SEMANTIC CONNECTIONS

explorations, theory and a framework for design

Bram van der Vlist

Semantic Connections

Explorations, Theory and a Framework for Design

Bram van der Vlist

The work described in this thesis was sponsored by the SOFIA project. SOFIA is funded by the European Artemis programme under the subprogramme SP3 Smart environments and scalable digital service.

Typeset with LaTeX 2_ε

LaTeXtemplate provided by Alex Juarez

Cover Design by Bram van der Vlist

Printed by Eindhoven University Press

A catalogue record is available from the Eindhoven University of Technology Library.
ISBN: 978-90-386-3536-1

Proefontwerp Technische Universiteit Eindhoven
Trefw.: Interaction Design, Design Semantics, Ontologies, Semantic Web

Semantic Connections

Explorations, Theory and a Framework for Design

PROEFONTWERP

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
rector magnificus, prof.dr.ir. C.J. van Duijn, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op woensdag 22 januari 2014 om 16:00 uur

door

Bram Jan Jacobus van der Vlist

geboren te Bemmel

De documentatie van het proefontwerp is goedgekeurd door de promotor:

prof.dr.ir. L.M.G. Feijs

Copromotoren:
dr. J. Hu PDEng MEng
en
dr. ir. S.A.G. Wensveen

Acknowledgements

The work presented in this thesis would not have been possible without the help and support of my supervisors, co-workers, colleagues, friends and family.

First of all I would like to express my gratitude to my promoter prof. dr. ir. Loe Feijs, and my co-promoters and daily supervisors dr. Jun Hu and dr. ir. Stephan Wensveen. Along my journey, you have kindly provided me with your guidance, advice and support. I value our discussions on our field of work, but also on all those other things life has to offer.

I would also like to thank the members of my reading committee: prof. dr. Steven Kyffin, prof. dr. David Keyson and dr. Jacques Terken, for their efforts and valuable feedback.

Special thanks to Gerrit Niezen , with whom I worked closely together during this endeavour. Without our cooperation, this thesis would definitely not have been possible. I sincerely value the way we worked together, and how our cooperation turned out.

I would like to thank Stefan Rapp, Aly Syed, Riccardo Trevisan, Sriram Srinivasan, Tanir Ozcelebi, Hans van Amstel, Jettie Hoonhout and Jolijn Teunisse for their contributions to the smart home pilot. My gratitude also goes out to the other project partners, for the pleasant cooperation during the SOFIA project.

My office mates and fellow PhD's at TU/e: Alex, Anna, Siebrecht, Misha, Marija, Jan, Roman, Boris and Martin. I will always remember the great fun we had, and of course the many serious discussions on science accompanied by tasty alcoholic beverages (obviously outside our office). Without you guys, doing my PhD would never have been so much fun. I would like to thank Ellen Konijnenberg for helping me with all the organisational issues, especially when I was no longer around at the department. And of course, a thank you to all other colleagues from the Designed Intelligence group.

I thank my friends from E.S.S.V. Isis: Theo, Olga, Rik, Eva, Nard, Floor, Rico, Rein, Joep, Gerwin(s) and many others, for the great sportive and non-sportive get-togethers, events and holidays. Nothing clears a busy mind better than speed-skating together and cycling up a kilometres long climb. A special thanks to the members of Isis pursuing similar academic goals: Isis PhD's, it's nice to be around people that understand the many peculiarities of PhD life.

My friends Simon, Liselot, Eelko, Kim, Niek, Christy, Wout and of course Café Samson: our evenings are priceless (although expensive).

Finally I would like to thank my parents, brother, sister, Rob, Maaike and the latest additions to our family; Zoey, Ties, Tygo, Boaz and Sam. Thank you for your support and always being there for me.

Summary

Semantic Connections

Explorations, theory and a framework for design

This thesis approaches the issue of interoperability between devices and services in the home from a design perspective. It builds on the fundamental idea of ubiquitous computing; that the majority of our products and devices will be able to interconnect and interoperate. This tenet faces designers with a challenge: to create meaningful interactions for users to deal with the complexity of the ecosystem of interoperating devices they function in.

When moving away from interaction with a single product towards interaction with a system of products, designers need to find ways to communicate the relationships between the products and the larger system they are part of. Additionally, designers are challenged to communicate the possibilities of new, *emergent functionalities*, that emerge when products are being interconnected. This paradigm shift changes the way action and function are coupled and spatially distributes user interaction.

This thesis introduces a *Semantic Connections Theory*, where we view smart environments in terms of connections and associations between the artefacts (*Smart Objects*) and actors within the environment. In this theory semantics is pivotal. A connection is a *semantic connection* when it describes the relationship between two entities in a smart environment and focuses on the semantics—or meaning—of the connection between these entities. The theory was developed through a series of design iterations, exploring and testing various ways of interacting with otherwise invisible connections, enabling users to manage the connections and information exchanged between devices in their homes.

Building on a *semantic interoperability platform* which enables devices to interconnect and interoperate at a semantic level, the design explorations were implemented and tested in users studies. These user studies aimed at eliciting the users' mental models to investigate how users conceptualise the connections and the information they carry.

Based on our theory, a framework is introduced in which meaning, action and function are coupled. The framework is based on existing theories of product semantics and user interaction. It uses and extends these theories beyond their traditional focus on the appearance of objects and interaction with them in isolation, towards designing for systems of inter-operating products.

The framework is evaluated by discussing examples of products and design prototypes that implement the underlying theory, and by asking six designers to use it analytically and generatively. The framework is aimed to help designers and developers of interoperable smart objects to deal with the challenges in contemporary interaction design. The results of the evaluation show that the framework is understandable, applicable and useful, and can be used by designers to analyse and aid the design of interoperable smart objects.

Samenvatting

Dit proefschrift benaderd interoperabiliteit van apparaten en services in de thuisomgeving vanuit een ontwerp perspectief. Het werk is gebaseerd op het fundamentele idee achter *ubiquitous computing*: dat het merendeel van onze producten en apparaten de mogelijkheid krijgen met elkaar te verbinden, informatie uit te wisselen en van elkaars functionaliteiten gebruik te maken. Deze stelling confronteert ontwerpers van zulke apparaten met de uitdaging betekenisvolle interacties the creëren, waardoor gebruikers controle krijgen over de complexiteit van het ecosysteem van genetwerkte apparaten waarin ze dagelijks functioneren.

Nu de gebruikersinteractie zich verplaatst van interacties met een opzichzelfstaand product, naar interacties met een systeem van producten, zullen ontwerpers manieren moeten vinden om de relaties van producten tot het grotere ecosysteem van producten waarvan ze deel uitmaken zichtbaar te maken. Daarnaast staan ontwerpers voor een uitdaging om de ontstane nieuwe functionaliteiten, die ontstaan vanuit het verbinden van apparaten, te communiceren en beschikbaar te maken voor gebruikers. Deze verschuiving van de interactie naar een systeem niveau veranderd de manier waarop actie en functie gekoppeld zijn, en spreid de interactie over meerdere fysieke locaties.

Het proefschrift introduceert een theorie van semantische verbindingen (*Semantic Connections Theory*) waarin slimme omgevingen worden benaderd aan de hand van verbindingen en associaties tussen de slimme objecten (*Smart Objects*) en de actoren in de omgeving. In deze theorie speelt semantiek een cruciale rol. We spreken van een semantische verbinding, als deze een betekenisvolle relatie beschrijft tussen twee entiteiten in een slimme omgeving, en zich richt op de betekenis van de verbindingen tussen deze entiteiten.

De theorie is tot stand gekomen door een serie van ontwerpiteraties waarin er verschillende manieren van interactie met anderszins onzichtbare verbindingen is onderzocht, met als doel gebruikers een beter beeld van, en controle te geven over de verbindingen en informatie die word uitgewisseld tussen de apparaten in het huis.

Bouwend op een semantisch interoperabiliteit platform (*semantic interoperability platform*), dat verbindingen tussen apparaten en apparaatinteroperabiliteit op het niveau van semantiek mogelijk maakt, zijn de ontwerpiteraties geïmplementeerd en getest in gebruikersstudies. De gebruikersstudies hebben als doel de mentale modellen van gebruikers te achterhalen en te onderzoeken hoe gebruikers de verbindingen en de informatie die deze dragen conceptualiseren.

Gebaseerd op de geïntroduceerde theorie, is een raamwerk ontwikkeld waarin betekenis, actie en functie worden gekoppeld. Het raamwerk is gebaseerd op bestaande theorie uit de productsemantiek en kennis van gebruikersinteractie. Het gebruikt en verruimd deze theorieën voorbij de traditionele nadruk op het uiterlijk van en interactie met producten die op zich zelf staan, naar een manier van ontwerpen voor genetwerkte en apparaten die samenwerken.

Het raamwerk is geëvalueerd door het toe te passen op voorbeelden van bestaande producten en nieuwe ontwerpen waaraan onze theorie ten grondslag ligt. Daarnaast zijn er zes ontwerpers gevraagd het raamwerk analytisch en generatief toe te passen. Het raamwerk heeft als doel ontwerpers en ontwikkelaars van slimme en samenwerkende apparaten handvatten te bieden om de uitdagingen van het hedendaagse interactie ontwerpen aan te gaan. De resultaten van de evaluatie laten zien dat het raamwerk begrijpbaar, toepasbaar en bruikbaar is, en door ontwerpers kan worden gebruikt voor analyse en het ontwerpen van samenwerkende slimme apparaten.

Contents

List of Figures

List of Tables

Part I

Research context and state-of-the-art

In this first part of the thesis, we introduce the context of the research. We define goals and research questions and review relevant literature, theory and related work.

Chapter 1

Introduction

The number of smart devices that people use is growing rapidly and will continue to grow in the foreseeable future. Many of these devices can connect to the internet, or to other devices in the same environment over a wireless or wired local network. Therefore, the way users interact with these devices is changing from interaction with a single device, into interaction with a larger system of interconnected devices. Some of the smart devices are becoming portals to information stored somewhere else (e.g. online services). Others have the potential to share information, data and capabilities. From a user's viewpoint there is the need to seamlessly operate among these devices, however this cannot be done successfully yet. In particular, there is a contrast between the current ways of user interaction and the way user interaction was envisioned in paradigms like Ambient Intelligence (Aarts and Marzano, 2003), Pervasive Computing (Satyanarayanan, 2001), Ubiquitous Computing (Weiser, 1991) and the more recent notion of an Internet of Things (van Kranenburg, 2008).

As a real-world and current example reflecting this problem, imagine the context of a modern-day bedroom. People may use various devices in their evening routine before falling asleep (e.g. tablet, e-reader, smart phone, radio, bed light), as well as devices and appliances that help them wake up pleasantly the next morning (e.g. alarm clock, radio/music player, wakeup light). More recently, sensors that check sleeping patterns have become available (e.g. Zeo Sleep Manager[1], Jawbone UP[2], Sleeptracker[3] and FitBit[4]) as well as smart phone applications that offer similar functionality using build in accelerometers. Such sleep monitors can track sleep cycles, and promise to wake a person up at a more appropriate time in their sleep, than a normal alarm would. Reports that summarize sleeping behaviour and efficiency can be reviewed and analysed later. Even though all these devices together could enable a pleasant wakeup experience by sharing information and capabilities, they are not interoperable as such.

Imagine connecting a sleep monitor to a bed light, helping you to pleasantly wakeup at the right time in your sleep cycle, by a simulated sunrise (similar to the functionality of a Wakeup Light e.g. as sold by Philips[5]). When adding even more devices, the wake-

---

[1] http://www.myzeo.com/sleep

[2] http://jawbone.com/up

[3] http://www.sleeptracker.com

[4] http://www.fitbit.com/product/features#sleep

[5] http://www.philips.co.uk/c/wake-up-light/38751/cat/

up experience could be enhanced by waking you up with your favourite music, taking the alarm-time—conveniently set on your smart phone into account, so you will be awake in time. By connecting a smart thermostat, your home could be at a pleasant temperature when you get out of bed. Currently such a (cross domain) scenario, while imaginable, can not easily be achieved. But even a single-domain scenario like sharing media among devices is often not straightforward, especially when the devices involved are produced by different manufacturers, that do not support interoperability outside of their ecosystem. The many difficulties in connecting devices in a home environment are well described in (Merabti et al., 2008). A more elaborate discussion of the state-of-the-art is available in Chapter 2. We will revisit this scenario in Chapter 5.

The key goal of ubiquitous computing[6] is "serendipitous interoperability" , where devices which were not necessarily designed to work together (e.g. built for different purposes by different manufacturers at different times) should be able to discover each others' functionality and be able to make use of it (W3C, 2004). Future ubiquitous computing scenarios involve hundreds of devices, appearing and disappearing as their owners carry them from one room or building to another. Therefore, anticipating all the different types of devices and usage scenarios upfront is an unmanageable task. To overcome these technological challenges, a European research project called SOFIA was commenced, which will be introduced in Section 1.1.

Besides the technological challenges, there also lies a challenge ahead for designers who are designing user interactions with such ecosystems of interconnected devices. When moving away from interactions with a single device towards interactions with systems which include multiple devices, designers need to find ways to communicate the relationships between the devices and the larger system they are part of. Additionally, designers need to find ways to communicate the action possibilities of new, or "emergent" functionalities, that emerge when devices are being interconnected.

An important problem that arises when designing for these systems of smart devices, is their highly interactive and dynamic nature (Frens and Overbeeke, 2009). The inherent ever-changing nature of these systems and the limited overview designers have of the ecosystem in its entirety, are the most important challenges a designer faces when designing for such systems. Moreover, such a system comprises many different "nodes" that the designer, at the time of designing, has no control over. Yet, when designing new devices that are to be added to the system, making them interoperable is crucial for success. When designing such interoperable devices, it is key to enable a coherent user experience among the full spectrum of devices, that are each designed differently, by different people.

In the past, (interaction) design has been focussing mainly on products (both hardware and software) and the interaction with them as solitary entities. Now that devices are increasingly being interconnected, user interaction often involves multiple devices, when still trying to reach one single goal. The devices themselves do not fundamentally change, nor does the interaction (i.e. there is no clear observable difference in appearance and interaction style between connected and non-connected products. See also Section 2.2). Since fundamentally changing the interaction users have with devices is difficult and perhaps even undesirable, our aim is to focus on that which is actually changing; the introduction of connectedness. As the connections—and with the connections, the information available

---

[6] in this thesis we adopt the paradigm of ubiquitous computing, as this matches our understanding of a smart environment the closest

on the devices and their capabilities/functionalities—are at the centre of this change, they will be the main focus of this thesis.

In this thesis we introduce a new approach to designing interoperable systems, focusing on the inter-device relations and connections that exist or may potentially exist. Such relations may be both real "physical" connections (e.g. wired or wireless connections that exist in the real world) and "mental" conceptual connections that seem to be there from a user's perspective. The context of the connections and the things that they connect, are pivotal for what they will come to mean to their users. For users to effectively function in future scenarios as were sketched earlier, it is necessary that users are able to make sense of the connected devices in their environment. Therefore, users should also be able to make sense of the connections that connect these devices. To support the process of sense making, *semantic connections* are introduced. A connection is a *semantic connection* when it describes a meaningful relationship between two entities in a smart environment and focuses on the semantics—or meaning—of the connection between these entities.

In the following, this chapter introduces the research context, the SOFIA project, the problems and challenges addressed, our approach in addressing these problems and outlines our contributions. We conclude this chapter by giving an outline for the remainder of this thesis.

1.1 The SOFIA project

The problem space and claims of this thesis cannot be explained without explaining its context, and the boundaries this context puts forward. Part of the problem addressed in the thesis is specified by the SOFIA (Smart Objects For Intelligent Applications) project consortium, including about 20 European partners from industry and academia. Among the larger industrial partners are Nokia[7], Philips[8], NXP[9], Fiat Research Centre (Centro Ricerche Fiat)[10], Elsag Datamat[11], Eurotech[12], ESI-Tecnalia[13], Indra[14] and VTT[15]. For an overview of the project and the contributing partners please refer to[16]. The SOFIA project aims to address the challenges of interoperability on a technical level, and served as the context of the research described in this thesis.

SOFIA is an European research project within the ARTEMIS framework that attempts to make information in the physical world available for smart services—connecting the physical world with the information world. The goal is to enable cross-industry interoperability and to create new user interaction and interface concepts, to enable users to benefit from smart environments. The heart of the SOFIA project is the development of an *interoperability platform*. Interoperability is the ability of a system or a product to work with other systems or products. At the start of the SOFIA project, the following objectives were defined in (ARTEMIS JU, 2008):

---

[7] http:www.nokia.com

[8] http:www.philips.com

[9] http:www.nxp.com

[10] http:www.crf.it

[11] http:www.elsagdatamat.eu

[12] http:www.eurotech.com

[13] http:www.esi.es

[14] http:www.indracompany.com

[15] http:www.vtt.fi

[16] http://www.sofia-project.eu/

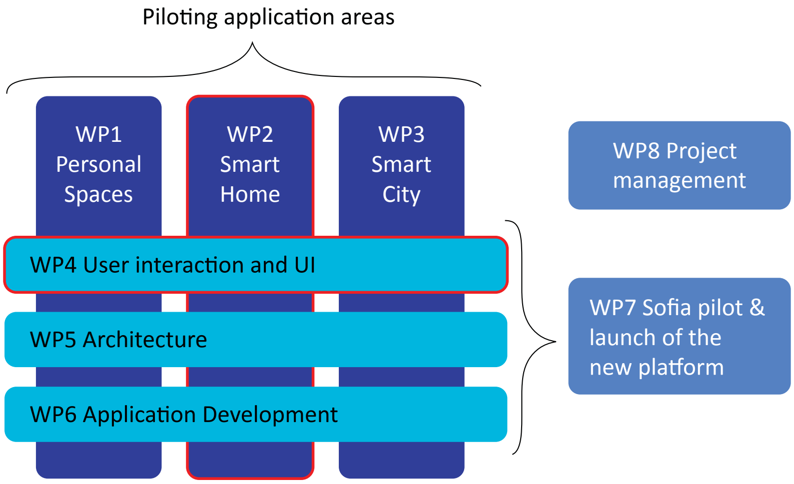


Figure 1.1: Schematic overview of the project structure

Objective 1: development of the SOFIA platform providing the interoperability levels that enable interaction and data exchange between multi-vendor devices. This platform should also take into account relevant devices already existing in a target environment. The platform will support a range of devices from limited resources to resource-rich devices.

Objective 2: interaction models and embedded devices to support a variety of smart spaces and a variety of users. This will move today's device oriented interaction models to a user goal and result oriented interaction paradigm.

Objective 3: methods, techno-economic structures and toolkits for the deployment of smart environments and for the development of services and applications based on smart environments.

