td>
<td>International Standardization Organization</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
</tr>
<tr>
<td>OWL</td>
<td>Web Ontology Language</td>
</tr>
<tr>
<td>RDF</td>
<td>Resource Description Framework</td>
</tr>
<tr>
<td>REST</td>
<td>Representational State Transfer</td>
</tr>
<tr>
<td>SW</td>
<td>Semantic Web</td>
</tr>
<tr>
<td>SWoT</td>
<td>Semantic Web of Things</td>
</tr>
<tr>
<td>ToU</td>
<td>Theory of Use</td>
</tr>
<tr>
<td>URI</td>
<td>Universal Resource Indentifier</td>
</tr>
<tr>
<td>W3C</td>
<td>World Wide Web Consortium</td>
</tr>
<tr>
<td>WoT</td>
<td>Web of Things</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>
Human Empowerment in a Semantic Web of Things

1. Introduction

Ever since I got my first entertainment cabinet I have been fascinated by the way it handles complexity: A set of devices, each device with its dedicated function and cables that connect them. Without connections, each device is of minor use, but once connected, they unleash their power.

The Internet of Things (IoT), a collection of connections between internet enabled devices and internet resources, is yet another entertainment cabinet. “The important part of a connected device is not the device. It’s what the device connects to.” (Dunn, 2013). Hardware is just a means to an end. It “helps your software ask better questions” (Dunn, 2013).

Connections are essential to support human high level goals (e.g., reducing energy consumption). They allow us to combine capabilities of otherwise independent devices. As long as we are aware and in control of connections, we can troubleshoot them (e.g., by checking the physical connections). If we are unaware of connections we are unable to predict effects, let alone recover from errors.

Today, combining capabilities of connected devices requires programming skills, thereby ignoring layman users. These users depend on the skills and intentions of professionals with few options to influence them. Every human has individual requirements. Something that seems appropriate for a developer can be inappropriate for end-users (Cooper et al., 2012, p. 3).

Artifacts and technologies influence our decisions and shape our everyday life (Johnson et al., 2009). If we de-skill and replace humans with technological devices, we have to increase the functionality of these devices to resemble human actions (Johnson et al., 2009). Professionals are the once to execute this process, thus they influence the way artifacts shape our lives and behavior.

In May 2014, Google presented their vision of future transportation: a self driving car (Simonite, 2014). They determined that humans are unable to be co-pilots in semi-automated cars, hence they removed all controls to prevent human intervention and replaced them with redundant systems. But, if such a vehicle has to cope with a situation unforeseen by professionals, humans are unable to intervene - “Anything that can go wrong will go wrong“ (Murphy’s Law, n.d.).

If the aforementioned issues are left unaddressed, connections between IoT enabled devices and internet resources will depend solely on the abilities and intentions of professionals. Layman users are de-skilled and unable to protect themselves, which creates social injustice and undermines their autonomy and agency.

Empowering users in the IoT seems counterintuitive to what Kevin Ashton (2009) envisioned when he coined this term. His vision was to empower
devices to autonomously collect and process information. Yet, these two visions are not mutually exclusive.

Empowering devices assumes that they provide means to achieve humans’ higher level goals. Thus, empowering devices means to empower humans to achieve their goals by delegating tasks to devices. If users have the ability to delegate tasks, we support their agency. Additionally, if they also have the freedom to define goals, we support their autonomy.

If professionals define connections between devices, users are neither agentic nor autonomous. Thus, users, not professionals, should have the power to decide which connections serve their purpose. Instead of reshaping users’ behavior we should allow *them* to shape artifacts. They need tools to setup and customize connections without programming skills, thus allow gradual achievement of mastery.

### 1.1 Objectives & Delimitations

Today, a standard for connecting internet enabled devices and internet services is unavailable. Vendors only provide Application Programming Interfaces (API) that require programming skills to use them. Without these skills, establishing connections is impossible.

The aim of this thesis is to identify needs, desires, and concerns of layman users in IoT environments. These requirements are used to design an IoT platform, to set-up and manipulate connections without programming skills.

Technical details such as encryption, efficient resource management, optimized data transmission and storage, etc., are considered out of scope. The selection of sketching tools, libraries, protocols, etc., are based on my prior knowledge rather than criteria deduced from requirements.
2. Background

2.1 Towards a Semantic Web of Things (SWoT)

2.1.1 Internet of Things (IoT)

According to Ashton (2009) today’s computers and the internet consist of and depend on human generated information. Processing such optimized web resources with machines is prone to errors. Thus, integrating devices in such an environment requires a shift in information representation. Based on this observation, the Auto-ID group around Kevin Ashton envisioned a scenario in which computers are empowered to autonomously collect information (Ashton, 2009). Their vision was „a world in which all electronic devices are networked and every object, whether it is physical or electronic, is electronically tagged with information pertinent to that object.” (Sarma et al., 2000, p. 4).

The Cisco Internet Business Solutions Group (IBSG), defines the IoT as “the point in time when more ‘things or objects’ were connected to the Internet than people.” (Evans, 2011, p. 2). According to Cisco, this point has been reached between 2008 and 2009.

I refer to the IoT as a network of real world objects with unique identifiers, that leverages the existing internet infrastructure to allow communication between these objects and internet resources. (EU FP7 Project CASAGRAS, 2009)

2.1.2 Web of Things (WoT)

The Web of Things (WoT) is a scenario in which web services and real world objects communicate with each other using web technologies, e.g., the Hyper Text Transfer Protocol (HTTP) (Vesyropoulos et al., 2013). It is the next phase of evolution of the IoT. The difference lies in the terms internet and web. Internet refers to the physical layer of network architecture (e.g., switches, routers) whereas web refers to the application layer on top of it (e.g., HTTP).