Objective 4: scenarios and use cases to demonstrate the capabilities of the SOFIA platform and the proposed interaction models and techno-economic structures in personal spaces, indoor spaces and cities. A pilot showing the interoperability between these domains shall be set up and evaluated.

The SOFIA project is structured according to eight work packages: three vertical work packages representing the three different applications areas; three horizontal work packages representing the key technical solutions; a work package targeting a large scale pilot for demonstrating all aspects of the vertical and horizontal work packages, and a project management work package. Figure 1.1 gives an overview of the project structure. We were involved in work package two (Smart Home) and work package four (User Interaction and UI).

Throughout the project there was cooperation between the project partners within the work packages (i.e. working together on the various defined tasks and deliverables), between horizontal and vertical work packages (e.g. between work packages two, four and five) and across vertical work packages.

Table 1.1: SOFIA deliverables where we contributed. Deliverables printed in boldface were not inside our work packages but we contributed nevertheless.

del. no.	deliverable name
D2.11	Use cases and related resources/services: requirements and specifications
D2.12	Preliminary use case demonstration
D2.13	First release demo's and progress assessment
D2.14	Use cases demonstration, results evaluation and impact assessment
D2.15	Report on pilot support
D2.31	Adapters for interoperable connectivity and service discovery in smart indoor spaces
D2.32	Adapters for interoperable connectivity and service discovery in smart indoor spaces
D2.51	Smart technologies for OIP
D2.52	Smart technologies for OIP
D4.11	Description and Assessment of Interaction Tool Set
D4.21	Analysis of Scenarios, Extracting of Technical Requirements
D4.31	Preliminary Design for Data Storage, Enrichment and Retrieval
D4.32	Design and implementation for Data Storage, Enrichment and Retrieval
D4.43	**Interaction support in the SOFIA IOP**
D4.51	Specification and Implementation of Semantic Transformers
D7.43	**Dissemination and exploitation report**

Besides the dissemination of SOFIA related results to the design and computer science community through the various publications (as are listed in Appendix G), results were also disseminated to the general public in a number of project deliverables. Contributions were made to the deliverables as are listed in Table 1.1. These deliverables were reviewed by reviewers assigned to the project and are publicly available[17] at[18].

The total eligible cost of the project was 36.54 M€, of which 6.1 M€ was contributed by ARTEMIS and 8.92 M€ was contributed by national funding bodies. These numbers indicate the commitments made by the industrial partners.

1.1.1 Implications for our research

The objectives as were stated by the SOFIA project have several implications for the research described in this thesis. Among other things, the SOFIA objectives implied:

- The SOFIA interoperability platform targets both newly designed smart objects as well as legacy devices. For our research, this means that we focus on both designing new smart objects as well as incorporating existing ones. One of the resulting challenges is to communicate connectivity and interaction possibilities, without physically altering the legacy devices. In the remainder of this thesis we consider both groups of targeted devices to be *smart objects*.
- In order to test the interoperability platform, usage scenarios and use cases will be defined, implemented and tested. To be able to test the platform in context, with

---

[17] Some of the deliverables are successive deliverables of which only the final deliverables were made publicly available.

[18] `http://www.sofia-project.eu/node/329` and `http://www.sofia-community.org/`

users, a large part of our contribution is on building prototypes and realising the infrastructure needed for testing in such environments.

- The focus of the SOFIA project was mainly on developing the needed *software* to achieve interoperability. To be able to introduce the innovations in the software domain to designers (and also developers), interaction models and methods were developed. This thesis describes such a method (introduced in Chapters 5 and 6).

1.1.2 The SOFIA interoperability platform

Interoperability is the ability of a system or a product to work with other systems or products. In the SOFIA project, interoperability is achieved by using a semantic interoperability platform (IOP), where information is exchanged between devices on a semantic level. The IOP makes extensively use of *Semantic Web technologies*. We participated in the development of the SOFIA IOP and our research prototypes were built by using and extending the IOP.

The SOFIA IOP consists of a shared, semantic-oriented store of information and device capabilities, called a Semantic Information Broker (SIB), and various Knowledge Processors (KPs) that interact with one another through the SIB. The platform addresses information-level compatibility and the collaboration between different producers and consumers of information on an abstract level. The goal is that devices will be able to interact on a semantic level, utilizing (potentially different) existing underlying services or service architectures. Part of this effort is to define a core ontology that describes commonly used concepts, and also model the related domains more thoroughly in a formal ontology that is expressed in RDF (Resource Description Framework, as used in Semantic Web technologies). This is described in more detail in Sections 2.3 and 2.3.4. We extended the SOFIA core ontology by defining our own, aimed at enabling user interaction in an intuitive way and safeguarding a coherent overall user experience.

Ontologies lend themselves well to describing the characteristics of devices, the means to access such devices, and other technical constraints and requirements that affect incorporating a device into a smart environment (Heflin, 2004). In addition to device capabilities and technical characteristics which have been described by ontologies in earlier attempts (for an overview of such attempts see Section 2.3.5), our efforts also focus on describing and modelling user interaction capabilities and interface elements that are important from a users' point of view. Using an ontology also simplifies the process of integrating different interaction approaches, as the different entities and relationships in the SIB can be referred to unambiguously. Because communication via the SIB is standardized, integrating cross-vendor implementations is also simplified, and technical incompatibilities can be captured by the ontology and can be communicated to the user. We believe that Semantic Web technologies and the SOFIA IOP may be used to support user interaction in smart home environments, as is described in more detail in (Niezen et al., 2010) and (Niezen, 2012).

1.2 Design Context: changing needs of the design community

To illustrate the difference between the work presented in this thesis and state-of-the-art design methodologies in design practice, we introduce a related project: the Lifestyle

Figure 1.2: An overview of the three different experience prototypes developed during the Lifestyle Home project. [image source: Philips - `http://www.design.philips.com/shared/assets/Downloadablefile/LifestyleHome_brochure-14047.pdf`.

Home project[19]. The Lifestyle Home project offers the opportunity to compare the design approaches followed by the designers working on the Lifestyle Home project, and the design approach to be developed in this thesis.

It must be stated that the two projects (i.e. Lifestyle Home and SOFIA) are considerably different. The SOFIA project and the work presented in this thesis have a strong software engineering component, whereas the Lifestyle Home project has a strong user centred approach (e.g. using personas). The Lifestyle Home project delivered experience prototypes that propose new ways of interacting with media in the home, in contrast to providing a viable interoperability platform which also enables a coherent user experience among its different components, as is the focus of our work.

The Lifestyle Home project presents a futuristic vision for "connected living" that "embraces a diversity of tastes, habits and needs" (Koninklijke Philips Electronics N.V., 2006). The project resulted in three experience prototypes that each target a different type of person (Figure 1.2). These prototypes were developed by employing (at that time) state-of-the-art user centred design approaches, centred around three different personas (Pruitt and Grudin, 2003) (Alexandra, Simone and Justin), that each targeted a different specific user group. The designs aim to "illustrate how a family of solutions can be adapted to support many different lifestyles", where "people are free to select, tailor and enjoy their digital media and connected experiences to best suit their unique situation"(Koninklijke Philips Electronics N.V., 2006).

The central element in Lifestyle Home's proposition, is a TV based menu that allows access to digital content stored both locally and online. The menu can be personalised by adding, removing, renaming and rearranging a set of standard menu categories. The menu also plays a central role in the management of "peripheral lifestyle devices" such as ambient lamps, photo frames and media tablets. Content can be shared with these devices by allowing users to load content on the peripheral devices, which will show up visually represented in the TV's home menu. There are also several ways to manage content among the peripheral devices in a more tangible way. For example, for the persona of Alexandra,

---

[19] `http://www.design.philips.com/lifestylehome/`

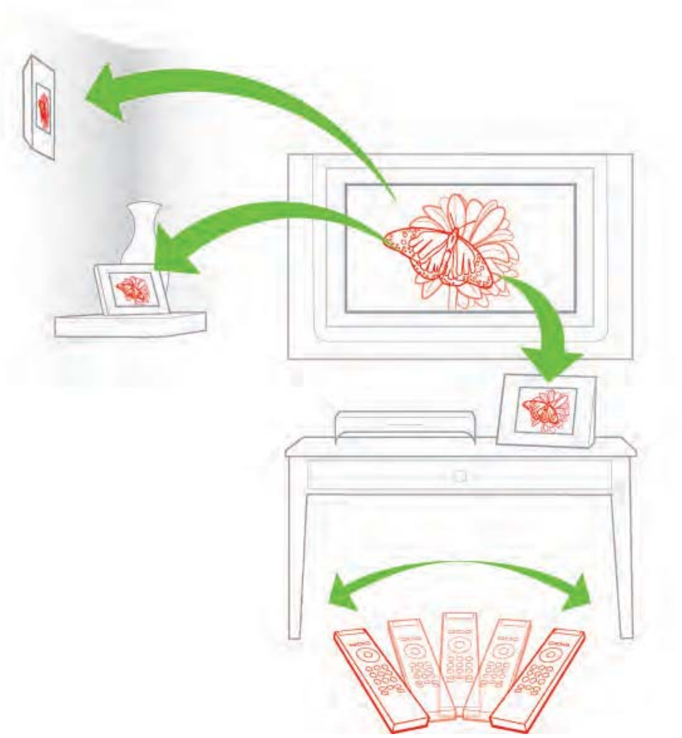


Figure 1.3: Example of a point-and-drag" style user interface. [image source: Philips - `http://design.philips.com/shared/assets/Downloadablefile/LifestyleHome_brochure-14047.pdf`.

a "point-and-drag" style user interface is proposed, to move content like photo's or movies to a media frame using a special remote (Figure 1.3).

Other user interface styles proposed by Lifestyle Home include:

- NFC (Near Field Communication) using NFC to establish a connection between a phone and the TV to exchange photo's (more on NFC can be found in sections: 2.2.2 and 4.3.1);
- a mobile tablet-style touch interface, serving as a remote control for sharing media in the environment and accessing digital content like movies, video's and internet services;
- a track-pad remote to browse the TV based menu;
- contact less payment through NFC.

1.2.1 Identifying design approaches

When comparing the outcomes of the Lifestyle Home (LH) project and the SOFIA project, we can identify similarities in the UI solutions that were developed (e.g. using point-and-drag, NFC (touch/proximity) based interaction.). Additionally, the intentions to make the user experience flexible and customisable, adapting to match a user's personal needs, is an aim that is shared. The main differences (beside the different aims of the projects) are, however, in the design approaches used in the different projects.

LH employed a very user centred approach, whereas the SOFIA project had a strong technological focus. More importantly, LH followed a typical top-down approach, which is suitable for developing a vision of the future, and where the absence of a software architecture and its complexities, allows for proper comparisons and a methodologically complete evaluation of concepts. In contrast, the SOFIA project aims to develop a technological framework that allows for multi-vendor devices to work together and share data and capabilities. For designers of such products this means that they are only designing a part of a larger, open ended system. Therefore, having a holistic view on the targeted user experience is difficult, if not impossible.

The type of design challenges which are inherent to the problem space identified by SOFIA, reveal an important gap in design methodologies currently available. Such design challenges ask for a bottom-up approach instead of a traditional top-down approach. A more bottom-up approach would enable designers to "prepare" their designs for integration in a yet unknown situation at a later stage. For this purpose, we developed an alternative design approach. We expect that our design approach, including the technological framework developed along with it, would suit the needs of design researchers and practitioners that are designing interoperable systems better than conventional design methods would.

1.3 Focus and Methodology

Based on the problem statement and the discussion of the industrial context of this research, we will more clearly identify the focus of this thesis and introduce the approach we intend to pursue.

As was sketched earlier in this introduction, the focus of this thesis is on the design of interoperable smart objects. When the design goal is no longer designing an intuitive, pleasant and coherent user experience for a single device, but that of a connected device which is part of a larger system, the design task increases in complexity. The aim of this thesis is to explore the difficulties involved in the design of such devices (which we from now on call smart objects). By shifting the focus from the smart objects themselves to the connections that exist or may potentially exist between the smart objects, we aim to better understand the meaning of these relationships. The lessons that we learned from these explorations are presented in this thesis, and will hopefully help designers to deal with the challenges in contemporary interaction design.

We used an *iterative design process*, starting with simple use cases to gradually build more complex ones. Because of the complexity of the design challenge at hand, gradually increasing complexity and implementing research prototypes to understand the problem better is necessary. In every iteration, the lessons that we learned previously were used to gradually define our notion of semantic connections and build our Semantic Connections Theory, also to be developed in this thesis.

This thesis employs a *research through design* approach. This approach is a form of action research, where design is employed to generate (scientific) knowledge (Archer, 1995). In this thesis, the designs are a vehicle to explore the notion of semantic connections, and investigate how users conceptualise these semantic connections by looking at the mental models they develop. The thesis describes two design explorations. The first includes a series of designs to explore the concept of semantic connections, and investigates the influence of physical designs to interact with semantic connections on the mental models users develop. In the second design exploration, our redefined notion of semantic connections is implemented in a more complex use-case, and two new designs are evaluated. Based on these explorations, a theory is defined which is evaluated by implementing it in a third use-case scenario. In this final iteration special attention is given to the design of the interactions in terms of *feedback* and *feedforward*, and the results are used to formulate a framework for interaction design.

1.3.1 A multidisciplinary approach to design

Designing interoperable products is a complex matter. As described earlier in the introduction, designers of interoperable products are designing only a part of a larger system, without having an overview of the system in its entirety. Moreover, products should not only be compatible with the current state of the system, but also remain valuable when new products are added. To ensure interoperability, designers of such products should use common concepts and principles.

Designing in such a context also asks for a multidisciplinary approach, as the resulting products are a combination of hardware, software and services. Therefore, designers and developers should have a common vocabulary and framework that helps them to cooperate to create successful products. This thesis attempts to investigate the knowledge that is required and aims at establishing a framework that can be used by both interaction designers and developers.

To reach this goal, close cooperation with other disciplines was necessary. Close cooperation with Gerrit Niezen (Niezen, 2012) was essential to ensure that the framework describes concepts relevant for both designers and developers. To test ideas and learn how other designers think about, and deal with designing for systems of products, there was involvement of other designers and design students. Working together with design students helped understanding the needs of current and future generations of designers when dealing with the design challenges that are addressed in this thesis.

As mentioned in the previous section, the PhD project was also part of the SOFIA project, which aim was to investigate the use of semantic web technologies in solving interoperability issues. Our focus within the SOFIA project was on the user interaction aspects of devices in the smart home environment. In terms of user interaction, design (as a discipline) had a leading role. Figure 1.5 summarises the design and research process that was followed and indicates how the various parties (and disciplines) worked together. Figure 1.4 shows the role Niezen and I had in the SOFIA project, bridging between the computer science, SOFIA and design communities.

1.3.2 Reflective Transformative Design process

The Reflective Transformative Design process (RTD process) was originally created to address the changing field of design and design education (Hummels and Frens, 2008). Among other things, it supports designing highly innovative products and intelligent, open systems. The RTD process contrasts established linear design processes like rational problem solving, and is more open and better suited to make design decisions based on (too) little information.

The design challenges addressed in this thesis ask for such an open, explorative approach, as the smart objects and the system they will become part of have an open nature. In contrast to a traditional user centred design approach, where users are interviewed (or are involved through other well established means) to identify user needs—which are then used

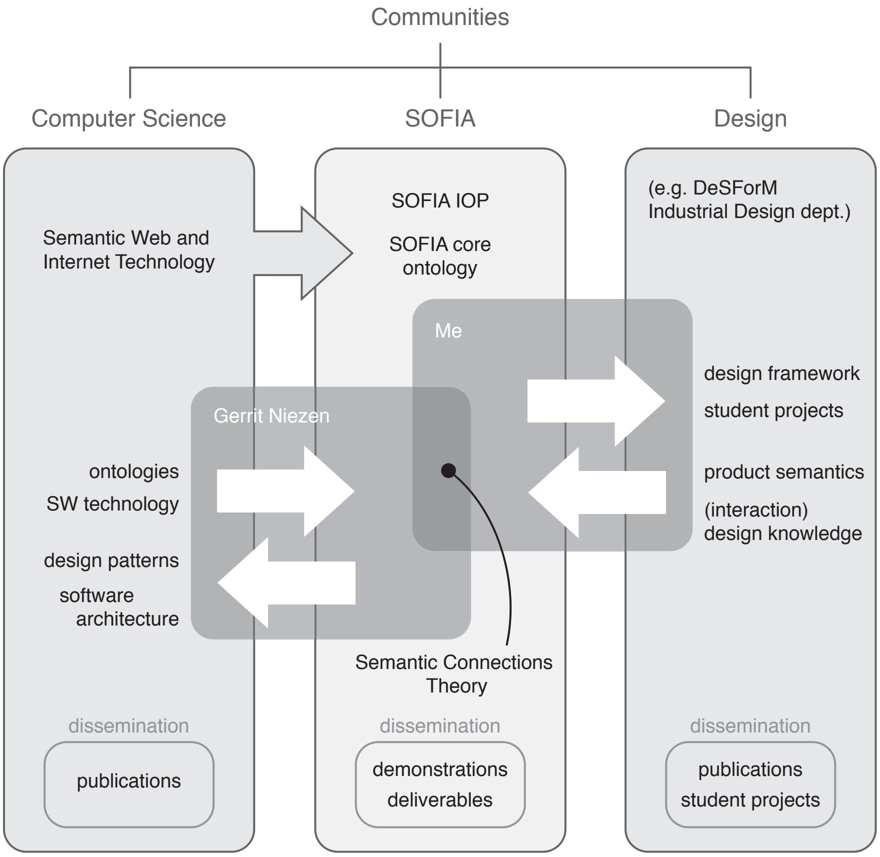


Figure 1.4: Schematic overview of knowledge exchange between communities

in the design process to design a product or system "top-down", we need a different approach. Instead, we are better helped by a process that allows for a "bottom-up" approach. In such an approach, users are confronted with mock-ups and prototypes throughout the design process. Instead of prototyping true design solutions, these prototypes are aimed at understanding how people think about, and use the concepts they prototype. Moreover, we employ qualitative research methods to get insights in users' understanding of these prototypes and mock-ups, to learn how they use them in context.

Explaining the RTD process in detail would go beyond the aim of this thesis, a detailed description of the process can be found in (Hummels and Frens, 2011). Figure 1.6 gives a graphical overview of the process. The RTD process consists of two axes: vertically *drives* are distinguished, and horizontally *strategies* for information gathering are distinguished. In the following, we explain how the RTD process was used in our research.

Envisioning is the first drive. Design decisions were directed through a vision, which was developed early in the project. Our vision of *semantic connections* (Section 3.2) was developed based on the technological advancements developed in the SOFIA project, and use cases that were defined at the start of the project. Our vision was aimed to transform today's device oriented interaction paradigm to a user goal and result oriented one. During the course of the project, this vision was adapted and sharpened[20], and eventually resulted in the interaction model, theory and framework (as are introduced in Chapters 5 and 6). A

---

[20] A redefined definition of semantic connections can be found in Section 4.2

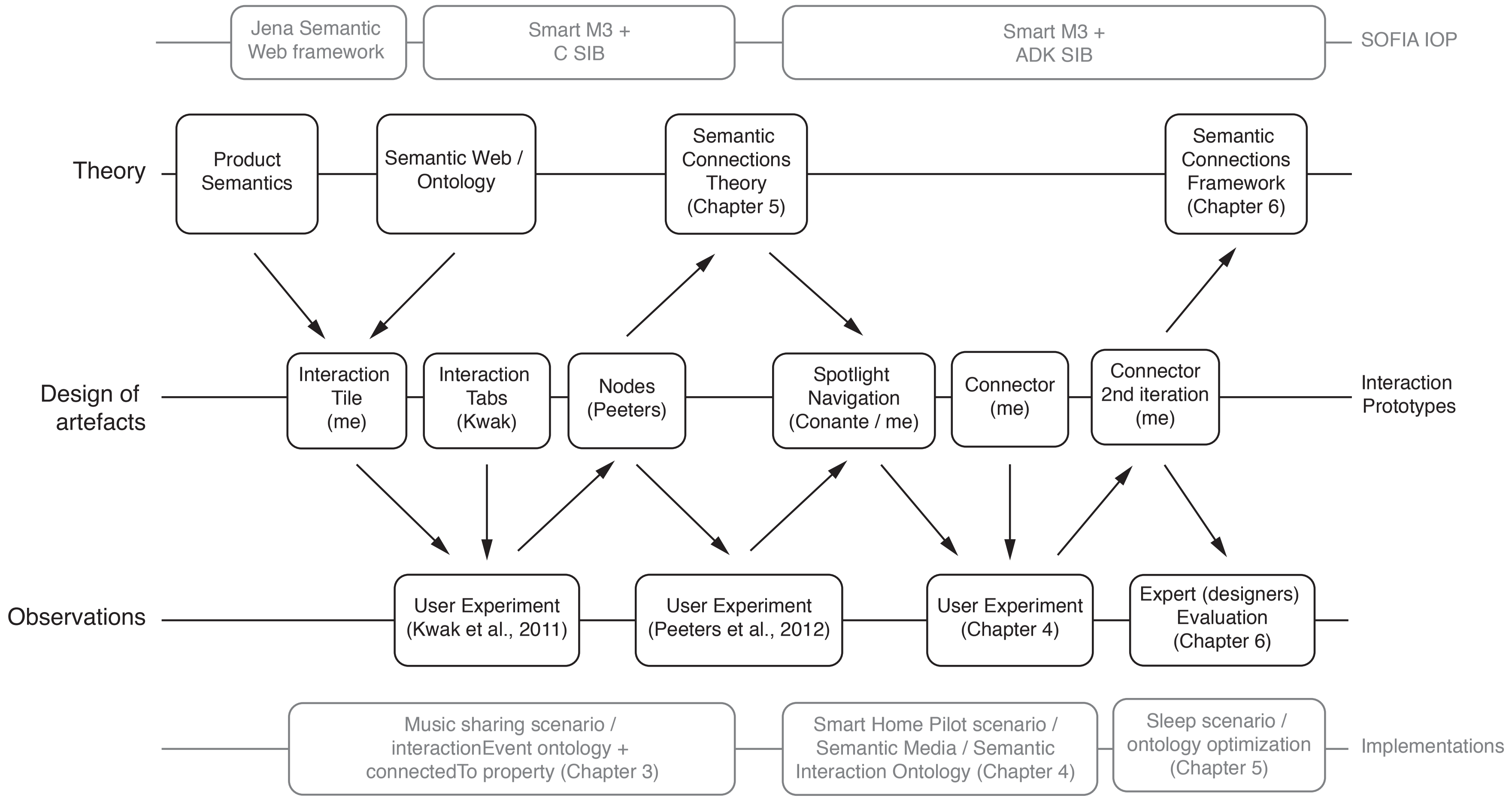


Figure 1.5: Design and research process that was followed, aligned to the triangulation framework of Mackay and Fayard (1997). Besides the interactions that are visualised in the diagram, there we also interactions between theory, observations, SOFIA IOP (top) and implementations (bottom), as they were part of the design iterations.

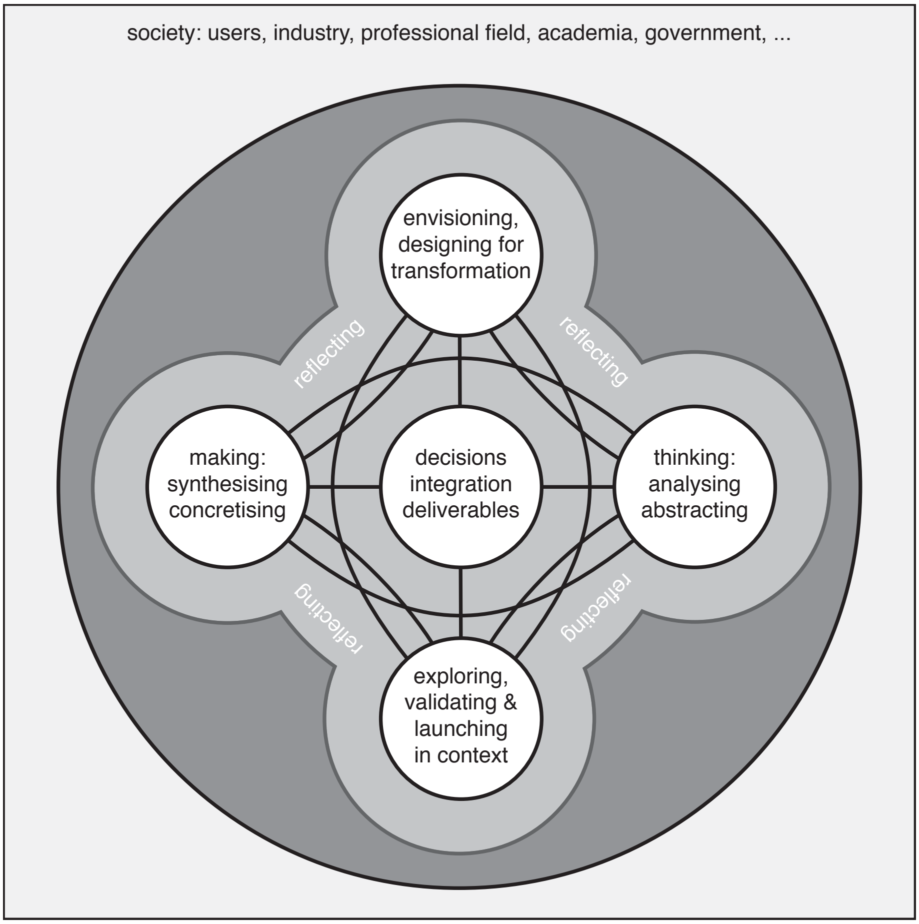


Figure 1.6: The Reflective Transformative Design process (Hummels and Frens, 2011).

second drive is aimed at *exploring and validating* the design decisions in a real life context. We employ these types of validations both on the level of (interaction) design decisions, as well as validating our design decisions made on the level of ontology design and the software architecture. Such validations can be found in Sections 3.4, 3.5, 4.7 and 5.4[21].

The two strategies (horizontal axis) include the the activities of *analysing, synthesising, abstracting and concretising.* The first strategy revolves around design action, and includes activities such as designing and building prototypes. *Synthesising* refers to the merging of elements into a specific composition for a specific purpose. Examples thereof are the implementations of our experimental setups. Examples of *concretising* can be found in the interaction designs, were interaction models and principles are applied to make concrete designs. This strategy produces experiential information to fuel the other activities in the design process. *Analysis and abstraction* form the second strategy. Important analysis steps are in Sections 3.6, 4.9 and 6.3. Important abstraction steps are the development of our semantic connections theory and framework (Chapters 5 and 6). Abstraction steps were also made in between the various iterations, feeding the development of the interaction model, theory and framework. By applying the acquired knowledge in new use cases, we tested

---

[21] The evaluation in Section 5.4 was aimed at evaluating the semantic connections theory, implemented in a realistic use case. The focus was more on validating whether the theory was complete rather then evaluating the use case with users.

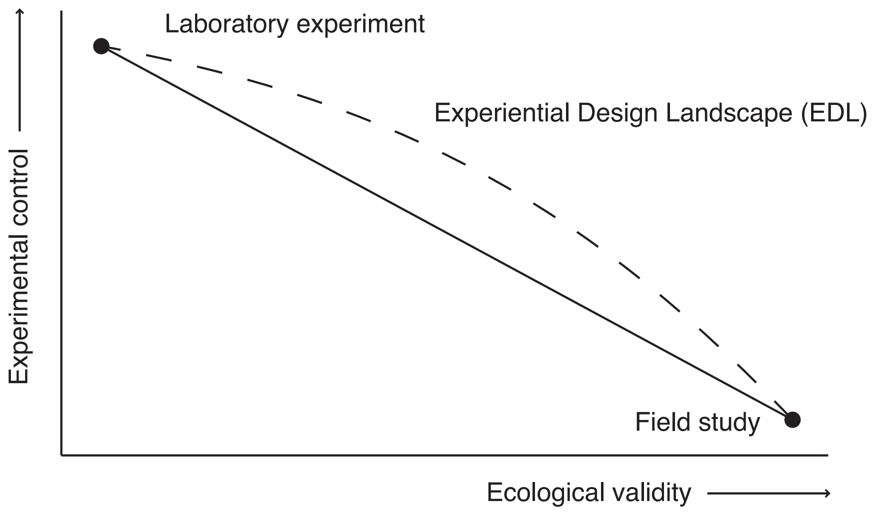


Figure 1.7: Experimental control vs. ecological validity: Experiential Design Landscapes (EDL).

whether this knowledge could indeed be generalised. When moving between the drives, strategies and developing the semantic connections theory and framework, reflecting on the work done plays an important role. Reflections can be found in the various discussion sections and more explicitly at the beginning and end of each chapter.

1.3.3 Experiential Design Landscapes

There is a difference between evaluating designs in a controlled lab setting and evaluating designs in the field. A compromise between high experimental control and a lower ecological validity in a lab setting, versus the higher ecological validity of a field test but less control, is inevitable. Figure 1.7 depicts this compromise. In the light of the RTD process as was described earlier, there is a need for (long term) testing in the field, and yet maintaining experimental control. Settings like the Experience Lab [22] and Living Labs [23] are examples of efforts to perform testing in a lab that approaches real life. Another attempt to provide design tools for designing intelligent systems, focussing highly on the context of use, are Experiential Design Landscapes (EDL) (van Gent et al., 2011).

For our design activities, it is imperative that they are evaluated in contexts that are close to reality. Computer scientists are often satisfied to see their prototype systems work, running code to simulate external factors (such as users) or having their creations function, spread out on a few desks in their office. Designers need, however, to see their work in the context of use, as only experiencing the designs in context will lead to appropriate design decisions. During the design process, we made extensively use of contextual settings like the Context Lab[24] and the Experience Lab. We also employed ideation sessions in context, like the *bodystorming* method (Oulasvirta et al., 2003).

---

[22] `http://www.research.philips.com/focused/experiencelab.html`

[23] `http://livinglabs.mit.edu/`

[24] The Context Lab is a lab at the Industrial Design department, furnished like a real home. It consists of a living room, master bedroom and a children's bedroom.

Additionally, the SOFIA project itself and the community of project partners are also considered a context. This group of mainly computer scientists, Semantic Web experts and software developers, functioned as a test-bed to implement our ideas on user interaction, modelled in ontologies. Vice versa, project partners (including us) in the vertical work packages (Figure 1.1) implemented and tested the advancements made on the interoperability platform.

The size of a project such as SOFIA and the complexity of the software architecture developed by it, prevents the project itself from being an experiment which could be repeatable in any sense of a classical experiment. In software engineering, the process of developing and adopting structuring concepts such as object-orientation, client-server architectures, Semantic Web technologies, etc. is continuous, and cannot be stopped for repeating an earlier act. In view of the size and complexity of the artefacts under construction, this is understandable, if not inevitable. In Software Engineering there is no tradition of controlled experiments; instead case studies, best practices descriptions, and integration projects confirm and consolidate the progression of the field. Just as in the RTD process, it means that reflection becomes more important and that societal impact grows. In this context there was a need for input from the design community into the SOFIA project, as was already described in Figure 1.4, which is the context for my contributions.

1.3.4 Research questions

Before discussing the research questions and goals of this thesis, we first introduce definitions of the used terminology. This terminology is partly based on concepts that are defined in the SOFIA project.

This thesis introduces *smart objects*. Smart objects are objects that have computational power to run software, and wireless communication technology to connect and communicate with a *smart space*. We describe a *smart space* as an ecosystem of interconnected and interoperating smart objects. Smart space refers to the digital domain, including the information-store which contains the information exchanged between the smart objects. The *physical* environments where the smart space and objects are, is what we call a *smart environment* (Figure 1.8). The smart objects category includes devices which are currently available, as well as smart objects which are to be developed.

We distinguish our view on a smart environment from the traditional view of "ambient intelligent" environments. In contrast to these environments, where technology and devices are envisioned to disappear into the surroundings, we believe that many physical products will remain, with their embedded and localized intelligence, as well as their ability of communicating to others. In our view these physical products and especially the relationships among them, can be used by users to express their intentions and to understand the situations in a natural way, both explicitly and implicitly.

Whenever people see, or interact with things in the world, they create a mental model of what they think will happen. Having a usable mental model is therefore key for users to interact with products or systems. As long as the underlying mechanisms of the working of a product are simple and visible, they have a bigger chance to be understood, and make sense (Norman, 1998). For understanding the smart objects, the smart environment these objects reside in, and in particular the relationships between the smart objects, it is important that users develop useful mental models. Throughout this thesis, mental models

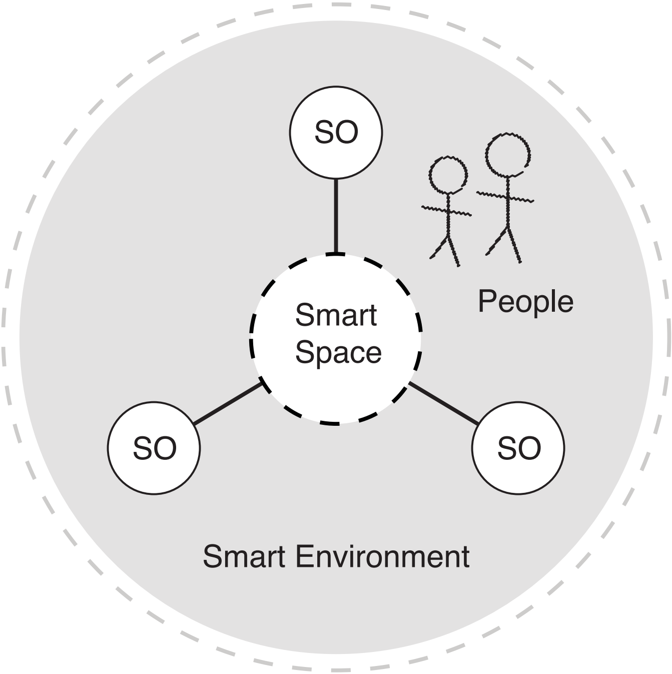


Figure 1.8: Schematic overview of a smart environment, containing smart objects (SO), people and a smart space

will be investigated and discussed to understand how users conceptualise the smart objects and the relationships between them.
This thesis aims to answer the research question:

How can the design of smart objects help users to construct meaningful mental models of:

1. *the smart objects and the smart environment these objects are part of*

   and:

2. *the connections (and their meanings) that exist between the smart objects?*

The mental models of the users about the smart objects and their connections provide the understanding and preferably the clues on how to interact with them. Moreover these mental models need to be constructed by the users themselves. The research question addresses whether the construction process can be supported by proper design of the objects and the interaction, providing the users with a portal to the information reality and the action possibilities. To be able to explore the research questions and find answers to them, a set of hypotheses were defined (these hypotheses are reflected upon in Section 7.3):

1. Smart objects can be designed in such a way that they invite for connecting actions, connecting these smart objects to one another, or to connect them to other smart objects.

2. The qualities of the connecting actions, such as dynamics, direction, order and location, can be used by users to express their intentions.

3. Enabling users to make connections between objects in the physical world can be used to infer a users' intention for information exchange between these connected devices.
4. Through actively exploring and managing the connections between smart objects, users will be able to create more useful mental models and therefore gain a better understanding of the emerging functionalities and information flows within the smart space.

1.4 In this thesis

The research described in this thesis was carried out in the context of the SOFIA project (Section 1.1), the aim of which is to build and experiment with an Interoperability Platform (IOP) that enables sensors, devices, appliances and services to exchange information on a semantic level. This IOP has been the starting point of our research, and as the project progressed, our research results have been integrated into the interoperability platform.

The research was carried out in close cooperation with another PhD candidate, Gerrit Niezen (Niezen, 2012). Both approaching the project from our own field of expertise (i.e. Industrial Design and Computer Science), we worked towards a joint result that brings together theories from the fields of design and computer science in the context of ubiquitous computing. This thesis reports on the designerly approach to interoperability and, to fully explain the work, also partially introduces the novelties at the level of software architecture and ontology development. I often refer to the work of Gerrit Niezen, and where necessary summarise parts of his thesis. Although our work had to have significant overlap, in this paragraph the details of the overlap and complementarity are untangled so it is clear who did what.

The design explorations described in Chapter 3, combine my work with the work Niezen and two master students. The first design iteration was a combined effort of which the result was offered to students as a starting point for their own projects (under my guidance). The results of these student projects are described in this thesis, and served as input for the design process. Chapter 4 describes design work of me, together with an implementation of this work in an evaluation setting, combined with the work of Niezen and other project partners. The evaluation described in Chapter 4 is considered to be exclusively my work. The development of a theory of *semantic connections* as is described in Chapter 5, is a combined effort, as well as the implementation that was described. The interaction design work involved and the evaluation in terms of feedback and feedforward was mainly performed by me. The ontologies described in Chapter 4 and 5 were were the result of a joint effort. More specificly, I made a substantial contribution to the ideas behind the interaction design specific elements the ontologies aim to describe. Chapter 6 is exclusively my work. In the remainder of this thesis, the first person plural style of writing is used, to improve readability and coherence with the work that was published earlier.