2.1.3 Semantic Web (SW)

Today’s Web content is designed for humans (Berners-Lee et al., 2001) with Hyper Text Markup Language (HTML), JavaScript and Cascading Style Sheets (CSS). The lack of distinctly defined semantics and context makes machine processing unreliable. The Semantic Web is a set of technologies, that structure and annotate web content into semantic constructs, so that both humans and machines can reason upon them.

In general, semantics represent „the meaning or relationship of meanings of a sign or set of signs” (Semantics, n.d.). In the context of the SW they are structured information sets with interference and mapping rules (Berners-
Lee et al., 2001). Using these constructs we can define things (e.g., persons, locations) with relationships (e.g., lives in) and corresponding values (e.g., another person).

Ambiguity is a problem of the aforementioned structures. Imagine two databases, each referring to a Person but with different identifiers, e.g., home and living at. Comparing or mapping these structures without interference rules is impossible. Ontologies provide a solution to that problem.

In philosophy, *ontology* refers to a “theory of the nature of existence” (Gruber, 2009). In computer science it “defines a set of representational primitives with which to model a domain of knowledge or discourse” (Gruber, 2009). In the context of semantic web, it refers to a formal definition of relations between terms. Typically, ontologies include a taxonomy that defines relationships among sets of objects (Berners-Lee et al., 2001) as well as a set of interference rules, e.g., the Web Ontology Language (OWL) (World Wide Web Consortium, 2012).

### 2.1.4 Semantic Web of Things (SWoT)

The Semantic Web of Things is an emerging field in IoT research, merging concepts from the SW and the WoT (Figure 2.1). The vision is to seamlessly integrate internet enabled devices with existing web resources through semantics.

![Figure 2.1: IoT terminology hierarchy](image)

### 2.2 Ethical concerns

#### 2.2.1 Autonomy

Autonomy can be defined as “freedom from external control or influence; independence” (autonomy, n.d.).

In 2013, the Joint Research Centre at the Institute for the Protection and Security of the Citizen of the European Commission has published a report on *Agency in the Internet of Things* (Pereira et al., 2013). The report analyses
major ethical concerns including agency, human autonomy, and social injustice.

The research group has issued an online questionnaire to identify general concerns related to the IoT. More than 600 answers (Table 2.1) point to concerns towards human autonomy and identity.

<table>
<thead>
<tr>
<th>Ethical concern</th>
<th>Agree</th>
<th>Disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>IoT threatens the protection of identity</td>
<td>&gt; 60 %</td>
<td>&lt; 20 %</td>
</tr>
<tr>
<td>IoT changes our definition of identity</td>
<td>&gt; 60 %</td>
<td>&lt; 20 %</td>
</tr>
<tr>
<td>IoT should operate under explicit consent – ‘autonomy’</td>
<td>≈ 80 %</td>
<td>≈ 10 %</td>
</tr>
<tr>
<td>Safeguard of autonomy should be sought if consent does not work</td>
<td>≈ 50 %</td>
<td>&lt; 20 %</td>
</tr>
<tr>
<td>IoT can interfere with individuals’ autonomy</td>
<td>&gt; 60 %</td>
<td>&lt; 20 %</td>
</tr>
<tr>
<td>IoT development shall not create social injustice</td>
<td>&gt; 80 %</td>
<td>≈ 5 %</td>
</tr>
<tr>
<td>Concern about IoT collected data</td>
<td>&gt; 80 %</td>
<td>&lt; 20 %</td>
</tr>
</tbody>
</table>

Table 2.1: Questionnaire results on concerns related to the IoT (Pereira et al., 2013, p. 30)

Pereira et al. (2013) conclude, that undermining human autonomy in IoT environments leads to potential unauthorized actions and social injustice. It can lure people into decisions controlled by technology rather than their moral and social norms. Those with knowledge of such technology are able to protect themselves, but laymen will be further de-skilled and become dependent on experts and their intentions.

2.2.2 Human agency

Human agency refers to “the capacity to exercise control over processes, motivation, action, and environment” (Jia et al., 2012, p. 1185).

Jia et al. (2012) have conducted a qualitative interview study to investigate the influences of the IoT on human agency and object / machine agency. Nine participants have been interviewed on their views, awareness, proficiency, fears, and expectations towards the IoT (Jia et al., 2012). The participants perceive current IoT technology as dominant and would like to have “more control (of the technology)” (Jia et al., 2012, p. 1187). The automatic nature of technology as well as manufacturers’ and developers’ opacity create further distrust and a feeling of being out of control.

Proxy agency, a special type of agency, “can be exercised by getting others with resources or expertise to achieve one’s own goal” (Jia et al., 2012, p.
The participants perceive IoT enabled devices and applications as proxy agents. They want them to be personal helpers or assistants supporting them with their daily problems. These helpers should be controllable via an interface integrated “into the devices that they already use daily” (Jia et al., 2012, p. 1186).

2.2.3 Correlation between autonomy and agency

In my opinion, the concepts of autonomy and agency are correlated, with agency being a precondition for autonomy. Without agency you are unable to act autonomously. Thus, autonomous agents act without external control.

Agency, on the other hand, can exist without autonomy. Non-autonomous agents act on behalf of others (proxy agents) or based on the influence and control of external forces with a limited set of choices.

For an in-depth discussion please refer to chapter 6.3.6 in (Laaksoharju, 2014).

2.3 Empowering users

Human empowerment has long been recognized as a fundamental goal when discovering new technologies (Shneiderman et al., 1990). Technology should support humans in the achievement of high level goals, e.g., a high degree of education (Shneiderman et al., 1990).

Ben Shneiderman (1990) proposes a Declaration of Empowerment to underline professionals’ responsibility in computer science towards user empowerment. This declaration includes three principles:

1. Professionals should recognize the influence of their actions and commit themselves to “enable users to accomplish their personal and organizational goals while pursuing higher societal goals and serving human needs” (Shneiderman, 1990, p. 2).