The thesis at hand consists of three parts. Part I (chapter 1 & 2) introduces the context of the research and reviews relevant literature, theory and related work. In Part II a series of design explorations is described, exploring and testing various ways of interacting with otherwise invisible connections, enabling users to manage the connections and information exchanged between devices in their homes. Chapter 3 describes three designs, including our first design iteration and two designs based on our ideas of semantic connections at that time. The two last mentioned designs were done by Industrial Design master students, Matthijs Kwak and Jeroen Peeters, as part of their M1.2 (2nd semester of 1st year master) research project. Chapter 4 introduces a *tangible* approach and an *augmented reality* approach to interface with semantic connections. Building on the SOFIA IOP, these design explorations were brought to live and tested in a user study which integrated the work of several partners within the SOFIA project. These user studies aimed at eliciting the users' mental models to investigate how users conceptualise the connections and the information they carry.

Part III focusses on building a theoretical framework from the lessons we learned during the design explorations. Chapter 5 introduces a *Semantic Connections Theory*, where we view smart environments in terms of connections and relations in-between the artefacts (*Smart Objects*) and people within the environment. In this theory semantics is pivotal. Based on our theory, a framework is introduced in Chapter 6 in which meaning, action and function are coupled. The framework is based on existing theories of product semantics and user interaction. It uses and extends these theories beyond their traditional focus on the appearance of objects and interaction with them in isolation, towards designing for systems of inter-operating products. The framework is evaluated by discussing examples of products and design prototypes that implement the underlying theory. The framework is aimed to help designers and developers of interoperable smart objects to deal with the challenges in contemporary interaction design. Chapter 7 concludes the thesis, and draws lessons and implications of our theory and framework on design in a more general sense, followed by a discussion of future research directions.

Chapter 2

Designing for Interoperability: theory and technologies

2.1 In this chapter

This chapter discusses trends in industry, summarises relevant related work and aims to illustrate how we can make use of the existing work as a foundation for our own. We introduce the concept of (semantic) interoperability and its challenges. It then discusses how we aim to solve the technological challenges of interoperability and how we use the opportunities and implications of this solution as a starting point for the design explorations. We discuss related work as a result of the past decades of research in the fields of Human Computer Interaction and Ubiquitous Computing. We conclude with a review of theories of design, product semantics and interaction models. We attempt to relate these theories to our concept of semantic connections, and discuss how we may use and extend these theories beyond their traditional focus on the appearance of objects and interaction with them in isolation, towards designing for systems of inter-operating products.

2.2 State-of-the-art

That devices can exchange information and functionalities with each other is a fundamental idea in this thesis. Before I discuss related work, research as well as the theoretical and technological background in Sections 2.3 through 2.6, I will first briefly discuss the state-of-the art and give examples of products and the emerging problems we target. Already

---

This chapter is based on:

Vlist, B. van der, Niezen, G., Hu, J., & Feijs, L. (2012). Semantic Connections: a New Interaction Paradigm for Smart Environments. In L.- L. Chen, T. Djajadiningrat, L. Feijs, S. Fraser, S. Kyffin, & D. Steffen (Eds.) *7th International Workshop on the Design & Semantics of Form & Movement (DeSForM) 2012.* pages 16–26. Wellington, New Zealand: Koninklijke Philips Electronics N.V.

Niezen, G., Van der Vlist, B.J.J., Hu, J., & Feijs, L.M.G. (under review). Semantic Connections Theory: Enabling Interaction Designers and Developers to Create Interoperable Smart Objects. *ACM Transactions on Interactive Intelligent Systems (TiiS).* 32 pages.

today there are ways to make products talk to one another. Basic data exchange between products has been established. However, despite the technical solutions, users have not yet managed to operate among these devices in a meaningful way. Kuniavsky (2010) sums up five main approaches to interconnecting devices that have emerged over the years:

- *Manual configuration:* This is the most widely used method, however it is time-, documentation-, and knowledge intensive. It is robust, but requires users to manually configure interconnection using manuals, configuration screens and device drivers. Setting up wireless networks, configuring and connecting devices and appliances to them, are examples of the tasks involved in this approach.
- *Vendor lock-in:* This approach makes it easy to interconnect a single manufacturer's devices, discouraging users to connect devices from other manufacturers. A good example is Apple's strategy of keeping products within their ecosystem. It allows Apple's products to talk to each other easily (e.g. Airplay), but does not allow easy interoperability with other products from other manufacturers.
- *Middleware:* The middleware approach involves special software that translates the communication between different standards to connect devices that use incompatible communication standards. This approach is limited to only connecting devices that developers of the middleware had access to at the time of development, which severely limits its application. Middleware for consumers products can be found in the multimedia domain e.g. Multimedia Home Platform (DVB-MHP) and Universal Home API (UHAPI). In the best case this leads to standardised protocols, which guarantee interoperability for conforming devices.
- *Introduction:* Introduction uses service discovery protocols (e.g. Universal Plug-and-Play; UPnP) for devices to passively communicate their existence and capabilities to other devices. This allows users to manually introduce devices to one another using passwords (e.g. Blue-tooth pairing uses passwords generated by one of the devices that facilitates introduction).
- *Matchmaking:* A trusted device is used in matchmaking to introduce two devices to each other to establish communication. For example, the SIM-cards used in mobile phones act as a trusted device to establish the connection between mobile phones and GSM networks.

Although the above mentioned strategies have their implications on the user experience of connecting devices together, each strategy still allows for different ways for users to establish the actual connections. During the course of this research, several innovations in terms of user interaction across devices have surfaced from industry, such as Apple's AirPlay[1] (Figure 2.2 and 2.1), AirPrint[2] and iCloud[3], Nokia's NFC[4] and Samsung's Allshare[5]. The outlook on technologies currently being developed is also worth mentioning, like the

---

[1] http://www.apple.com/ipad/features/airplay.html

[2] http://www.apple.com//ipad/features/airprint.html

[3] http://www.apple.com/iphone/icloud/

[4] http://europe.nokia.com/nfc

[5] http://www.samsung.com/uk/article/galaxy-guide

Figure 2.1: Selecting AirPlay enabled speakers

Figure 2.2: AirPlay icon in a smart phone's GUI

Android at home[6] developments (where the Android OS is running on everyday devices and appliances to enable interoperability). These innovations make it possible and easy to exchange certain types of data among devices. Although this list is not intended to be exhaustive, these recent developments coming from the large technology companies show directions which device interconnectivity might develop into, in the near future.

2.2.1 Media sharing and cloud storage

Apple's AirPlay, AirPrint and iCloud are examples of the *vendor lock-in* strategy as was described above. They are considered to work well within the ecosystem of products from the same manufacturer (or sometimes also from certified manufacturers). But neither are they available to users of products manufactured by others nor do they work with products users may own from third parties. These solutions are also limited to certain domains, like AirPlay and Samsung's Allshare only apply to the media domain (for streaming music and video) and AirPrint only offers printing functionality. Scalability is also an issue. Even though it is reasonable to believe that these solutions work well for users with an average number of products (in their restricted domains), the same user interaction style cannot easily be extended to a larger number of devices or different kinds of devices, across domains. For example, with AirPrint one may select the target printer from a list of available printers. It is fair to assume that this way of selecting a printer works fine for home use, where usually only one or perhaps a few printers are available. The same holds for selecting a target for your audio, video or images. It is reasonable to assume that the number of AirPlay enabled devices that someone owns can be remembered, especially when these are given a unique and descriptive name. When interoperability has been established across domains, the number of devices increases and the type of devices diversifies. Moreover, the interaction style proposed by Apple requires the source devices to have a GUI for selection of the target, which potentially excludes devices without GUIs to take part. This way of

---

[6] `http://www.androidathome.com/`

Figure 2.3: A user paring an NFC enabled smart phone with the Nokia 360° speaker [image source: screenshot taken from advertisement video, available from: `http://nokia.ly/ingpDX`*]*

selecting sink devices therefore limits the appropriateness of the interaction style and ignores much of the interaction potential a smart environment with physical devices has to offer.

Allshare is Samsung's trademark name for DLNA (Digital Living Network Alliance) certified devices. It builds on UPnP (Universal Plug and Play) media management, discovery and control. Allshare products offer user friendly ways of wirelessly displaying media such as images and video across devices, for example, from mobile devices on the TV's screen. It uses the mobile device's GUI to establish the connection and select the media to be shared.

iCloud uses yet another strategy and is a good example of a *cloud (storage) service*, where content is stored in "the Cloud" that refers to on-line storage at a server from the cloud-service provider. Similar popular cloud storage services include e.g. Dropbox, SugarSync and ZumoDrive, but iCloud is well integrated in the OS and does not require third party apps. Cloud services work well to synchronise content between various devices and allow access to data stored remotely. However, they do not allow for much control on what data is available and where, or requires browsing and searching to find the specific item one is looking for. Moreover, cloud services such as iCloud mainly focus on availability of media on multiple devices owned by one user. Others do allow sharing between users by means of public and shared folders, but do this in a rather file-oriented fashion.

2.2.2 Near-Field Communication and proximal interactions

Interesting from a user interaction point-of-view is Nokia's NFC (Near Field Communication) which entails embedding RFID (Radio Frequency Identification) technology in their devices[7]. By touching two NFC enabled devices together, a connection is established and

---

[7] NFC builds on RFID technology but allows two-way communication between NFC enabled devices instead of earlier RFID systems that could only read passive *tags*. NFC enabled devices can still read

data is exchanged over the near field channel. Alternatively, the near field channel can be used to identify devices and exchange configuration information to establish a connection using another wireless communication technology. An example is the Nokia 360° speaker system (Figure 2.3), where music can be streamed to a speaker from a smart phone wirelessly, after pairing the two devices by touching them together. Similar interactions can be found on the market today in apps like Bump[8], which allows for wireless exchange of data by bumping two phones together (location and time of the "bump" event are then used to exchange information like contacts and images using the internet connection).

2.2.3 Device Pairing

On a practical level, a part of the design problem targeted in this thesis may be reduced to designing *device paring*, however this thesis introduces an approach that goes beyond this. While many of the devices that users own today can be interconnected, the process of making the actual connections and exchanging the information between them is often difficult as configuration details and connectivity settings are often nested within menu structures. Even with the connections in place, exchanging the actual information is cumbersome, and users may have to dig into the information management structures to find the information they want to exchange. In contrast, from a user's perspective, a connection might be easy to be seen, simply because the devices are physically related, close to each other, or maybe even touch each other. Information to be shared might be currently on the screen. Even beyond sharing basic information, users may desire the devices to share their functionalities. This thesis proposes another, more extensive solution, in which easily manageable connections can be used to exchange information and functionalities. This enables users to create ensembles of devices and functionalities going beyond those that were thought of when the devices were designed and developed.

2.2.4 Multi-screen Design Patterns

A closely related field of research is task migration, focussing on moving a partially completed task from one information appliance to another. Especially with the recent availability of smart phones and tablets, solutions for continuing a task on another mobile device are emerging. Targeted tasks range from continuing listening to music seamlessly at another device when you move from the one place to another, continuing to work on the same office document on your smart phone or tablet computer after you leave your desktop computer. Furthermore, in entertainment activities users often have multiple screens at their disposal which may potentially be used to create an integrated user experience. To deal with multi-screen situations, Precious Design Studio [9] has developed design patterns to approach such situations (see Figure 2.4). These design patterns might be applicable in more general terms concerning user interaction, which will be explored and discussed in Part III of this thesis.

---

passive RFID tags and can also simulate a tag.

[8] `http://bu.mp/`

[9] `http://precious-forever.com/`

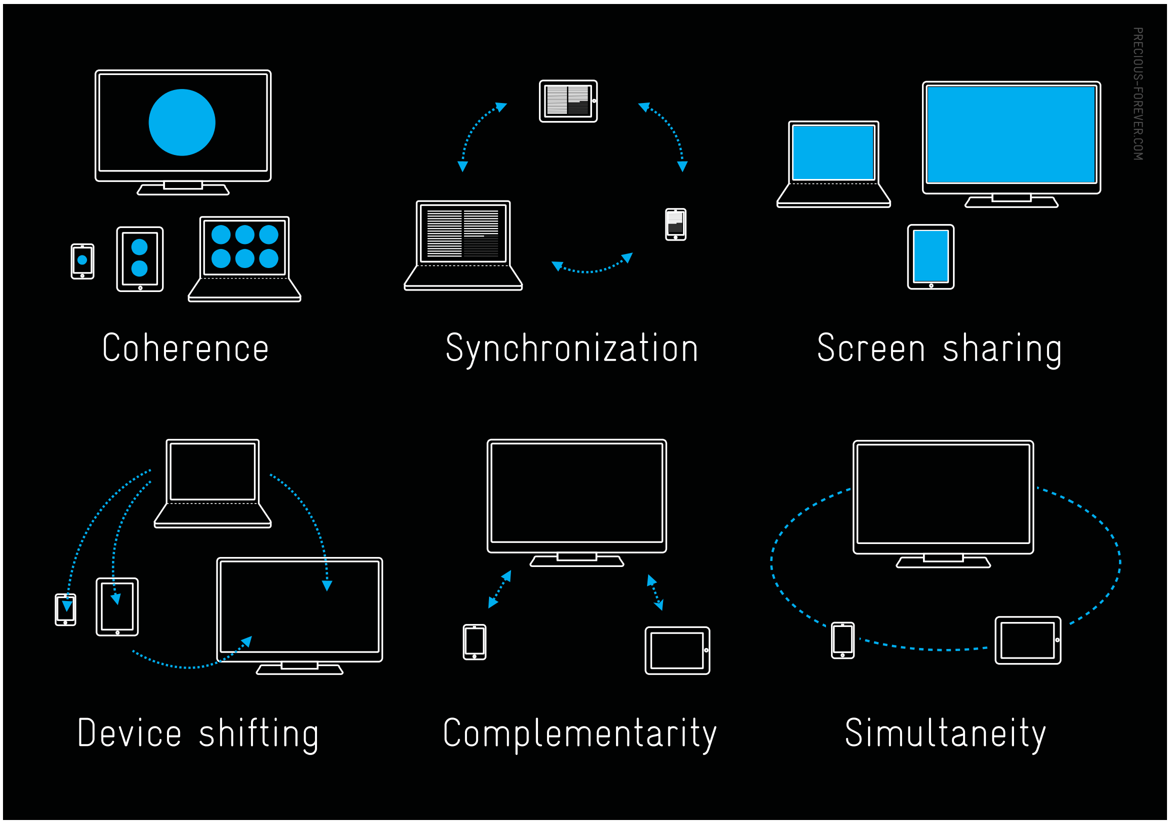


Figure 2.4: Design patterns to design for multiple screens [image source: Precious Design Studio, available from: `http://precious-forever.com/2011/05/26/patterns-for-multiscreen-strategies`*]*

2.2.5 "There is an App for that"

With the introduction of smart phones and tablet computers, applications (or a more popular term: apps) to run on these devices gained popularity. At the same time, other everyday devices became networked and internet connected. Today many examples of such connected devices form an *Internet of Things*. Networked weight-scales, fitness sensors, intelligent shoes, blood pressure meters, thermostats, lights, etc. are examples of devices that connect to the internet. Today, these internet enabled devices connect predominantly to apps (linked to a web account) on smart phones and tablets, and data is synchronised and uploaded to web services (e.g. Withings scale[10] and blood pressure monitor[11], e-thermostat[12], Fitbit[13]). Additionally, apps on tablets and smart phones are used to remotely control lamps, thermostats and other home appliances. Even though these developments are gaining popularity and bring many practical applications to users' homes to make their lives easier, it seems that we are heading into a reality where we need a smart phone, web tablet or an internet browser to perform rather simple tasks (Geere, 2011). While in contrast, much of the data that is generated by these internet connected devices can potentially be used by other devices to have even more added value. Without having users explicitly deal with the information that these devices generate.

---

[10] `http://www.withings.com/en/bodyscale`

[11] `http://www.withings.com/en/bloodpressuremonitor`

[12] `http://www.icy.nl/consument/producten/e-thermostaat`

[13] `http://www.fitbit.com/uk/product`

2.2.6 An ecology of smart objects

With the observation and discussion of the trends in the previous sections, we can now start to envision how a reality with interconnecting and interoperating smart objects may look like. In this *ecology of smart objects*, users will need to develop an understanding of the connections and information flowing between the objects to take control over the new possibilities such an environment has to offer.

Although already today our environments contain numerous digital devices such as smart phones, tablets, TV screens, notebooks, desktop computers, people have not yet achieved seamless operation among them. Each and every one of these devices demands our attention, uses a different user interface and allows access to only few other components within the environment. While many of the devices are, or can be networked, the process of making the actual connections and exchanging the information between them is tedious and difficult without networking knowledge. There are examples of easy ways to share information among devices (described in Section 2.2), however, they only work well within the ecosystem of one manufacturer or within one domain. In contrast to these rather complicated ways of interconnecting devices, user's may perceive the devices to be easy to connect, since they are physically close to each other and have overlapping functionalities. The information they want to share might currently be on the screen and exchanging the information could form part of the interaction, depending on the user's intention. The nuisances described here are unnecessary from a users perspective and make their "smart" products look ignorant and unaware of each other.

Some of the irritations that users face today are a consequence of market mechanisms that imply different goals for different stakeholders. Designers of devices need to have a strong device-oriented view, whereas users' goals are often more easily resolved with a system-oriented view. Designers are concerned with the functionality and usability of the device at hand, possibly harmonizing its usage over the range of products provided by this specific manufacturer. Users, on the other hand, find themselves with a set of devices and services from different manufacturers, or even different industries. As an example: users still have to set the clocks of many devices, even if they are all connected to each other or to the internet, and could potentially obtain their time automatically from a time server. If you want to share an image that is currently at display on your mobile phone's screen with your friend sitting next to you, the steps required to do so seems unnecessarily cumbersome. Seemingly easy tasks are not possible, because at development time, nobody thought about it, and only minimal cross-device and in particular cross-vendor capabilities have been implemented.

Even with the technical complications cleared away, the challenge for designers remains. When the focus of the design task at hand is shifted from designing an intuitive, pleasant and coherent user experience for a single device to that of a connected device which is part of a larger system, the design task changes in nature and increases in complexity. Suddenly the device needs not only work in isolation, but also work together with any other devices that have overlapping or complementary functionalities. Interestingly, some of the *emerging functionalities* that may surface when two devices are connected, may only exist in the combination of the devices but not in any of the devices in isolation. It seems that the connections themselves or more generally the relationships between the devices are key when designing for ecosystems of devices. Perhaps even more important, the key is the meaning of the relationships between the devices.

To investigate the meaning of the connections, we aim to look at physical connections between artefacts, as well as conceptual and metaphorical connections that users have in their minds. They all play an important role. Devices can be physically connected by wired or wireless communication, but people also tend to group artefacts that are not physically connected together by finding resemblances in their meaning. In smart environments with many interconnected and interoperable objects—hiding their physical connections—these conceptual and metaphorical connections become even more valuable, and maybe even crucial for the understanding of a smart environment. Without this understanding there is the risk of engendering a mismatch between the system's model of interaction and the user's mental model of the system. In these conditions, using explicit design semantics may be used to help users to construct helpful mental models, in order to minimize the mismatches.

We believe it is crucial for users to have an overview of, and preferably high-level control over connections between networked devices. Moreover, new functionalities that emerge by interconnecting several devices have to be introduced to users in an appropriate way. Enabling users to actively explore connections and connection possibilities may contribute to the understanding and transparency of the smart environments and their services. We consider this a worthwhile design goal.

2.3 Semantic Interoperability

Interoperability is the ability of a system or a product to work with other systems or products. In the SOFIA project, interoperability is achieved by using a semantic interoperability platform (IOP), where information is exchanged between devices on a semantic level. In this section, the underling principles and technologies of semantic interoperability are introduced, followed by a description of the SOFIA IOP itself.

2.3.1 Ontology and the Semantic Web

Pivotal to our research is the notion of semantics, or meaning[14]. The envisioned semantic interoperability is about making the meaning of data machine-readable and exchangeable by annotating data with semantics. It even goes further, as the meaning arises from the way the data is stored, by storing data in triples formed by subject-predicate-object constructions and defining object properties and relationships between them using the predicates (Allemang and Hendler, 2008; Hebeler et al., 2009).

The technology that is used to achieve semantic interoperability was originally developed in the context of the Semantic Web. The Semantic Web is a vision aiming at extending the current Web by adding semantics to its documents (websites). The added semantics should be machine-readable and understandable, enabling machines to reason on the Web's content using logic. In this vision semantics will be added by creating *ontologies*, describing and hierarchically categorising concepts and their relationships. The term ontology, as it applies to the field of computer science, originally comes from philosophy. In philosophy, the word ontology refers to the study of being or existence (Jepsen, 2009) and is used

---

[14] Semantics is the study of meaning. The word *semantic* is frequently used in this thesis, in two different contexts. It refers to the use of the theory of *product semantics*, the study of how people attribute meaning to artefacts, and Semantic Web technologies, which are introduced in this section.

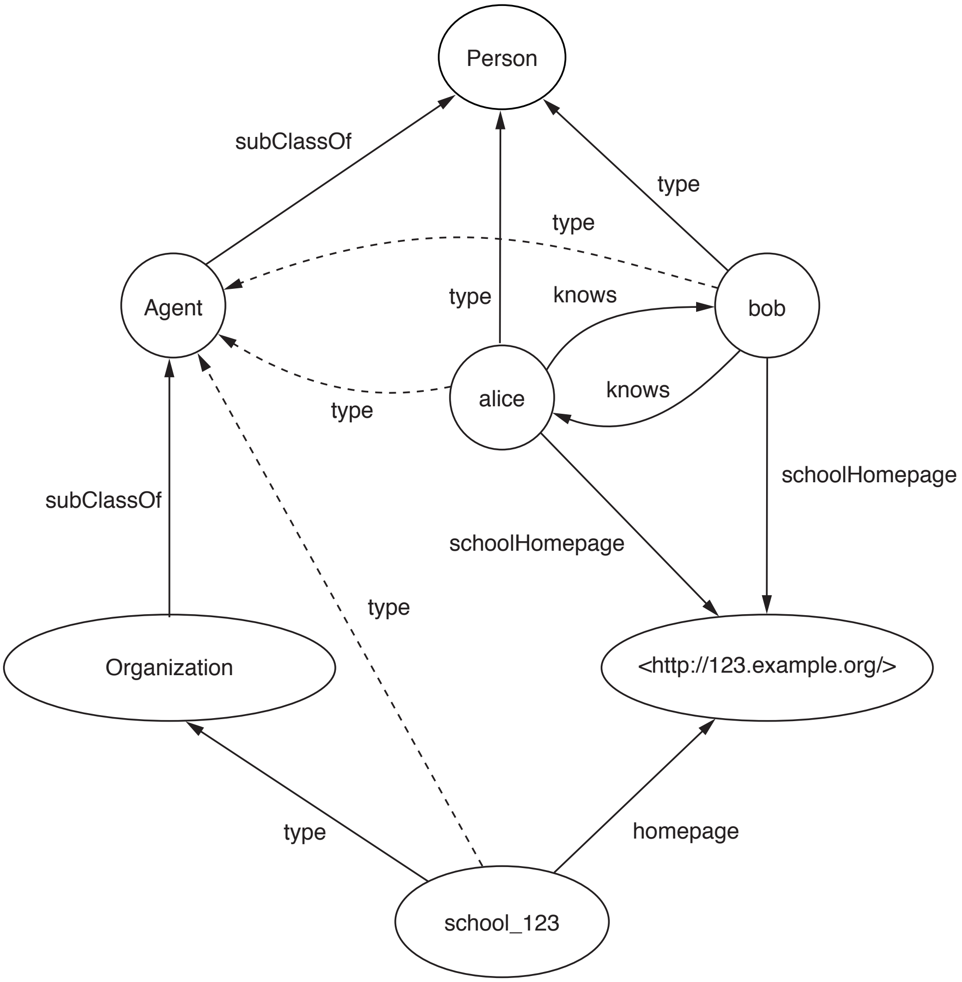


Figure 2.5: Example of a FOAF ontology, describing the relationships between various entities. Dotted lines are inferred relationships. [image source: Dan Brickley on `http://www.flikr.com`]

to hierarchically categorise all entities that make up reality. A graphical representation of a FOAF (Friend of a Friend) ontology is shown in Figure 3.6. FOAF is an ontology to describe persons, their activities and their relationships with other persons and objects.

Ontologies are (in the context of Semantic Web) used to describe and represent items of knowledge (e.g. ideas, facts, things) in a way that defines the relationships and classifications of concepts within a specified domain of knowledge (Jepsen, 2009). This makes ontologies domain specific, as an ontology represents all the knowledge within a domain and the relations between the concepts that make up the domain. Hepp (2007) gives a comprehensive and usable definition of ontologies, and discusses common misconceptions and different notions that have been assigned to the term. Although there are differences between ontologies, there is also general agreement on many issues; Chandrasekaran et al. (1999) sum up the following list:

- There are objects in the world.
- Objects have properties or attributes that can take values.
- Objects can exist in various relations with each other.

- Properties and relations can change over time.
- There are events that occur at different time instants.
- There are processes in which objects participate and that occur over time.
- The world and its objects can be in different states.
- Events can cause other events or states as effects.
- Objects can have parts.

2.3.2 Upper-Ontology

If one uses a higher-level ontology and extends this ontology into a domain specific ontology, this higher-level ontology is usually referred to as an upper-ontology or foundational ontology. The discussion about the use and feasibility of developing "one" upper-ontology is controversial and still ongoing, however, upper ontologies have shown important use during development time of domain ontologies (Scherp et al., 2011). Upper-ontologies can be used to align many different domain ontologies, relating them to very general concepts that are the same across these different domains. Scherp et al. (2011) propose a layered approach comprising domain ontologies and core ontologies that are based on a foundational ontology. They have been a number of upper-ontologies (or foundation ontology) defined. Popular and matured ones are:

- DOLCE (DOLCE-lite and DOLCE-litePlus) [15]
- ExtendedDnS[16]
- COSMO[17]
- SUMO[18]
- Cyc (or a subset of that: OpenCyc[19])

For the Semantic Web, the most popular language to describe ontology is OWL (Web Ontology Language) (Heflin, 2004). OWL is built on RDF (Resource Description Framework) that provides a way to describe information in a structured way using triples and commonly used *predicates* like `rdf:Type`, indicating that an object is of a certain type. However, RDF alone does not provide the expressive power that ontologies can potentially have. OWL does provide this added semantics by means of defining properties and restrictions such as symmetry and the `owl:sameAs` property. These properties allow for reasoning; as for an symmetric property the inverse can be inferred (e.g. brother hasSibling sister then sister hasSibling brother).

2.3.3 Semantic Reasoning

Semantic reasoning about statements defined using OWL, is performed by a semantic reasoner. This reasoner is a piece of software that is capable of inferring additional statements

---

[15] http://www.loa.istc.cnr.it/DOLCE.html

[16] http://www.loa.istc.cnr.it/Papers/D07_v21a.pdf

[17] http://micra.com/COSMO/

[18] http://www.ontologyportal.org/

[19] http://sw.opencyc.org/

which can be derived from the original descriptions. This is a very powerful mechanism in the context of interoperability. When properly aligned to a well defined ontology, devices can insert certain information about themselves into a triple store to describe their capabilities with their types and properties. A semantic reasoner can then be used to infer and insert additional information which creates a more complete model or *inferred model* with additional, inferred information based on the original *asserted model*. As an example: a device inserting a triple including its identity and its type, e.g. `device1 rdf:Type SmartObject`, will inherit all the super-classes and (OWL) restrictions of the objects of class *Smart Object*.

2.3.4 Smart M3

The Smart M3 (multi-device, multi-vendor, multi-domain) architecture is an interoperability platform based on a blackboard architectural model that implements the ideas of space-based computing (Honkola et al., 2010). It consists of two main components: a SIB (Semantic Information Broker) that acts as a common, semantic-oriented store of information and device capabilities, and KPs (Knowledge Processors), virtual and physical smart objects that interact with one another through the SIB. Various SIB implementations exist that conform to the M3 specification, of which Smart-M3 was the first open source reference implementation released in 2009[20]. The M3 architecture is the main underlying component of the SOFIA IOP, and is therefore used as a backbone in our research prototypes and was a starting point to develop our *Semantic Connections Theory*.

The SOFIA software platform, which was developed throughout the course of the SOFIA project, utilizes the blackboard architectural pattern to share information between smart devices, rather than have the devices explicitly send messages to one another. The blackboard architectural pattern inherits its name from a traditional blackboard, which represents a place where "experts" can together write information that everyone can see to solve a problem. When this information is also stored according to some ontological representation, it becomes possible to share information between devices that do not share the same representation model, using the *semantics* of that information (Oliver and Honkola, 2008). Rather than promoting the compatibility within one specific service solution in terms of aligned protocols, it addresses information-level compatibility and the collaboration between different producers and consumers of information on a more abstract level. The goal is that devices will be able to interact on a semantic level, utilizing (potentially different) existing underlying services or service architectures.

Part of this solution is a *core ontology* that describes commonly used concepts, and also model domains more completely, in a formal ontology that is expressed in RDF.

Ontologies lend themselves well to describing the characteristics of devices, the means to access such devices, and other technical constraints and requirements that affect incorporating a device into a *smart environment* (Heflin, 2004). Using an ontology also simplifies the process of integrating different interaction approaches, as the different entities and relationships in the SIB can be referred to unambiguously. Because communication via the SIB is standardized, integrating cross-vendor implementations is also simplified, and technical incompatibilities can be captured by the ontology that the user can be made aware of.

---

[20] `http://sourceforge.net/projects/smart-m3/`

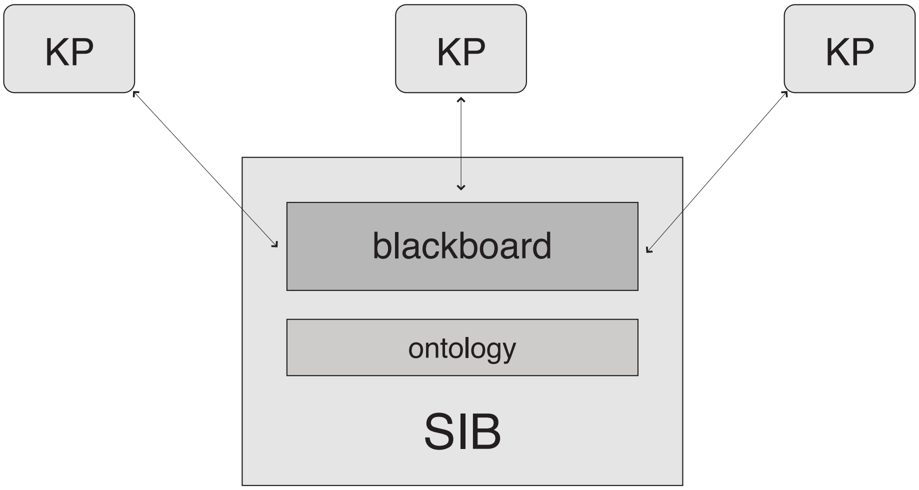


Figure 2.6: Smart-M3 infrastructure model, showing the interaction between the main components: Knowledge Processors (KP's) and the Semantic Interaction Broker (SIB).

Ontologies are used to enable the exchange of information without requiring up-front standardization. A notable feature of the SOFIA IOP is the capability to subscribe to changes of data (stored as triples) in the triple store, and be notified every time these triples are updated, added or removed. Smart-M3 takes the blackboard and publish/subscribe concepts and implements them in a lightweight manner suitable for small, mobile devices. Figure 2.6 shows a simplified overview of the Smart-M3 infrastructure.

For applications, a Description Logic (DL) based ontology can be created in OWL. In the ontology we developed during our research, all user interaction within the system is described in terms of interaction events (Niezen et al., 2010). To enable our semantic connections interaction model the connections between the devices need to be modelled. A `connectedTo` relationship can be added or removed between two existing devices in the ontology. It should be noted that this relationship is both symmetric (only in our first implementations) and irreflexive. An irreflexive property is a property that never relates an individual to itself. This allows us to restrict our model by not allowing a `connectedTo` relationship from a device to itself.

We aim to enable users to explore and make configurations in a smart environment, on a high semantic level without bothering them with low-level details. We believe this can be achieved by making use of Semantic Web technologies and the SOFIA IOP, to achieve and support semantic interaction in a smart home environment, as is described in (Niezen et al., 2010; Niezen, 2012) and will be explained in more detail in Part III of this thesis.

2.3.5 Ubicomp ontologies

Besides the development of upper-ontologies, of which a list of examples is briefly mentioned in Section 2.3.2, there also exist many domain ontologies and ontologies that were developed in the domain of ubiquitous computing. Various Ubicomp ontologies have been developed for context-aware computing and more specifically to model devices and their capabilities. Examples of ubicomp ontologies are:

- SOUPA (Chen et al., 2004)

- BDI (Bratman, 1987) and the MoGATU ontology
- Gaia (Ranganathan et al., 2004)
- CAMUS (Ngo et al., 2004)
- SPICE DCS (Villalonga et al., 2009)

Other related (event) ontologies are:

- The Event Ontology[21]
- DOLCE+DnS UltraLight (DUL)[22]
- Linked Open Descriptions of Events (LODE)[23]

Although the development of an ontology for user interaction was a large part of our effort for the SOFIA project, discussing ontologies that are considered state-of-the-art in ubiquitous computing is considered to be outside of the scope of this thesis. The thesis of Niezen (2012) fully describes the development of the ontologies and therefore also includes a thorough discussion of the state-of-the-art. The ontologies described later in this thesis (Sections 4.6.2 and 5.4.1 build upon these existing ontologies, but with a stronger focus on user interaction-related aspects. Please refer to Sections 2.2 and 8.1 of (Niezen, 2012) for an overview of existing ubicomp ontologies and relevant related ontologies.

2.4 Interaction models

An interaction can be divided in seven stages of interaction. Norman (1998) describes an interaction using seven stages of action:

1. Goal - What we want to happen
2. Intention to act - An intention to act as to achieve the goal
3. Sequence of action - The actual sequence of actions that we plan to do
4. Execution of action sequence - The physical execution of the action sequence
5. Perceiving the state of the world - What happened as a result of your actions
6. Interpreting the perception - Interpreting the perception according to our expectations
7. Evaluation of interpretations - Comparing what happened with what we wanted and expected to happen

In the interaction models described below, these stages are described using different levels. We discuss the interaction models that are considered relevant to our work.

2.4.1 Foley's linguistic model

Foley's model defines the following levels (de Ruiter, 1988):

- Conceptual level - definition of main concepts and the possible commands (equivalent to user model (Buxton, 1983))
- Semantic level - defines the meaning of the commands

---

[21] `http://motools.sf.net/event/event.html`
[22] `http://www.loa.istc.cnr.it/DOLCE.html`
[23] `http://linkedevents.org/ontology/`

- Syntactic level - describes the form of the command and parameters (syntax)
- Lexical level - defines lowest input symbols and their structure

Buxton (1983) extended Foley's model to include a pragmatic level that defines the issues of gesture type (e.g. pointing with a tablet versus a mouse), device location and spatial placement. While a keystroke will be defined at lexical level, the homing time and pointing time will be defined at pragmatic level. Buxton had the foresight (Buxton, 1983) to comment on the difficulty of multi-device environments where the different levels are managed by different entities. He noted that it has a strong effect on the semantics of the interactions that could be supported: If the computing environment is managed by one entity, the semantics and functional capabilities by another, and the user interface by yet another, there is an inherent danger that the decisions of the one will adversely affect the other.

Dix et al. (2008) noted that Buxton's work emphasized the way in which the lexical level design of the physical interface can simplify syntax in interaction. These ideas have been extended by Ullmer et al. (2005) into a *digital syntax* that is embodied by the physical design, resulting in a grammar for mapping physical relationships into digital representations.

2.4.2 Arch/Slinky model

Bass et al. (1992) contends that no single software architecture will satisfy all the design goals of an interactive system. With the Arch/Slinky model the buffering of a system from changes in technology was selected as the most important criteria. Here are some of the other design criteria they defined, which we consider to be especially important to ubiquitous computing systems:

- target system performance (e.g. size and speed)
- buffering from changes in application domain and hardware platform
- conceptual simplicity
- complexity of specification
- target system extensibility
- compatibility with other systems

They define an *application* to be the total system that is developed for its end users, while the *application domain* is the field of interest of, or reason for, the application. They also extended the definition of User Interface Management Systems (UIMS) to a User Interface Runtime System (UIRS) - the run-time environment of an interactive application.

The Arch model creates a bridge between the physical interaction device and the application domain. The following 5 components are defined:

- Interaction Toolkit Component - implements the physical interaction with the user (also called physical level)
- Presentation Component - provides a set of implementation-independent objects, e.g. a "selector" object can be implemented by both radio buttons or a drop-down menu in a GUI (also called lexical level)
- Dialogue Component - does task-level sequencing and maps between domain-specific and UI-specific formalisms (also called dialogue level)

Table 2.1: Nielsen's virtual protocol model

Level	Layer	Example
7	Goal	Want to delete the last section of a letter in a word processor
6	Task	Delete the last six lines of the edited text
5	Semantic	Remove a line with a given line number
4	Syntax	DELETE 27
3	Lexical	DELETE, DEL or other lexical token
2	Alphabetic	Letter "D" or other lexeme
1	Physical	User presses D-key on keyboard

- Domain Adaptor Component - triggers domain-initiated tasks, organizes domain data, detects and reports semantic errors (also called functional core adapter)
- Domain-specific Component - controls, manipulates and retrieves domain data

The separation of functionality into the different components was done to minimize the effects of changes in technology. The Slinky metamodel is a generalization of the Arch model, providing a set of Arch models with different weights assigned to each component.