2. Professionals should prepare a Social Impact Statement for every project to determine potential users and training requirements, negative side-effects and monitoring procedures.

3. Professional organizations should prepare “an agenda of vital, specific, and realizable goals for the next decade” (Shneiderman, 1990, p. 3) to inspire our profession and others.

Based on these ideas, empowering users means supporting both their autonomy and agency. It “give[s] (someone) the authority or power to do something” (empower, n.d.) and “make[s] (someone) stronger and more confident, especially in controlling their life and claiming their rights” (empower, n.d.).
## 2.3.1 The importance of semantics

Humans and IoT resources such as sensors create huge amounts of data every single day. Making sense of them requires a transformation into meaningful information. Semantics provide means to provide such transformation by structuring and abstracting information for a given purpose. This information can then be used by both humans and machines.

The DIKW hierarchy (Table 2.2), also Knowledge Hierarchy, Information Hierarchy or Knowledge Pyramid, describes the transformation of data into wisdom from a human point of view (Rowley et al., 2007). Due to the absence of commonly agreed upon definitions, the following table illustrates my usage of the concepts based on Rowley’s (2007) survey of available definitions.

<table>
<thead>
<tr>
<th>Data</th>
<th>Unorganized and unprocessed facts or observations that are meaningless without context or interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Information</strong></td>
<td>Formatted and processed data for a specific purpose and context</td>
</tr>
<tr>
<td><strong>Knowledge</strong></td>
<td>“Information combined with understanding and capability” (Laudon &amp; Laudon, 2006, p. 2) that “exists along a continuum between tacit knowledge (know how) and explicit knowledge (know what)” (Jashapara, 2004, p. 17)</td>
</tr>
<tr>
<td><strong>Wisdom</strong></td>
<td>Acquired knowledge, that can be transferred from one domain to another. (Jessup et al., 2003)</td>
</tr>
</tbody>
</table>

**Table 2.2: Elements of the DIKW Hierarchy**

From a user's point of view, semantics provide the foundation for knowledge and wisdom. They allow users to decide how and when they want to delegate parts of their agency.

From a machine point of view, semantics allow different stakeholders to unambiguously access and interpret data (interoperability). They allow for data representation and management of the physical world (data abstraction and access), and the creation of new information and potentially knowledge through semantic reasoning. Semantics also help to integrate data into already existing processing chains (Barnaghi et al., 2012).

## 2.3.2 The road to human empowerment in the IoT

In a worst case scenario, the current trend in IoT research leads to autonomous agentic devices with layman users losing their agency. Since autonomy depends on agency, users lose their independence. But, without professionals there are no means to control agency in an IoT. The solution to this dilemma: balancing agency between humans and devices (Jia et al., 2012).
Users transfer parts of their agency to professionals, who in turn transfer it to software and devices, transforming them into proxy agents. Professionals acknowledge and support human agency, and provide tools so that users can control the degree of proxy agency in the IoT.

2.4 Summary

Threats towards human autonomy and agency concern people in IoT environments. Instead of decreasing human agency and autonomy, these environments should empower users to stay in control of the information exchange between devices and internet services. They should work as personal assistants or helpers executing tasks on behalf and with the explicit consent of their owners (proxy agents). Users dislike special configuration devices and prefer to leverage the devices they already own (Jia et al., 2012).

Semantics provide the basis for balancing the agency between humans and machines. They allow humans to make informed decisions when delegating parts of their agency. Devices use semantics for interoperability between resources, data integration, data abstraction and access, as well as semantic reasoning and interpretation.
3. Methods

“Everything is best for something and worst for something else.”


3.1 The problems with user-centered design

A user-centered design approach is an iterative process, that identifies users’ needs in a given context, creates possible solutions and evaluates them with users. These iterations stop when predefined criteria (e.g., efficiency, effectiveness and satisfaction according to ISO-9241) have been reached. Iterative refinement is suitable to make systems more usable, but does not contribute to the development of novel ones (Greenberg et al, 2008).

Usability evaluation techniques gauge designs or systems against requirements to ensure that they meet the needs of users (Dix, Finlay, Abowd & Beale, 1993). Evaluating usability makes sense if a product or technology has reached cultural and technical readiness and adoption by a critical mass of users (Greenberg et al, 2008). These techniques produce valuable results in well-defined and well-understood settings but can be harmful when applied to novel products and technologies. They lack consideration of the cultural context as well as the adoption over time when applied to innovative products (Greenberg et al., 2008). This cultural adoption is unpredictable but is an essential driver for product maturation and evolution (Greenberg et al., 2008). Evaluating such innovative products outside their cultural context “is almost pointless” (Greenberg & Buxton, 2008, p. 118).

Today, IoT enabled devices and applications are scarce. Cisco estimates that only around 0.6% of all available objects are connected to the internet today (Bradley et al., 2013). Based on these numbers we cannot assume a high degree of cultural adoption, let alone a well-understood and well-defined setting. Thus, a user-centered design approach and the evaluation of usability seem impractical.

3.2 Designing for appropriation

You might have seen people who are able to open a beer bottle with any object they have at hand. This act of using a tool contrary to its intended purpose is called appropriation. According to Alan Dix (2007), appropriation refers to the adaptation and adoption of tools or artifacts in ways their designers have never anticipated.

Such adaptations and adoptions occur when

- a given task requires non-existing tools or
the costs of learning or acquiring existing tools exceed the appropriation of tools or artifacts at hand.

Technology Appropriation Cycle framework (Carroll, Howard, Vetere, Peck, and Murphy, 2001) describes the process of technology appropriation. In this framework, technology falls into either of two categories: technology-as-designed (provided by the designer) and technology-in-use (embedded into users’ lives). “These are then linked through a process of appropriation whereby technology is either never seriously considered (non-appropriation) or taken on board selected and adapted by users (appropriation), but even if appropriated may at some stage be rejected (dis-appropriation).” (Dix, 2007, p. 28)

Designing for appropriation means to “design so that people are more likely to be able to use what you produce for the unexpected – they do the final ‘design’ when the need arises.” (Dix, 2007, p. 28). This can be achieved by designing according to the following principles:

- Let users define their own meaning for certain elements (allow interpretation).
- Supply enough information, that users can anticipate the effects of actions (provide visibility).
- Make design decisions understandable for users (expose intentions).
- Provide alternatives for task completion (support not control).
- Allow the user to combine parts in different ways. (plug-ability and configuration)
- Let more confident and technically advanced users share their appropriations (encourage sharing).
- Observe how people have adapted and adopted technology and identify needs for a specialized tool (learn from appropriation).

3.3 Theory of Use Protocol

Mikael Laaksoharju (2014) proposes a “deductive theorizing protocol based on falsification, inspired by Karl Popper” (Laaksoharju, 2014, p. 164). The result of this protocol is a description of a given system and its mechanics (model), that explains the intention of such description (theory) and provides specific predictions motivated by current knowledge (hypothesis). It is an explication of a designer’s understanding of the problem (Theory of Use) that can be scrutinized by users.

A Theory of Use (ToU) introduces the identified problem, includes qualities of a good solution from the designer’s point of view as well as assumptions of the usage situation and the users. These qualities and assumptions take the form of statements to allow falsification. This ToU is then presented to users
so that they can react to it. Based on these reactions, the ToU is then revised until it is entirely accepted.

Based on the ToU, alternative designs are created. These can be considered hypotheses claiming to solve the problem. These hypotheses are then put up for reactions and might result in a further revision of the ToU to rectify identified errors.

### 3.4 Sketching and prototyping

In *Sketching User Experiences*. *Getting the design right and the right design*, Bill Buxton (2010) discusses the important differences between sketches and prototypes. According to Buxton, sketches and prototypes differ on four attributes: costs, timeliness, disposability and quantity.

Sketches are cheap to create, easy to dispose, can be built when needed and come in numbers. Prototypes, on the other hand, are more expensive to create, harder to dispose due to their refinement, require more time to make and come in small numbers.

Sketches and prototypes can be seen as two ends of a continuum. An artifact lies somewhere in between these two extrema.

<table>
<thead>
<tr>
<th>Sketch</th>
<th>Prototype</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evocative</td>
<td>Didactic</td>
</tr>
<tr>
<td>Suggest</td>
<td>Describe</td>
</tr>
<tr>
<td>Explore</td>
<td>Refine</td>
</tr>
<tr>
<td>Question</td>
<td>Answer</td>
</tr>
<tr>
<td>Propose</td>
<td>Test</td>
</tr>
<tr>
<td>Provoke</td>
<td>Resolve</td>
</tr>
<tr>
<td>Tentative</td>
<td>Specific</td>
</tr>
<tr>
<td>Noncommittal</td>
<td>Depiction</td>
</tr>
</tbody>
</table>

*Table 3.1: Sketch to Prototype continuum (Buxton, 2010, p. 140)*

When Buxton talks about sketches he includes everything that posses a certain set of qualities to explore the design space. According to Buxton sketches ...

... can be made quickly (*quick*).

... can be made when needed (*timely*).

... are cheap to make (*inexpensive*).

... can be disposed if no longer needed (*disposable*).

... are part of a series or collection to convey their meaning (*plentiful*).
... reinforce their nature through their representation (clear vocabulary).

... suggest openness and freedom instead of precision (distinct gesture).

... include only necessary details (minimal detail).

... reflect the level of certainty in the designer’s mind (appropriate degree of refinement).

... do not answer questions but suggest possible directions (suggest and explore rather than confirm).

... leave room for interpretation (ambiguity).

3.5 Summary

A user-centered design approach is suitable for well-defined and well-understood settings, but does not bring value to the initial design of innovative products. Usability evaluations can be dangerous when applied to novel products because of a lack of cultural adoption.

The ToU protocol explicates my understanding of the problem space based on anticipated usage scenarios and ethical concerns. Once the technology reaches a solid cultural adoption, other designers can use this ToU to make adjustments to the framework, if needed, thereby allowing evolution and maturation. Hardware and software sketches are used to explore the design space and refine the ToU. The result of this thesis incorporates the principles of designing for appropriation, to allow users to tailor the solution to their needs.
4. Results

"Machines only manipulate numbers; people connect them to meaning."

4.1 ToU

The following ToU incorporates the findings by Jia et al. (2012) and Pereira et al. (2013). It has been refined with insights gained from paper and hardware sketches (see 4.5 Proof of concept).

4.1.1 Model

The following model describes the characteristics of the usage situation as well as needs and desires of users in IoT environments.

Users ...

... dislike mindless repetitious tasks.

... do not want their agency (see 2.2.2) to be limited by technological devices.

... want transparency on how their data are being used.

... prefer integrated interfaces on devices they already own (e.g., smartphone) over special devices for configuring.

... use internet services, devices and appliances from different companies.

... want support and recommendations when setting up new things instead of starting from scratch.
4.1.2 Qualities of a good solution

The resulting framework conforms to the following qualities that are separated into three categories based on different points of views.

(1) Users

(1.1) Allows users without programming skills to add, remove and connect devices and internet services without programming skills.

(1.2) Information exchange is transparent, so that both humans as well as machines can understand the data being transmitted.

(1.3) Supports gradual achievement of mastery.

(1.4) The establishment of connections requires explicit user consent.

(1.5) A consent for a specific connection can be revoked at any time.

(2) Framework

(2.1) Additional hard- and software from different vendors can be added.

(2.2) Connections can be established between two devices, a device and an internet service as well as between two internet services.

(2.3) Works with or without internet access.