2.4.3 Nielsen's virtual protocol model

Nielsen's virtual protocol model for human-computer interaction (Nielsen, 1986) was inspired by the 7-layer OSI model for computer networks, as shown in Table 2.1. Although the examples given in Table 2.1 are outdated, the ideas behind it are still relevant. The task layer deals with general computer-related concepts that are representations of the real world concepts from level 7, that may have to be realized by a sequence of operations from level 5. *Level 5 handles the meaning of the interaction, where there are a finite number of concepts in the system and each have an exact definition.* The lexical tokens on level 4 ("DELETE 27") realizes the semantic command "remove a specific line". Lexemes are information-carrying units that do not have any meaning by themselves. Screen layout could be considered a two-dimensional syntax that can also be defined in terms of lexical tokens. Direct manipulation (Shneiderman, 1997) could be seen as using the syntax level to mirror the semantics level.

Nielsen compared his model to Foley's model and Buxton's extended version, as well as an earlier model by Moran called Command Language Grammar (CLG) that consisted of six levels: task level, semantic level, syntactic level, interaction level and device level (Nielsen, 1986). He noticed that all models seem to agree on the visible (defining the form) part of the communication, as well as the invisible part (defining the meaning). Nielsen noted that Foley's model does not include the real-world concepts of his goal level, or the hardware-related detail of his physical level.

According to Nielsen the purpose of his model is to improve the usability of software. He noted that some people will consider it a useful abstraction, while others will prefer other models, similar to how everybody has their own favourite programming language.

2.4.4 Tangible Interaction (MCRpd) Model

Tangible Interaction aims for bridging the digital with the physical world, by giving physical control over digital bits of information. It was introduced by Ishii and Ullmer (1997) and later developed into a conceptual framework for Tangible User Interfaces (TUIs) and the MCRpd interaction model (Model-Control-RepP-RepD; where RepP is an abbreviation for physical representation and RepD for digital representation). The interaction model describes tangible interfaces, that give physical form to digital information, using physical artefacts as both representations and controls for digital media (Ullmer and Ishii, 2000).

The MCRpd model (Figure 2.7) is based on the MVC (Model View Controller) model, which describes the separation of the visual representation (view) provided by the graphical display and the control of a GUI, mediated by the mouse and keyboard. For tangible user interfaces, the *control* element from the MVC model is made physical by introducing physically represented digital information, and the *view* element is replaced with physical representations (RepP) for the physically embodied elements, and with digital representations (RepD) for the mediated components without physical embodiment (e.g., light and audio) (Ullmer and Ishii, 2000).

The MCRpd model highlights the following three key characteristics of TUIs (Ullmer and Ishii, 2000):

- Physical representations (RepP) are computationally coupled to underlying digital information (model).
- Physical representations embody mechanisms for interactive control (control).
- Physical representations are perceptually coupled to actively mediated digital representations (RepD).
- Physical state of tangibles embodies key aspects of the digital state of a system.

The *semantic connections interaction model* introduced in Section 4.2.1 was inspired by the MCRpd model

2.4.5 The ASUR interaction model

ASUR is a notation-based model to describe user-system interaction in mixed interactive systems (Dubois and Gray, 2008) at design-time. It describes the physical and digital entities that make up a mixed system and uses directed relationships (arrowed lines) to express physical and/or digital information flows and associations between the components.

Both components and relationships may have characteristics. For components, this includes the location where the information is perceived (e.g. top of table) and action/sense required from the user (e.g. sight, touch or physical action). For relationships, characteristics include the dimensionality of the information (e.g. 2D or 3D) and the type of language used (e.g. text or graphics).

A sequence of such entities and their relationships in an interaction forms an *interaction path*. The interaction exchange or action between elements in the path is conducted via one or more *interaction channels* along which information or action is communicated. An interaction channel may be described in terms of its properties, either physical or digital depending on the channel, e.g. a digital channel may be described in terms of bandwidth, uptime and the nature of the connection.

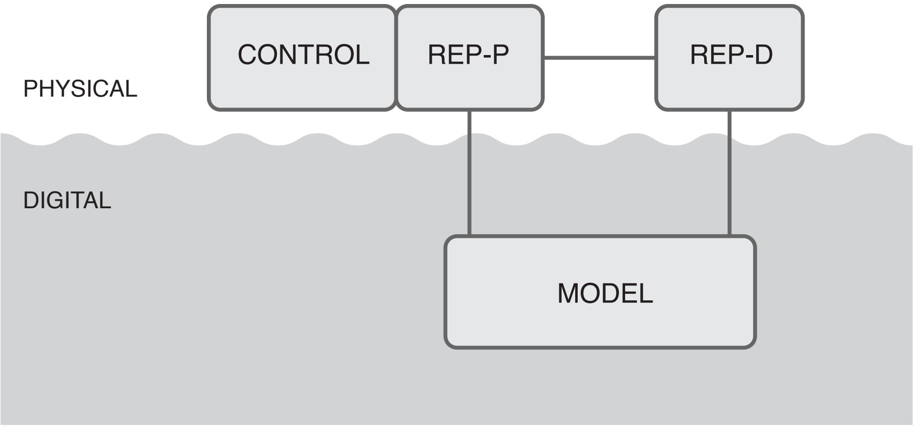


Figure 2.7: The MCRpd Interaction Model [image source: redrawn by author; based on MCRpd model in (Ullmer and Ishii, 2000)]

Adaptors perform a transform information from the physical environment to the digital world and vice versa. An accelerometer for example may be modelled as a separate device, but if integrated into smart phones they can be abstracted away as part of an interaction path.

Interaction carriers are mediating entities that are necessary for information communication. Passive carriers can carry and store part of the information communicated along an interaction path, e.g. a tangible object left in a particular position. Active carriers are transmitter of non-persistent information along the interaction path, e.g. a stylus used to transmit a precise position on a touch screen. Contextual entities are physical entities involved in an interaction (e.g. a table), and are also considered mediating entities.

The intended user model refers to what the user should know about the interaction in order to carry it out successfully. It may refer to one atomic interaction path (e.g. a channel, source and destination), or it may refer to more complex paths.

An interaction group refers to a set of entities and channels that together have properties that are relevant to a particular design issue. Some of these groups will be applicable to any design, while others will depend on the task and context:

- Entities and channels may be *grouped for feedback*, to identify an interaction flow that links the response of the system to the actions of the user.
- User interface elements may be linked to application concepts in order to express a semantic association. The goal is to help the user to cognitively unify elements of the group (helping to establish the intended user model).
- Sets of input (e.g. speech input for gesture input - "put that there") that must be combined to perform a certain task, may be grouped for multimodal interaction.
- A grouping may be used to assert that a set of services must reside on the same machine or be distributed over multiple devices.
- A grouping of paths may show information flows among or between multiple users.

Table 2.2: Interaction tasks mapped to logical and physical interaction devices

Interaction Task	Logical Device	Physical Device
Position	Locator	Tablet, touch panel, trackball, joystick, mouse
Select	Choice Pick	
Path	Stroke	
Quantify	Valuator	Dials
Text entry	String	Keyboard
Orient		

Dubois and Gray (2008) considers their interaction model similar to that of (Coutrix and Nigay, 2006), in the sense that both combine the physical and digital dimensions of the interaction.

2.4.6 Task Models

Foley defined six basic interaction tasks (BITs) that correspond to the lexical level. A BIT is the smallest unit of information entered by a user that is meaningful in the context of the application. He noted that there are far too many interaction techniques to give an exhaustive list, and that it is impossible to to anticipate which new techniques may be created. In table 2.2 we map them to possible logical and physical interaction devices. The six types of logical devices were also defined by Foley in (Foley et al., 1996).

Some characteristics of the physical interaction devices are not shown in the table. The positioning of tablets and touch panels are *absolute*, while that of trackballs, joysticks and mice are *relative*. A touch panel is considered *direct*, as the user directly points at the screen, while a tablet is *indirect*. Joysticks, tablets and mice are *continuous*, while a keyboard is *discrete*. Dials can either be *bounded* or *unbounded*.

The *positioning* interaction task involves specifying an (x,y) or (x,y,z) position. Characteristics of this task include different coordinate systems, resolution and spatial feedback. The *select* interaction task involves choosing an element from a choice set, while the *text* interaction task entails entering character strings to which the system does not assign specific meaning. The *quantify* interaction task involves specifying a numeric value between some minimum and maximum value. The *path* interaction task consists of a number of positions over a specific time or distance interval. The *orient* interaction task is also called *rotate*, but is not often used (de Ruiter, 1988).

Card et al. (1991) argued that the Foley taxonomy has not tried to define a notion of completeness, and is thus not generic enough. They pointed out that single devices appear many times in the levels of the tree, which makes it difficult to understand the similarities among devices. Mackinlay et al. (1990) extended Buxton's work to propose additional physical properties that underly most devices. They follow mappings from the raw physical transducers of an input device into the semantics of the application.

Dix et al. (2008) noted that Card et al.'s analysis are not only relevant to GUIs, as they used a running example of a radio with knobs and dials. Their work not only abstracts devices into classes, but also takes into account that rotating a dial is different from moving a slider, i.e. the physical nature of the interaction is also important.

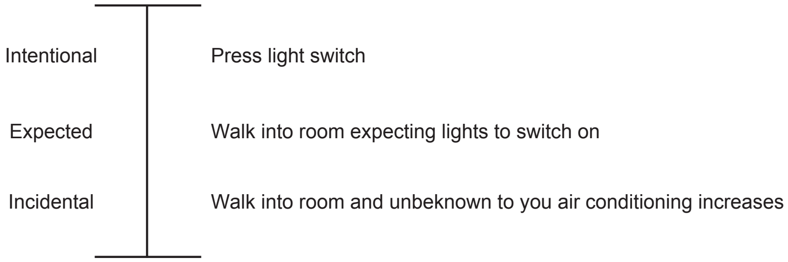


Figure 2.8: The continuum of intentionality

Ballagas et al. (2006) surveyed interaction techniques that use mobile phones as input devices to ubiquitous computing environments, and used Foley's six interaction tasks as a framework for their analysis. In their work on iStuff (Ballagas et al., 2003) they state that the set of interactions tasks are only sufficient for describing graphical user interfaces, not physical user interfaces, or user interfaces in general. The same paper notes that Buxton's taxonomy, and the extension by MacKinlay, Card and Robertson, is too narrow for ubiquitous computing environments, as it does not classify devices with different modalities and only describes input devices. They extended the taxonomy further to describe attributes like direction and modality. The direction attribute is used to indicate whether a device provides input, output or both. The modality attribute describes different visual, auditory haptic and manual modalities for input and output. Additional attributes they identified include directionality/scope (where a device is targeted to one, many, or all the users in a room) and mount time (the effort necessary to use an interaction device).

2.4.7 Models of Intentionality

At the semantic level, we are interested in the meaning of the action. A gesture may mean nothing, until it encounters for instance a light switch (Bongers and Veer, 2007). In traditional software applications, a user is expected to have a clear intention of what he/she wants to achieve, with purposeful and direct actions. In ubiquitous computing scenarios, the interactions are less explicit. Input is implicit, sensor-based and "calm", and output is ambient and non-intrusive. With *incidental interactions* (Dix et al., 2004), a user performs an action for some purpose (say opening a door to enter a room), the system senses this and incidentally uses it for some purpose of which the user is unaware (e.g. adjust the room temperature), affecting the user's future interaction with the system.

The *continuum of intentionality* in Figure 2.8 has normal, intentional interactions at the one end of the spectrum (e.g. pressing a light switch), expected interactions in the middle (e.g. walking into a room expecting the lights to go on), and incidental interactions at the other end. As users become more aware of the interactions happening around them, they move through the continuum toward more purposeful interaction. For example, with *comprehension* an incidental interaction (lights turning on when you enter the car) turns into an expected interaction. With *co-option*, an expected interaction turns into an intended interaction (e.g. deliberately opening and closing the car door to turn on the light).

Incidental interactions do not fit existing interaction models based on the conventional intentional cycle, like Norman's Action Cycle Diagram (Norman, 1998). The purpose of the user's activity is distinct to the intended outcomes of the system. Feedback may be unobtrusive (and not noticed), or delayed (like the temperature slowly changing). There are two tasks that are occurring:

- The user's purposeful activity
- The task that the incidental interaction is attempting to support/achieve

2.5 Mental Models

Whenever people see or interact with things in the world, they create a mental model of what they think will happen. Consider the simple example of a round (unrestrained) object on a slope, because we see all relevant parts involved and know the laws of physics, we can predict what will happen. The round object will roll down. Similar mechanisms are at work when users interact with products or systems. In order to understand products and systems, humans develop a conceptual model of how we believe things work and how they should be used. As long as the underlying mechanisms of the working of a product are simple and visible (reside in the physical world), they have a bigger chance to be understood, and make sense.

Mental models or conceptual models become particular important for designers of products or systems that include (micro) electronics. Norman (1998) calls these internal artefacts, in contrast to surface artefacts (what you see is all you get), of which the first needs interfaces to represent and allow control over its internals. Here, the mental model is a person's mental representation of the *internal working* of the product or system. Getting it wrong can make these products very difficult to use.

Mental models tell people how artefacts could work, when to do what and what to expect as a consequence of one's actions. Affordances and (physical) constraints are important mechanisms to invite users into actions and guide users in an artefact's possible use. Related to affordances are other conceptual handles for designing interfaces like *informatives* and *indication functions* (from theory of product language) and essentially guide and inform users about the flow of the interaction. Informatives include: signals, state indicators, progress reports, confirmings, affordings, discontinuities, correlates, maps of possibilities, error messages and instructions. Al these concepts help users understand interfaces and guide them in what to do next.

Users' mental models consist of both semantic and procedural knowledge about a product or system. Semantic knowledge is about what something is, what it can be used for. Procedural knowledge deals with the how, helping users to go through the different steps of a user interaction. (van der Veer and del Carmen Puerta Melguizo, 2003).

In a user centred design approach (as was also discussed in Section 2.6.2) it is important for designers to know (or at least try to understand) how users understand a product or system, and what the mental models look like (second-order understanding). Figure 2.9 shows Norman's (1998) interpretation of conceptual models. There are two conceptual models (which is usually the origin of a possible mismatch); the *design model* and the *user's model*. The design model is the conceptual model of the designer when he designs a product or system. A good designer will attempt to make his/her model clear through the design. However, the designer can only communicate the model through the design of the

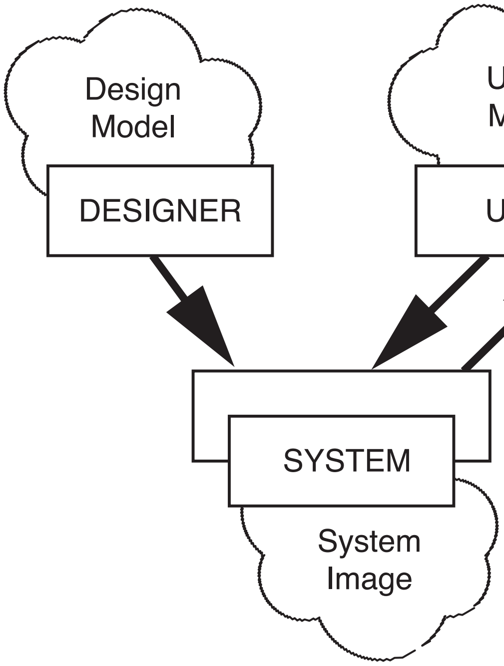


Figure 2.9: Conceptual models [image source: by author, based on depiction in (Norman, 1998, p. 16)]

system. The *system image* is what the user perceives of the system and includes the visible part of the system and its behaviour (but also documentation, instructions and labels).

With our concept of semantic connections we aim at making the invisible parts of a smart environment, the connections and relations that exist between the smart objects perceivable. Not only by having users experience the behaviour of this system, but also by enabling them to easily configure the information exchange between the smart objects and allow them to explore the connections that exist. A good measure for success would be mental models in users that are compatible with the system, i.e. the user does not have to know the technical ins-and-outs like they often know now (are the connections WiFi, Bluetooth, USB or maybe HDMI) but only be concerned with the connections at a semantic level. What the connections can do for them.

2.6 Theories of Design and Semantics

In this section several design- and semantic theories that are considered relevant to our work are discussed. Before we introduce our concept of semantic connections more closely in Part II and III, we now attempt to interpret the theories to see how they could apply to connections and relations between objects instead of the objects themselves. At some points, premature assumptions about semantic connections are made with the purpose of speculating of how the theories could be applied. We end this section with concluding remarks, that will provide a starting point for defining a preliminary Semantic Connections Framework. Each of the following sections will explain and discuss the theory at hand, and conclude with a discussion about its implications for our concept of semantic connections.

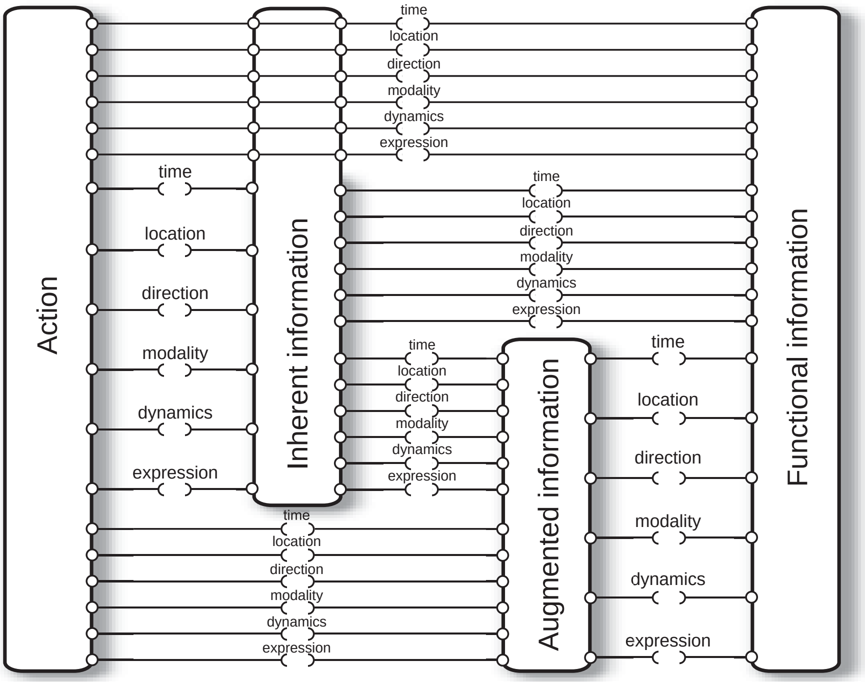


Figure 2.10: The Interaction Frogger Framework showing the different couplings between user action and the products' function [image source: (Wensveen, 2005)]

2.6.1 Direct approach—Interaction Frogger Framework

The Frogger framework, as was introduced by Wensveen (2004), describes user interaction in terms of the information a user perceives, (like feedback and feed-forward) and the nature of this information. It distinguishes between inherent, augmented and functional information. These types of information can serve as couplings between user actions and the products' functions in time, location, direction, modality, dynamics and expression (Figure 2.10). Although the framework was designed to describe the interaction with electronic devices and their interfaces, many of the concepts in the framework are applicable to our semantic connections concept as well.