(2.4) Supports daisy chaining of devices / internet services.

(3) Configuration Interface

(3.1) Supports alternatives to create connections.

(3.2) Setup and modification of the platform works through an integrated interface on devices the user already owns (e.g., smartphone, tablet, desktop computer).

4.3 Hypothesis: The Butler Framework

Based on the ToU, one hypothesis has been developed. The hypothesis is called The Butler Framework. The name derives from a human butler who supports his employer with everyday tasks. The framework allows layman users to control connections between internet enabled devices and internet services. It is constructed from a ToU, sketches and the principles of designing for appropriation to allow adjustments by users and professionals. Semantics are the foundation of information representation, allowing users to understand data exchange and protect themselves from unintended information exposure.
The butler framework consists of three parts: a runtime environment, a hub to run this environment as well as configuration interfaces.

A special hub or internet enabled device (e.g., a router) executes the runtime and provides connectivity to devices and the internet. A permanent connection to the internet is optional allowing connections between devices to work offline.

The hub allows configuration via an API. Configuration user interfaces (e.g., smartphone applications) use this API to allow users to delegate tasks and configure the environment.

### 4.3.1 Resource Types

Resource types are aggregations of primitive data types with meta information to model semantic concepts (e.g., temperature, weight, persons). Every resource type can have properties of other resource types (e.g., a weather forecast has properties for temperature and pressure). The framework provides four built-in resource types: Number, Text, Date and Time.

Resource types are either integrated into the framework or come with a family. They are stored centrally on the hub to allow dependency tracking e.g., a family uses a resource type from another family that needs to be downloaded first.

<table>
<thead>
<tr>
<th><strong>Temperature</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
</tr>
<tr>
<td><strong>Unit</strong></td>
</tr>
<tr>
<td><strong>Properties</strong></td>
</tr>
</tbody>
</table>

**Table 4.1: Temperature resource type**

<table>
<thead>
<tr>
<th><strong>Weather forecast</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
</tr>
</tbody>
</table>

*Figure 4.1: Framework hierarchy*
4.3.2 Abilities & Interests

Abilities and interests are the foundation for connections between butlers. Abilities send independent data points in regular or irregular intervals (e.g., sensor data) whereas interests accept such data points. For such communication to work, they both need to be of the same semantic resource type. An ability can also be an interest and vice versa.

4.3.3 Conversations

Conversations are connections between abilities and interests of the same semantic resource type.

4.3.4 Butlers

Butlers are applications of arbitrary complexity. Every butler provides abilities and interests as well as additional configuration options (e.g., login data). There are four types of butlers that have been derived from the qualities of the ToU (1.1, 1.3, 2.1, 2.4):

- *Product butlers* are virtualized products (e.g., a central heating control).
- *Device butlers* are virtual representations of physical devices (e.g., a light bulb).
- *Service butlers* encapsulate API functionality of internet services (e.g., a weather service).
- *Aggregation butlers* are compositions of connected butlers that expose abilities and interests of the contained butlers.
4.3.5 Families
A family manages a collection of butlers. Families can either discover their butlers (e.g., BLE devices in near proximity) or let users add them (e.g., adding a new facebook account). Additional configuration options (e.g., polling intervals) can be used to adjust their behavior (e.g., when discovering butlers).

Weather forecasts

<table>
<thead>
<tr>
<th>Description</th>
<th>A collection of different weather forecast services.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Butlers</td>
<td>klart.se, weather.com</td>
</tr>
</tbody>
</table>

Table 4.5: Weather forecasts family

4.4 Usage scenarios
Instead of cluttering up our environment with additional devices, we could reuse things we already have (lights, shutters, heating units, smartphones etc.) by combining their capabilities.

4.4.1 Wakeup alarm
The Philips Wakeup Light\(^1\) alarm clock leverages the human body’s sensitivity to light and sound. 30 minutes before wakeup time it gradually increases the brightness, reaching the preset intensity at wakeup time. At this time it also plays either sounds or music with increasing volume.

This wakeup alarm can be modelled with the butler framework to allow reusability of artifacts. Today’s smart phones and tablets come with a built-in alarm clock application. Setting an alarm is similar to having an appointment. Thus, we could create an Appointment resource type. For simplicity reasons, that resource type consists of a description as well as date and time.

An alarm clock family discovers all of them via BLE or WiFi and lists them as butlers. Each butler exposes a Wake up time changed ability of

\(^1\) http://www.philips.de/c-p/HF3470_01/wake-up-light (last access June 15th, 2014)
type *Appointment* that fires whenever the user sets a new alarm on the corresponding device.

Our shutters and lights are represented as device butlers. The shutters expose a *Position* interest of type *Percent*; our lights have an *Intensity* interest of type *Lux*.

A translator butler accepts an appointment and transforms it into a percentage value to connect our lights and shutters with the alarm clock. A very simple translator could start 30mins before the target time to increase the percentage using linear interpolation (1% every 18 seconds). More sophisticated versions could use the input from two light sensors (one outside the house and one inside) to decide if it makes sense to open the shutters (e.g., it might still be dark outside) or to just increase the intensity of the lights. It could also combine shutters and lightning to reach a defined brightness level.

![Diagram of the information flow](image)

**Table 4.6: Wake up alarm information flow**

### 4.4.2 Automatic watering systems

Green Electronics LLC offers an automatic watering system called RainMachine\(^2\) that evaluates recent weather forecasts before watering your garden (subdivided into zones), saving both money and water.

Instead of just relying on the weather forecast and a subdivision into zones we could incorporate the soil humidity. This assumes that we have knowledge about the optimal level of soil moisture for a given type of plant.

A watering butler determines the best time for watering plants, if necessary, based on the current weather forecast and the humidity level of the soil. It has interests for weather forecasts and soil moisture as well as an ability of type *Percentage* to regulate water flow.

A weather butler (e.g., klart.se) with an ability of type *Weather Forecast* and a device butler that provides soil moisture readings connects to the interests.

---

\(^2\) [http://www.rainmachine.com/](http://www.rainmachine.com/) (last access June 15th, 2014)
Each of our sprinklers is represented as a device butler with an interest of type *Percentage* to connect to the watering butler.

<table>
<thead>
<tr>
<th>klart.se</th>
<th>Soil moisture</th>
<th>Watering</th>
<th>Sprinklers</th>
</tr>
</thead>
<tbody>
<tr>
<td>New weather forecast</td>
<td></td>
<td>Weather conditions changed</td>
<td></td>
</tr>
</tbody>
</table>

![Graph showing the information flow](image)

**Table 4.7: Rain Machine information flow**

### 4.5 Proof of concept

The framework has been sketched with an Apple iPad Air and Arduino open source hardware components to evaluate the concept and refine the ToU.

---

3 http://www.arduino.cc (last access June 2nd, 2014)
Human Empowerment in a Semantic Web of Things

(Figure 4.2). The proof of concept consists of a configuration application, a hub and nodes (mockups of connected devices).

![Connections in the prototype](image)

**Figure 4.3: Connections in the prototype**

### 4.5.1 Hub

An Arduino Yun provides the foundation for the runtime of the framework. It has access to the Internet via LAN and WiFi and can communicate with nodes via an XBee series 1 module attached via USB. It discovers nodes in range by sending out ping messages every 5 seconds. These pings are also used to determine if a node is still available or has gone offline.

The software is written in C and C++ and compiled with the GNU Compiler Chain 4.7. It uses the libxbee⁴ C-library to communicate with the XBee module.

The API is REST based and provides data via HTTP using the GNU libmicrohttpd⁵ HTTP server library. Consumers of the API can retrieve information in JSON format on available resource types, families, butlers, abilities and interests.

### 4.5.2 Nodes

The nodes are based on Arduino Uno open source hardware. Each of them is equipped with a Seeedstudio XBee Shield and an XBee series 1 module. Custom Arduino sketches written in C and C++ use the XBee Arduino library⁶ to access data from the XBee module. The nodes can transmit information about their capabilities in JSON format on demand.

⁴ [https://code.google.com/p/libxbee/](https://code.google.com/p/libxbee/) (last access June 2nd, 2014)
⁶ [https://code.google.com/p/xbee-arduino/](https://code.google.com/p/xbee-arduino/) (last access June 2nd, 2014)
4.5.3 iPad application

An iPad application provides a configuration interface for the prototype. It is built for an iPad Air 32GB with 4G running iOS 7.1. It communicates via HTTP with the hub, retrieves and parses JSON data and displays it to users. It shows the available families and butlers of the hub and allows the establishment of conversations between butlers.
The user interface design of the application prototype is for demonstration purpose only and has not been tested with users. The idea is to leverage the conversation metaphor so that conversations between butlers look similar to chats in message applications.

The application consists of two screens: a family-butler screen and a settings screen. The family-butler screen shows all families and butlers available on the hub in a Master-Detail layout with families to the left (Master) and a list of butlers to the right (Detail) (Figure 4.6).

Figure 4.6: Butler overview for a facebook family

4.5.4 Information representation

The hub uses JSON structures to represent and transfer information (Figure 4.7). These structures describe resource types, families, butlers, abilities, and interests. The configuration application uses them to allow users to inspect the internals of resource types in the information flow between butlers. Butlers use them to check the compatibility of abilities with interests and vice versa. The hub supports basic interference and mapping rules for semantic reasoning, even though they have not been used for the prototype.
4.5.5 Usage

The hub needs no further setup if connected via LAN. Users connect it to a power outlet and wait until it is booted. After the initial boot, users can access and configure the hub using the iPad application. Initially, the hub comes with an XBee family that leverages the connected XBee module to discover any XBee nodes in range. The hub loads this family on startup, discovers the nodes, and provides access via the API.

Users can configure the families and butlers using the iPad application. They can select any family from the left list view and then see the corresponding butlers in the list to the right. By selecting one of those butlers, the right side of the screen shows the details to the corresponding butler including an icon, a description, abilities, and interests (Figure 4.8).
The detail screen allows users to connect compatible abilities and interests of other butlers. If the interests tab is selected (Figure 4.8), the app displays a list of interests on the right side of the details screen. Selecting one of these interests reveals a list of butlers and their abilities to the left side of details screen.
screen. Selecting or deselecting them establishes or removes a conversation to this interest. If the abilities tab is selected (Figure 4.9), users see a list of abilities to the left. Selecting an ability reveals a list of butlers with compatible interests to the right. The user can then establish or remove conversations to this ability.

### 4.5.6 Limitations

The proof of concept is the result of several sketches, thus limited to the aforementioned functionality. Users cannot download additional families from other platforms, e.g., a store. The only way to add additional families is by subclassing and recompiling the whole runtime environment. Furthermore, the runtime does not include a family which allows users to add butlers, e.g., a facebook family. Thus, only XBee butlers can have conversations with each other.

Selection of another hub is unsupported because the settings screen was considered out of scope for the sketches. Thus, changing to another hub requires alteration and recompilation of the iPad app or the implementation of a settings screen.
5. Discussion

The scarcity of IoT enabled devices prevents a thorough evaluation of the presented framework and prototype. Therefore, this discussion is based on the ToU protocol and correlations to the principles of designing for appropriation. It considers the presented ethical concerns by Jia et al. and the JRC of the European Commission.

5.1 Theory of Use

For easier reference, the qualities of the ToU are repeated below, followed by a description of how they are fulfilled in the hypothesis.

(1) Users

(1.1) Allows users without programming skills to add, remove and connect devices and internet services without programming skills.

(1.2) Information exchange is transparent so that both humans as well as machines can understand the data being transmitted.