When a user performs an action and the device responds with information that is directly related to the function of that product (lighting switching on when a light switch is operated), we speak of *functional feedback*. When a device has more than one functionality, functional feedback should be viewed with respect to the users' intentions and goals when performing the action. If there is no direct link between a user's action and the direct function of the product or when there is a delay, *augmented feedback* can be considered to confirm a user's action. This feedback is usually presented in the form of lights, sounds or labels. *Inherent feedback* is directly coupled (inherently) to the action itself, like the feeling of displacement, or the sound of a button that is pressed.

While feedback is information that occurs after or during the interaction, feedforward is the information provided to the user before any action has taken place. However, feedback of an action may also serve as feedforward for the next action. *Inherent feedforward* communicates what kind of action is possible, and how one is able to carry out this action. Inherent feedforward is in many terms similar to the concept of affordances, revealing the action possibilities of the product or its controls (Wensveen, 2005). When an additional source of information communicates what kind of action is possible it is considered *augmented feedforward*. *Functional feedforward* communicates the more general purpose of a product. This type of information often relies on association, metaphors and the sign function of products, which are described by theories such as product semantics (Krippendorff, 2006) and product language. Good practice in creating functional feedforward, is making the functional parts of a product visible, informing users about the functionality of the product (Norman, 1998). Other good practice is to only show feedback and feedforward in the state in which it is relevant (Frens, 2006).

2.6.1.1 Implications for Semantic Connections

If we consider semantic connections in the Interaction Frogger framework, the following interesting insights emerge:

Feedback: When we consider multiple interconnected devices and the functionalities and services they provide, information like feedback and feedforward gets spatially distributed. A user may operate a device, receiving inherent feedback locally, but receiving augmented and/or functional feedback remotely. As inherent feedback is inherent to the operational controls of the device, these reside only in the physical world and are local to the device. Augmented feedback is feedback that is augmented from the digital domain onto the physical world. This type of feedback is subject to change when devices get connected to other devices. In the domain of networked digital artefacts, functional feedback is of a digital nature. Data, media and services that exist in the digital domain become available in the physical world, through the various devices and their connections. Although many functionalities of digital devices can be regarded as (displaying) media, data or services, for some simple functionalities this seems problematic. If we, for example, look at functional lighting, it seems that the presence of light as the functionality of a lighting device is not a concept that is part of the digital domain. However, if we view a lighting device as a networked smart device, the presence of lighting, based on some sensor data, can be considered the functionality of a digital service.

But what about the semantic connections themselves, do they have these types of feedback as well? If we consider the connections to be physical entities with which one can interact, be it through an interaction device, they do provide these types of information as well. Cables are examples of physical connections (or connectors) and give users inherent information like the end points where the male part (plug) fits the female part (socket). This also provides direct inherent feedback like a snap or a click when the plug is fitted and by tracing the cable one can observe whether the connection is completed by finding out if both endpoints are connected. Inherent feedback is feedback that is inherently physical and should in case of invisible wireless connections be mediated through an interaction device, as one can not manipulate the connection directly. This inherent feedback may however be closely related to the action of making or breaking a physical connection, like a snap or click when the connection is made or broken. Augmented feedback may be in the form of lights indicating a connection or projected or displayed lines. Functional feedback is information about the actual function of the connection, like the sound from a speaker that was just connected to a media player. This type of feedback always reaches the user through the devices being connected. Figure 2.11 shows examples of these types of feedback and feedforward in how they are currently used in making connections.

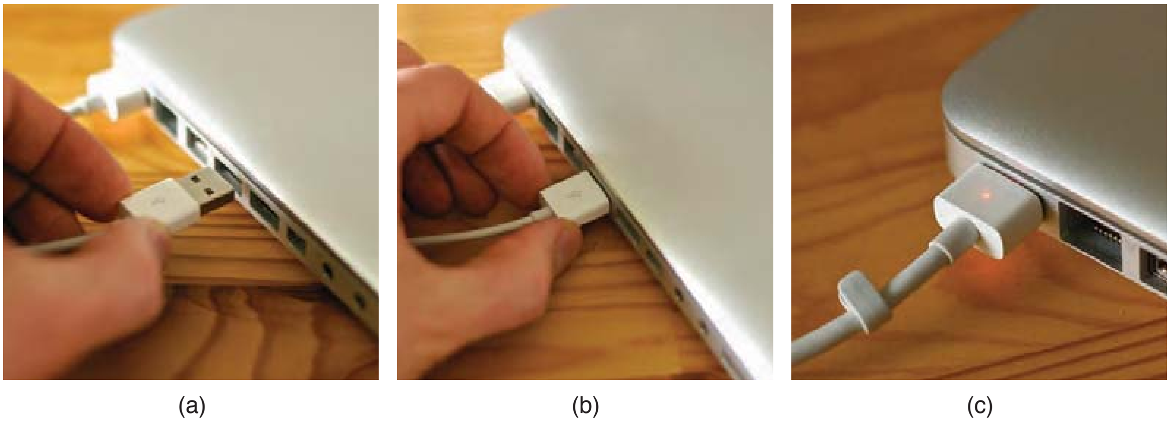

(a) (b) (c)

Figure 2.11: Examples of the different types of feedback and feedforward: (a) Inherent feedforward; visual similarities between the socket and the plug. (b) Inherent feedback; the feeling of resistance and a "snap" when the USB plug is inserted. (c) Augmented feedback; (amber coloured) light shows that the connection is working and the battery is currently charging.

Feedforward: Inherent feedforward, conceptually similar to the notion of affordances (Norman, 1998), provides information about the action possibilities with the devices or the individual controls of an interface. Similar to this are also *informatives* (Krippendorff, 2006, p. 117) and partially also *indication* or *marking functions* as defined in the theory of product language (Gros, 1983). Inherent feedforward is always physical and locally on the device. However, when devices or objects are part of a larger system, feedforward also emerges where interaction possibilities between objects exist (e.g. a key that fits a lock, a connector of one device or cable that fits another). The same holds for augmented feedforward, where lights, icons, symbols and labels provide additional information about the action possibilities. These may concern the action possibilities locally at the device, as well as action possibilities that concern the interaction with other devices in the environment.

While inherent and augmented information are primarily concerned with "the how", functional feedforward communicates "the what", the general function of the device or the function of a control. This type of information often relies on association, metaphors and the sign function of products and are described in theories such as product semantics and product language. With multifunctional digital artefacts and even more with networked artefacts this becomes increasingly difficult. Introducing the concept of semantic connections tries to address these problems, therefore the functional feedforward is the main challenge when designing semantic connections. Functional feed-forward should give information about the function of the semantic connection before the interaction takes place. Properly designing functional feedforward may be the crucial part of understanding semantic connections, smart services and smart environments.

Wensveen et al. (2004) further proposes that in interaction, these types of information can link action and function together in time, location, direction, modality, dynamics and expression. Strengthening these couplings between action and function will lead to richer and more intuitive interactions (Wensveen, 2005).

We can also view semantic connections in the Frogger framework in more general terms. Although semantic connections are not a physical device or product, but rather describe the structure or configuration of a system of devices, the Frogger framework can teach us important lessons. When we look at the link between action and functional information in time or location, a strong link would mean they coincide in time and location. For location this would mean that the connection that is made between devices corresponds to the location of the actual devices in physical space. Additionally, the direction of the action of connecting/disconnecting devices, being moving devices towards or away from each other, would strengthen the coupling in terms of direction. Also, the direction of the action could have a link to the directionality of the semantic connection that is made. Couplings in dynamics (of the action) can be used in similar ways and may express the persistence of the connection that is made. Similar to this are the couplings in dynamics.

The harmony in sensory modalities between the user's action and the connections are more problematic. The functional result of the connections can have many different sensory modalities. In a sense, the connections themselves have, or carry these modalities, from one object to another. Possibly, augmented feedback and feedforward could have similar modalities as the modalities that are carried over the connections. Expressiveness can be viewed as a more abstract property when applied to semantic connections. We might say that, when properly designed, semantic connections should give users a way to successfully express their intentions in connecting the various devices together, to accomplish a certain higher level goal. However, this differs from the original definition of the concept of expressiveness in the framework.

2.6.2 Product Semantics

The theory of product semantics describes and analyses the meaning of products in terms of what a product is and to a certain extent how it can be operated. Product semantics is a theory about how products acquire meaning. Product semantics was defined by Krippendorff and Butter in 1989 as being both:

> "A systematic inquiry into how people attribute meanings to artefacts and interact with them accordingly."

and

> "A vocabulary and methodology for designing artefacts in view of the meanings they could acquire for their users and the communities of their stakeholders."
> (Krippendorff and Butter, 1989)

Product semantics shares many concepts with information theory and semiotics, the theory of signs (Chandler, 2002). Krippendorff states in his work the Semantic Turn (Krippendorff, 2006): "Humans do not see and act on the physical qualities of things but on what they mean to them" (p. 47) and "One always acts according to the meaning of whatever one faces. [...] It always concerns sets of possibilities and presupposes human agency" (p. 58)

Krippendorff (2006) thus differentiates between the intended meaning of the designer, leading to the design, and the meaning it eventually acquires after interpretation and reinterpretation by the user during use. These two meanings are different things and the meaning that a design has for its user may be a different one than the meaning the designer intended. Krippendorff's concept of meaning is in accordance with information theory, where the designer is viewed as a communicator of a message in the form of a product and the user as a receiver of that message (Crilly et al., 2008).

Krippendorff's (2006) semantic theory has, as briefly discussed before, a very human-centred approach; as he states: "meanings are always someone's construction [...] meanings are always embodied in their beholder" (p. 56). He also argues for conceptual openness, as meaning emerges in the process of human interaction with artefacts. "Meanings are neither intrinsic to the physical or material qualities of things, nor can they be located within the human mind. [...] Meanings are constructed from previous experiences, expanded on them and drift, much like imagination does" (Krippendorff, 2006, p. 56). This human-centred approach thus implies that humans create their own meaning. They do this in a way so that the artefacts fit their own needs, applications and rituals. Consider the simple example of a chair. Although everyone will agree that a chair is to sit on, a chair may also be used by someone (small) to reach high up and thus means something else to this user. All meanings are context dependent as usually many meanings are possible, but only few of them make practical sense. Artefacts may mean different things in different contexts and may mean different things to different people. Contexts limit the number of meanings as "artefacts mean what their contexts permit" (p. 59). Contexts work in two directions in the sense that one thing provides the context for the other and vice versa. For artefacts this means that "the meaning of an artifact's parts depends on the meaning of their arrangements, just as the meaning of its arrangements depends on that of its parts" (p. 61). Krippendorff compares understanding complex artefacts with reading texts, with the (rather big) distinction that one can interact physically with an artifact, in contrast with only visually perceiving a text. Because of the human-centred approach to meaning, different stakeholders (in design) have different meanings. To a user an artefact might mean something completely different than to a manufacturer or a salesperson. Also designers may intend a completely different meaning when designing an artefact. To overcome these differences in meaning, a human-centred approach requires designers to have a second-order understanding of the artefacts they produce. This means the designer should have the capacity to understanding others' understating of artefacts. Only then can designers successfully design artefacts and begin to comprehend what their design may come to mean for their stakeholders (Krippendorff, 2006, p. 65-70).

Krippendorff (2006) speaks of four main mechanisms of how artefacts acquire meaning:

- Meaning of artefacts in use,
- Meaning of artefacts in language,
- Meaning in the lives of artefacts, and
- Meaning in an ecology of artefacts.

When considering semantic connections, both the first and the latter of these mechanisms invite a closer look (meaning that arises from language and during the lives of artefacts is difficult to predict and therefore considered less relevant at this early stage).

Meanings of artefacts in use: Norman distinguishes between surface artefacts (what you see is all you get) and internal artefacts, of which the latter needs interfaces to represent and allow control over its internals. The majority of problems with usability and the constructions of meaning occur with internal artefacts. Krippendorff (2006) describes interfaces and states that: "Humans always act so as to preserve the meaningfulness of their interfaces" (p. 84). When using a well designed interface users go through the stages of :

Figure 2.12: Example of using a metaphor to design a phone with a digital phonebook. Design by: LIsa Krohn, Forma Finlandia [image source: Finnish Industrial Design Archives — `http://www.elka.fi/fida`]

Recognition: correctly identifying what something is and what it can be used for;

Exploration: figuring out how to face something, how it works, what to do to achieve particular effects, and

Reliance: handling something so naturally that attention can be on the sensed consequences of its use.

For recognition, (product) categories, (visual) metaphors and attractiveness play an important role. By finding resemblances in form and finding closeness to ideal types of a product category, people can recognise artefacts for what they are. Artefacts deviate from ideal types in dimensions, varying within certain boundaries of dimensions that define an artefact. They may also vary in features, dispensable additions to an artefact that do not alter its identity. As an example consider a smart phone. With or without many of its features it would still be a phone as long as its core function is preserved.

When we have to recognise new artefacts we can rely on the meaning of existing artefacts by using metaphors. "The essence of a metaphor is understanding and experiencing one kind of thing in terms of another" (Lakoff and Johnson, 1980, p. 5). An example of using a metaphor in design can be found in Figure 2.12.

Central to the stage of exploration are User Conceptual Models (UCMs), which are mental models of how artefacts could work, when to do what and what to expect as a consequence of one's actions. Affordances and (physical) constraints are important mechanisms to invite users into actions and guide users in an artefact's possible use. Metonymy, parts of a whole that stand for the whole they are part of, are, in contrast to metaphors, important to increasing the efficiency of exploration rather than recognising new things. Other conceptual handles for designing interfaces are *informatives* and *semantic layering*. Informatives are similar to the concept of "indication functions" in the theory of product language (as will be discussed later) and essentially guide and inform users about the flow of the interaction. Informatives include: signals, state indicators, progress reports, confirmings, affordings, discontinuities, correlates, maps of possibilities, error messages and instructions. Applying semantic layering is being selective in when to show what elements of an interface.

Meanings in an ecology of artefacts: Looking at artefacts as a species that are part of an ecology of things, is an interesting viewpoint. There is a crucial difference between ecologies of things and biological species however, as is pointed out by Krippendorff (2006) "biological species interact on their own terms; artefacts interact on human terms" (p. 195). Technological artefacts do not know of each other but "interact with each other on account of designer's specifications and/or users' desire to connect them" (Krippendorff, 2006, p. 195). He further distinguishes between diachronic accounts and synchronic accounts to analyse ecologies of artefacts. While for a diachronic account, artefacts are being traced according to their evolutions, a synchronic account "describes the network of concurrent connections between artefacts that co-determine their use" (p. 197). Important here are:

- **causal connections**: actual physical connections;
- **family resemblances**: belonging to the same product family, part-whole relationships;
- **metaphorical connections**: carry meaning between one, more familiar species of artefacts to another species;
- **institutional liaisons**: different institutions are depending on the same species of artefacts.

"In an ecology of artefacts, *the meaning of an artefact consists of its possible interactions witch other artefacts*, both, with its own kind, but more importantly with artefacts of other species" (p. 198)

Within the context of smart environments, an increasing amount of automation and increasing interconnectedness may have a negative impact on the meaningfulness of products. Artefacts can no longer be considered in isolation, as they are part of a larger ecosystem of technologies that we interact with. Therefore, designers need to provide users with handles and clues to make them understand and enable them to be effective in such an ecosystem of technologies, to understand what is happening and allow them to be and feel in control.

2.6.2.1 Implications for semantic connections

Considering the theory of product semantics, and in particular Krippendorff's view on semantics, we can start defining what implications this has for our concept of semantic connections.

Building on Krippendorff's user-centred approach to meaning, we should be careful when indicating that a certain connection has a certain meaning. Although it might have a certain predefined functionality, what it will come to mean for its users is not entirely for the designer to control. By taking a second-order viewpoint, and using principles such as metaphor, affordances and informatives to support the phases of recognition, exploration and reliance, designers can, however, provide circumstances that increase the probability of the intended meaning to come across. For semantic connections this might mean that we have to look for reliable metaphors like physical cables and the interactions with them.

Physical constraints and informatives like signals, state indicators affordances or discontinuities (in form) might help to indicate where and how to act; how to make or break connections and what devices allow and which do not allow to be connected. Additionally, the notion of causal connections that link artefacts together, like wired or wireless networks, that are known and understood, provide helpful clues. Also the notion of family resemblances, portable media players, stereo sets and speakers belonging to the same product families, might provide practical understanding of what a connection, connecting products of this family (with music playing capabilities) together might mean and what the emerging functionally will be.

Looking from the perspective of ecology, the following should be considered:

- The meaning of a semantic connection depends on the meaning of the artefacts it connects.
- Semantic connections work in mutual cooperation. They depend on other species (smart objects) and also support them.
- Semantic connections might also have *competitive* interactions with other artefacts. Emergent functionalities through interoperability between artefacts could eventually lead to less objects around us. By combining the functionality of several artefacts, others might become obsolete (e.g. combining a printer and a scanner gives copying functionality).
- Semantic connections may also have a cooperative relationship with other artefacts, because more smart objects might result in more semantic connections being made.

2.6.3 Product language

The theory of product language, also referred to as the Offenbach approach (Gros, Fischer, Bürdek and Steffen) regards design as a language of designed products. Products are considered to have practical functions and communicative functions on several levels, that users may (or may not) interpret. The communicative functions or *product language functions* are subdivided in (analogous to the distinction between syntax and semantics in language) *formal aesthetic functions*—those aspects that can be observed independent of content or meaning—and *semantic functions* (also emblematic functions, (Gros, 1983)). *Practical functions* are those functions of a product that have a direct practical result. Formal aesthetic functions can be considered to form the grammar or syntax of a design. It describes concepts from Gestalt psychology (see also Section 2.6.5) like order and complexity (Steffen, 2000). Semantic functions are considered the bearers of meaning and can again be subdivided in *symbolic functions* and *indication functions*. Indication functions (marking functions in (Gros, 1983)) are closely related to the properties of a product and communicate on two levels: indication of the nature of a product and indication of its practical functions. The first allows a product to be recognised for what it is, or the product category to be recognised. This is similar to the concept of recognition and the product functions that allow for it, as described by Krippendorff (2006) (see Section 2.6.2). Secondly, indication functions indicate the practical functions of a product and how it can be used. It indicates e.g. orientation of a product, idle-function, stability, changeability/configurability, operation, precision and reference to the human body (Bürdek, 2005). Similar to the concept of affordances, these indications come naturally with well designed mechanical products, but when a product's functioning is based on microelectronics, the product's form becomes a user interface. *Symbolic functions* are those functions that have an associative or symbolic meaning for users, e.g. style functions and brand (image). They are merely in a user's mind and are based in culture and convention. Examples are the perceived status a product might bring its user and the (social) values it communicates but also its context of use, like sports apparel and equipment, and its specific "sporty" style.