(1.3) Supports gradual achievement of mastery.

(1.4) The establishment of connections requires explicit user consent.

(1.5) A consent for a specific connection can be revoked at any time.

(2) Framework

(2.1) Additional hard- and software from different vendors can be added.

(2.2) Connections can be established between two devices, a device and an internet service as well as between two internet services.

(2.3) Works with or without internet access.

(2.4) Supports daisy chaining of devices / internet services.

(3) Configuration Interface

(3.1) Supports alternatives to create connections.

(3.2) Setup and modification of the platform works through an integrated interface on devices the user already owns (e.g., smartphone, tablet, desktop computer).

The central part of the framework is the hub with its runtime environment. An API allows its configuration, including installation of families and butlers from every platform that implements this API (3.2).
Butlers are either virtual representations of devices, internet services or virtualized products (2.1). Families and butlers can be installed from different sources making programming skills unnecessary (1.1). Product butlers and device butlers work without internet access (2.3).

Conversations between butlers can be established and removed by the user at any time (1.1) (1.4) (1.5) (2.2). Each conversation is a connection between an ability and an interest of the same semantic resource type so that both humans and machines can understand the information exchange (1.2) (2.4). Conversations can be established and removed either from the source butler or from the destination butler (3.1) (2.4).

The non-fixed complexity of butlers allows users to master the framework through composition and decomposition of connected butlers. When a user feels more confident he can decompose aggregated butlers and adjust their connections. Technically advanced users can also implement their own families, butlers, abilities, interests and semantic resource types (1.3).

5.3 Designing for appropriation

The principles of designing for appropriation allow users to tailor the framework to their needs. Users can define their own conversations between butlers thus allowing them to delegate tasks. These connections provide meaning to otherwise independent butlers (allowing interpretation). This independence allows multiple ways to achieve the same result (support not control, plug-ability and configuration).

Information exchange is transparent through semantic resource types that are understood by both humans and machines. This transparency allows users to form a mental model of conversations and predict the effects of their actions (provide visibility).

Tasks can be modeled like pipelines, with semantic information flowing from one butler to the next. These pipelines can become very complex. Aggregated butlers can reduce this complexity by composing several connected butlers into one (learn from appropriation).

The butler framework provides a wealth of information for users to make sense of its entities. Families, butlers and their abilities provide names and descriptions in human readable form. Resource types are semantic concepts (e.g., persons, temperature) allowing users and machines to operate on them. It helps users to understand the relationship between entities and form a mental model (expose intentions).

Third party developers can implement their own families and butlers to integrate their devices, services or virtualized products. It allows the framework to grow gradually, as users can extend their hub by downloading families that others have created (encourage sharing).
5.4 Limitations

5.4.1 Technical limitations

The number of families and butlers running on a hub is unlimited, with butlers of arbitrary complexity. If several butlers have high resource demands, the hub cannot react to other events in a timely manner. This might be solved by moving computation intensive butlers to a cloud server, or distributing the computation among hubs.

Families and butlers are running on the hub, making their configuration dependent on it. If you want another environment to react in the same way as your home, they need to be synchronized. Integrating the butler framework into portable devices (e.g., smart phones or tablets) could solve this problem.

Conversations in the framework form a complex undirected graph, that is prone to cycles (paths, where the first and last element are the same) and self-loops (connections of an element to itself). Cycles and self-loops can result in unpredictable behavior and unsafe acts.

5.4.2 Conceptual limitations

The framework’s value for people’s lives is uncertain, because of a lack of cultural adoption and technical readiness of IoT enabled devices. The principles of designing for appropriation, and a ToU, keep the solution open for adjustments. Yet, the framework needs to be evaluated once we meet the required preconditions.

The framework terminology (families, butlers, interest, abilities, resource types, conversations) uses metaphors to ease the creation of mental models. The use of metaphors could also be a problem. Metaphors come with conceptual limitations related to the real world and depend on cultural context (Cooper et al., 2012, p. 269). Therefore, terms need to be evaluated as well to make sure that users can conceptualize their meaning.

The non-fixed number of families, butlers, and conversations is boon and bane at the same time. On the one hand, users can break down complex tasks into manageable units and connect them. This allows debugging, exploration, and the forming of a proper mental model. On the other hand, the more butlers and connections exist, the more difficult it is to form a mental model.

We describe the world around us with the help of different measurement scales: temperature in terms of degrees Celsius, Fahrenheit, or Kelvin (just to name a few); value in currencies (ISO 4217 lists at least 170 different currencies); distance in inches or centimeters (among others). The butler framework does not provide automatic conversion of such concepts into one another. However, a special family could provide converters for these tasks.
5.5 Related Work

5.5.1 User support in Web-of-Things enabled Smart Spaces

Boussard, Christophe, Le Berre, & Toubiana, V. from Alcatel Lucent Bell Labs in France (2011) are working on an SWoT enabled platform and application model for smart spaces. Their solution is based on three key components:

- a development framework to host and publish objects
- user interfaces and semantic based tools to find and compose real world objects
- an application model with a set of ontologies to consume them.

Connected devices are represented as virtual objects which are semantic digital representations to allow semantic reasoning, filtering and composition. Each virtual object includes

- a representation for users
- RESTful APIs
- semantic description (OWL)

This concept uses a gateway resolver, a gateway repository, and virtual object gateways. Virtual object gateways represent smart spaces (e.g., home, office) and provide access to virtual objects in these spaces. The gateway resolver keeps track of running virtual object gateways and provides access to them.

5.5.2 The Semantic Smart Gateway Framework

A similar approach to the work of Boussard et al. has been developed at the University of Tampere in Finland by Kotis & Katasonov (2012). The work is based on the observation, that the IoT inevitably consists of heterogenous device sets and communication strategies. Kotis et al. argue, that interoperability between devices needs to be addressed at multiple levels. This includes semantic interoperability to preserve the meaning of data when transferred using web protocols.

The result of their work is a semantic gateway, that allows communication between and integration of heterogenous devices. It uses SWoT technologies and four core principles to support extensibility:

1) Possibility for progressive growth of IoT environments
2) Possibility to connect devices from different vendors with each other
3) Possibility to install third party software
4) Possibility to develop generic applications to run on various IoT device sets
5.5.3 IFTTT

If-this-then-that (IFTTT) is an online platform, that allows users to create connections between internet services, e.g., facebook, dropbox, etc. Channels are the basic building blocks of recipes and consist of triggers (this) and actions (that). Additional configuration data, called ingredients, allow adjustments of triggers. These connections (or recipes) can also connect devices with internet services. Yet, the platform prevents third parties to integrate their own channels, triggers, and actions. Thus, users are limited to what the platform offers.

5.5.4 Comparison with the butler framework

The architecture of the butler framework is similar to the framework by Boussard et al. (2011). It includes a hub with a runtime environment and uses semantics to represent information. A hub controls several devices and has access to web resources. Connections between devices are based on compatible resource types. Yet, the butler framework is not limited to devices and allows connections between devices (device butlers), internet services (service butlers) and software applications (product butlers). Furthermore, it also allows the decomposition of complexity (aggregation butlers) as well as daisy chaining.

The butler framework conforms to the principles of extensibility by (Kotis et al., 2012). It allows gradual growth through addition of families and butlers (1). Families and butlers allow the integration of hardware and software from third parties (3). Abilities and interests allow connections between arbitrary butlers as long as their semantic resource types are compatible (2). Depending on the implementation of families and butlers they can run on various device sets as long as they provide a compatible run time, e.g., Java Virtual Machine (4). The framework by Kotis et al. (2012) lacks daisy chaining and the decomposition of complexity.

Channels on IFTTT are similar to butlers. Yet, butlers are not limited to devices and internet services, but can also represent aggregations (aggregation butlers) and applications (product butlers). Recipes are the pendant to conversations, without the ability of daisy chaining. Ingredients can be compared to configuration options on families and butlers. The IFTTT concept of triggers and actions is similar to abilities and interest, but the communication between them is opaque to users.
6. Future Research

Today, IoT technology lacks maturity and adoption. Long term observations could identify potential applications, that users are unaware of today. The quantified self movement, an approach to collect data about one’s life, could be used in conjunction.

The entities of the butler framework are tied to a single hub which can lead to problems if there are many butlers running at the same time. Distributing resource intense computation (e.g., cloud, several hubs) could decrease such constraints, and improve the scalability of the framework.

The liberty of connecting butlers with each other creates a data structure known as undirected graph. These graphs become overwhelming if there are many nodes and connections. Visualizing such a network in a meaningful way supports the forming of a mental model and allows exploration. Such visualization could also provide interactivity to manipulate the graph to a certain extend.

Setting up and modifying conversations requires user interfaces for configuration. Jia et al. (2012) concluded that users want unified interfaces on the devices they already own. There are plenty of possibilities for such user interfaces: from smart phones to tablets, up to desktop or web applications.

1 http://www.quantifiedself.com/ (last access June 15th, 2014)
7. Conclusion

IoT enabled environments threaten human autonomy, agency and social justice. Users want protection and the ability to delegate automatable tasks themselves. Information exchange between their devices and internet services should be transparent. They prefer integrated interfaces with useful defaults, on devices they already own (e.g., smartphone) over special configuration devices.

Designing innovative products requires methods to support adaptation, adoption, evolution and maturation over time. The ToU protocol allows evolution and maturation by explicating the status quo (model) and qualities of a good solution. These statements can be scrutinized by users and altered if needed. The principles of designing for appropriation allow to adapt and adopt a solution in ways unforeseen by designers.

The butler framework allows the establishment of connections between devices and internet services from different vendors, based on semantic concepts. It is built on the idea of a Semantic Web of Things. Information exchange is transparent by using semantic constructs that both humans and machines can understand.

Families, butlers, abilities, interests and semantic resource types are the key elements of the framework. Families are collections of butlers that allow the creation or discovery of butlers. Butlers represent either devices, internet services, virtualized products, or an aggregation of the former. Users can setup conversations with each other based on abilities and interests of the same semantic resource type.

The butler framework supports users’ autonomy and agency by providing means to setup and configure connections and understand the information exchange between entities.

The scarcity of IoT enabled devices and the related lack of cultural adoption leaves the framework as hypothesis. In a best case scenario, it will provide valuable insights for current and future research. In a worst case scenario, professionals need to reject the hypothesis and create a new one based on the ToU.
Human Empowerment in a Semantic Web of Things
Acknowledgments

I could not have finished this piece of work without the support and contribution of people around me. It was a great journey that finally led to this thesis and the following people deserve special acknowledgment for their contribution.

The Olympic torch goes to my parents for their continuous support that allowed me to pursue whatever crazy idea has crossed my mind - which eventually brought me to Sweden. They deserve my deepest appreciation and gratitude.

The philosopher’s stone goes to my supervisor Mikael Laaksoharju for making sense of my scattered thoughts. Without your inspiration, feedback and critique this thesis would have been a lot less fun. Thank you for that!

The golden soldering iron goes to Martin Jacobsson for pushing me in the right direction and sharing his knowledge on wireless sensor networks and smart spaces. I hope that the ideas in this thesis can contribute to your research as well.

And last but not least, the Pulitzer Prize for Criticism goes to my reviewer Lars Oestreicher for his valuable feedback and critique.
References


Carroll, Jennie., Howard, S., Vetere, F., Peck, J. and Murphy, J. (2001) Identity, power and fragmentation in cyberspace: technology appropriation by young people. In Proc. of Australian Conf. on Information Systems