2.6.3.1 Implications for semantic connections

For our semantic connections, indication functions are considered to be interesting in particular. These functions consist of communicating what a product is, and how it can be used. They can be considered equivalent to "informatives" as Krippendorff (2006) defines them, although indication functions can include other concepts like metaphors and metonymy as well. Interesting are indication functions of networked artefacts. How can users see that these artefacts have networking capabilities and how do users recognise what kind of networking capabilities these are. For semantic connections, indication functions are important to afford the function of the connections and how they can be used, i.e. how connections can be created and what these connections mean. From Product language literature, we observe that indication functions are mainly concerned with physical products. To be able to use the same terminology to specify and describe semantic connections, we need to objectify the semantic connections. This prescribes that there will be physical objects that mediate or represent the semantic connections. These objects can have the physical attributes that serve to indicate what the objects are and how they can be used. However, when we consider indication functions in terms of communicating product category, we run into difficulties. How can this type of smart space configuration device be indicated, as there is no such product category yet. However, users are familiar with the concepts of physical wires, wireless networks and related products.

For indicating how semantic connections can be used, we can add physical features to the physical representations of the semantic connections or the interaction device that will give physical control over them.

2.6.4 Semantic connections from a semiotic viewpoint

Semiotics is the theory of signs. It is subdivided into three branches: syntactics, semantics and pragmatics. When applied to design, semiotic theories usually regard the meaning of artefacts as signs and their referents—the things that are referred to—or signifier and signified (using Saussure's terminology). As in language, this distinction between sign and referent results in the idea that a sign, which in language might be a word, refers to something else, usually a physical entity. Charles S. Pierce (Chandler, 2002) distinguishes between signs based on the nature of the relationship between sign and referent, and divides them into categories of symbols, icons or indices. Note that these categories are not mutually exclusive, as signs can belong to more that one category. Symbols have no natural relationship with what they refer to (Figure 2.14). They are arbitrary, based on convention and need to be learned. Icons are signs that have a similarity in form with what they refer to (Figure 2.13). In indices there is a casual or physical relationship between sign and referent, e.g. smoke refers to fire and footprints indicate someone has been there.

When we place semantic connections in a semiotic framework, we can say that a semantic connection itself is a sign for something else, an associative link between two (or

Figure 2.13: The old iTunes icon using a combination of iconic and symbolic representation. The icon of a Compact Disc stands for the musical content of the application.

Figure 2.14: The new iTunes icon only uses symbolic representation. The musical note is a symbol as it has no natural relationship with what stands for, and needs to be learnt. For the new icon it seems that the designers wanted to put more emphasis on digitally stored music by leaving out the Compact Disc.

more) artefacts. Semantic connections cannot be observed directly, they are always mediated. The meaning of the semantic connection depends on what things it connects and the commonalities of the meanings of these artefacts. For example, a connection between a music player and a speaker set might mean a music connection to its user because of the resemblances in semantics of the connected devices. The process of sign production might be summarised as an interpretation of something referring to something else. In the case of semantic connections, this something else is a functionality that exists because of this connection. We can investigate how to use symbolic and iconic meaning to convey the properties of the connections. Designing semantic connections using a semiotic approach will mostly rely on some form of visual language—including the visual perception of form—communicating for what purpose and how a connection can be used. We now consider symbols, icons and indices and how they can be used for the design of semantic connections.

2.6.4.1 Symbols

In the tradition of GUIs a visual language for exploring and manipulating connections between devices seems attractive. Relying on convention, solid or dotted lines, arrows e.g. can be used to describe connections. Using displays or projection, these connections can be projected in the real world, conveying practical information. For labels we might rely on symbols that are known already, or we can create new ones. When creating new signs we are dealing with the process of introducing new relationships in the collective memory of people, a process which Umberto Eco (1977) refers to as first usage (ratio difficilis) as opposed to second usage (ratio facilis).

2.6.4.2 Icons

Iconic representations might be easier to understand and should be considered when communicating the type of contents of the connections or subjects of the relationships. They could also be used to indicate where one can act.

2.6.4.3 Indices

Very direct links between form and function as well as form and manufacture can be viewed as indices, as the artefact is the natural effect of a functional design or production process. There is some controversy however on the use of the Pierceian conception of index in design. e.g. (Guldberg, 2010). Additionally, it is not certain what signs are relevant for a user or what signs they might interpret. A user might not recognise a chair as the result or the hard work of a carpenter, but just as his/her grandma's favourite chair. For our semantic connections, we can regard physical cables between two objects as a kind of index, the physical cause of two objects being connected. But perhaps also semantic connections in general or their results can be viewed as indices, as there is a causal relationship between the perceivable result and the connection. Analogous to smoke being an index of fire, the sounds of music are an index of a producer of sound waves, which may be a speaker. Similarly, the sound of this speaker is coming from a sound source, which may be the result of a connection between a media player and the speaker.

2.6.5 Gestalt, colour and form

Form, and especially colour can also have a predominantly signalling function, without relying on symbolic, iconic or indexical meaning per se. In biosemiotics they address non-verbal signs, which are not interpreted by human intellect but rather from within the body. This is similar to the perspective taken in the ecological approach to visual perception as described by Gibson. Colours can, even if they are not part of a codified system, produce behaviour by attracting or rejecting our gaze (Zingale, 2010). Therefore, without relying on a symbolic meaning, colour (and/or form) can be used to invite users to certain behaviour. When navigating through a smart space, colours could be used to attract attention to points of interaction or could—when using different colours—differentiate between different types of connections, without relying on symbolic meaning. Matching colours and the form of arbitrary, meaningless symbols (Gestalt Law of Similarity) may be used for for matching connection possibilities (figure 2.15). Additionally, certain abstract properties of form may be used to suggest a path, direction and the like, without requiring users to learn a certain sign language.

We can also use some of the concepts of Gestalt psychology. Particularly interesting for connections are the laws of Prägnanz or grouping, the main principle in Gestalt psychology. Gestalt psychology revolves around the principle that the human mind is holistic and that it has self-organizing tendencies in its perception (Rock and Palmer, 1990). The laws of grouping (Figure 2.16) are a set of hypothetical laws that allow for prediction as to how visual information is grouped according to certain characteristics.

The Law of Similarity: elements that are similar in physical attributes tend to be grouped

The Law of Proximity: elements that are close together in space or time tend to be grouped

The Law of Continuity: elements that appear to follow in the same direction tend to be grouped

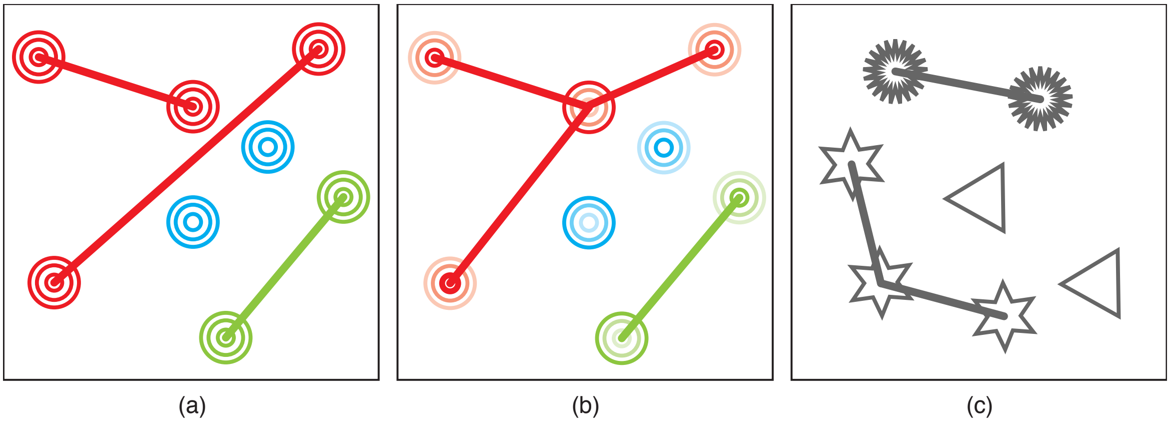

(a) (b) (c)

Figure 2.15: Examples of using colour and form without relying on the symbolic meaning of colour per se: (a) using matching colour to indicate connection possibilities. (b) adding a bit more semantics by giving an impression of directionality in an abstract manner. (c) using matching random shapes for indicating connection possibilities (Gestalt law of Similarity).

The Law of Closure: where we deliberately want to perceive the "open" form or image as a "closed" and completed shape or form

The Law of Common Fate: elements that appear to move together tend to be grouped

The Law of Symmetry: elements that form symmetrical and enclosed objects tend to be grouped

Interesting for semantic connections is the Law of Closure: The mind has a tendency to complete incomplete forms, effectively seeing something for which it does not receive stimuli, and could potentially be used to visualize something that is invisible (like a wireless connection). The law of Proximity, objects or artefacts that are physically close tend to be grouped together and may be used to indicate a connection between objects that are close together. The law of Continuity may be used as objects that are part of a network and seem to be on one line tend to be grouped together.

2.6.6 Ecological Perception

Although the theory of ecological perception and the concept of affordance has been briefly discussed in some of the previous sections, we would like to discuss the theory and its implications a bit further. While many of the semantic theories discussed depart from a semiotic/linguistic and communication perspective, the ecological approach to perception has an entirely different theoretical foundation. Despite these differences, it will also become clear that on a practical level, the resulting designs might rely on similar perceptual qualities.

Affordance, which is a central concept of ecological perception theory, is an object's quality that appeals to our sensory-motor skills, like a door-handle that "affords" to be grabbed and a chair that "affords" to be sat upon. When the insights of ecological perception were introduced into design by Norman (1998), it fuelled the design community to try and solve many usability problems. Whereas on a practical and application level not necessarily relevant, Norman's view of affordances is slightly different from the original thoughts of Gibson and many like-minded psychologists (Norman, 1998, p. 219). Central to the

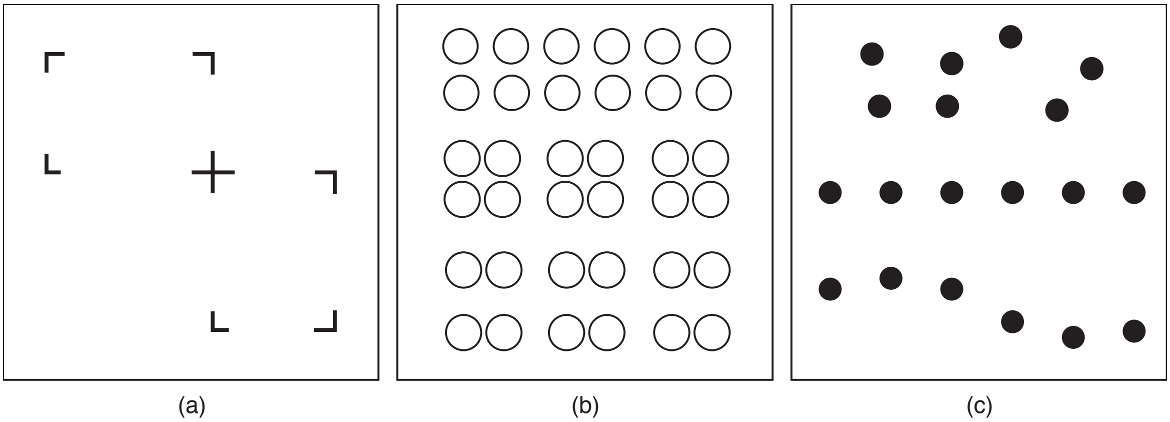

(a) (b) (c)

Figure 2.16: Examples of using Gestalt laws of Grouping: (a) Law of Closure; the human mind will create two rectangles, while they are not there. (b) Law of Proximity; by using proximity between objects we group objects close together. (c) Law of Continuation; objects that follow the same (line) direction are grouped.

notion of affordances is the inseparability of humans and their environments, as humans have always dealt with their environments going through evolution. Affordances can thus neither be seen independently from humans, nor can they be viewed independently from the environment. For affordances to be detected, they need to be available as information that can be perceived by the human perceptive system. Secondly, they will be viewed in relation to the bodily properties of every individual. While chairs may afford seating for adults, it may afford something else for children that might play underneath it. Conversely, perception requires actors (Djajadiningrat, 1998).

Although we can give physical controls and simple products affordances, the resulting design of such objects by taking either an "affordance driven" or a "product semantics/language driven" approach, may often be similar. On a practical level, visualising the functional parts, designing indication functions proportional to the human body and other good practices of design might lead to similar results as an "affordance driven" approach. Furthermore, when designing complex products and interfaces, affordances often work well for inviting users to perform certain actions which the controls allow for. This does not necessarily indicate what the results of such an action will be. This is acknowledged by Djajadiningrat (1998); however, he also successfully shows that the notion of affordances can be used as a framework for design.

2.6.6.1 Implications for semantic connections

Because connections/relationships between networked artefacts are not always direct and physical, and perhaps only mental constructions, affordances are a difficult concept in this context. We can create affordances for the control over these connections, but they will most likely only reveal how to manipulate the connections/relations and not be very informative about the nature of these connections. Here, associations and meanings of the artefacts and their capabilities are important, which are learnt and rooted in convention and previous encounters with products. However, affordances can be used to invite users to perform certain actions, and these actions can carry meaning. To give a few examples of possibilities, the affordance of a control to make a connection can be shaped in such a way that it invites to an action that may associate with permanent or non-permanent connections, like a locking action after inserting a connector into a socket. Furthermore, there can be the affordance that invites the movement of a control in a certain direction (e.g. a sliding switch). This direction may in turn translate into the directionality of a connection. Some of these ideas have been implemented in the design of a digital camera (Frens, 2006) and a VCR controller in (Djajadiningrat, 1998).

2.6.7 Discussion

In this section we have discussed various theories of design and sense making. Much of the design theory described is, however, about the meaning of objects (or sometimes language) and originates from the era of non-interactive, mechanical and electric products and machines. With the introduction of microelectronics and digital electronics, many of these theories have been reconsidered to accommodate for *interfaces* and interactivity, and some have evolved into new ones. Now that we have entered the era of digital networked artefacts, which introduces additional concepts and complexity, these theories may need to be reconsidered; especially when considering that the networking technologies that connect these devices are wireless and thus invisible.

Today networked objects are often recognised by their LCD screens, as part of a product category of “smart objects”, or desktop, portable or wearable computers. Developing a form language and interaction paradigms for such products is a challenge that a large part of the (interaction) design community is and has been working on. Despite these efforts, today’s products remain mainly GUI-based and these GUI’s are the most important means for controlling connectivity.

The *semantic connections* concept and underlying theory proposes to reveal the many invisible connections and allow direct physical control over them, like we have control over many physical wired connections. To support this, part of our solution also proposes a software architecture to solve the current interoperability problems to a certain extent. This software architecture enables networked devices to exchange information and share device capabilities. Together this is expected to enable users to interact with the various devices in the system on a higher, more goal-oriented level, moving away from the current device-oriented way of interaction.

2.6.8 Conclusions

Having discussed some of the important theories about how designed objects acquire meaning, it is clear that, although starting from different viewpoints and theoretical backgrounds, they have many commonalities. The design community has developed ways to describe and address many usability problems by communicating the relevant action possibilities to invite for the right actions. If well designed, handles for communicating such action possibility can be provided by affordances (ecological perception theory), informatives (product semantics) and indication functions (product language). With these frameworks, designers are still challenged when they need to communicate the purpose or possible result of an action before it takes place (Wensveen et al., 2004). The actions that a user has to perform to complete a task are not the goal of the user, but are the means to reach the goal which is fulfilling the task. Norman (Norman, 1998) describes the stages a users typically follows from defining a goal to task completion in his “7 stages of action”.

Before users can understand the result of an action, there are different stages of sense making which they have to go through first. First of all, users should be able to recognise an artefact and perhaps more importantly its functioning. To realise this, designers can rely on using metaphors and making the functional parts of a product visible. After establishing what the device is and how it might function, there still remains the challenge for designers of communicating the results of interacting with the identified interactive parts and controls. For simple single-function products this sometimes follows naturally from the design, but with many of today's multifunctional products this has become more of a challenge.

Wensveen et. al. (Wensveen et al., 2004) propose an approach they refer to as the *direct approach*, which departs from the idea that not only the physical appearance of a product, but also the actions it invites users to perform, are carriers of meaning. They argue for a strong link between the qualities of an action and the result of that action, as is described in the Frogger framework (Wensveen, 2005). The notion of feedforward is pivotal in this direct approach, especially functional feedforward (as described in section 2.6.1). For our notion of semantic connections, we rely on several mechanisms to provide this (functional) feedforward.

First of all we rely on *natural mappings* (Norman, 1998). The connections or associative links are created between devices, places, persons or interactive parts of devices, that all exist in the physical reality. Instead of relying on identifying networked devices by name or other types of representation, we identify them by their physical location, where users can perceive them, point at them and touch them. Secondly, we rely on the meaning of the devices that are being connected, in particular the *resemblances in meaning* of the devices being connected. Important here is the change in meaning that might occur, when users view the device no longer in isolation, but as part of a larger system. Krippendorff (Krippendorff, 2006) discusses these part-whole relationships (as is described in section 2.6.2). Thirdly we rely on feedback and feedforward being provided by a mediating device or service, which has the special purpose to enable exploring and manipulating the—otherwise invisible—connections. We not only consider *which* things are connected, but also *how* these connections are made. This is where we have the freedom to carefully craft the way we discover and manipulate these connections, to provide additional information about what the connection will mean once it is made. Once the connection is active, in many cases the functional result in the physical environment will give additional feedback on the success and functionality of the connection that was made.

Part II

Design Explorations

The second part of the thesis describes a series of designs to explore and evaluate various ways of interacting with otherwise invisible connections. From these explorations we draw conclusions which are then used to define a theory of *semantic connections* and a design framework, to be described in the final part of this thesis.

Chapter 3

Design Exploration I

3.1 In this chapter

To get a better understanding of what semantic connections could be like, a first, exploratory iteration was performed with the primary aim to go full-circle. This included a preliminary definition of a semantic connection (3.2), a physical interface to these "invisible" connections, a series of *smart objects* and their digital counterparts (KPs, see also Section 2.3.4), a software infrastructure to create semantic interoperability and a simple use case scenario. This chapter describes this initial design exploration from a design perspective, which is described in Section 3.3. Based on this, a more robust prototype was created which served as a demonstrator and was used in a series of user experiments. Two Industrial Design master students then used our work to design alternative interfaces to semantic connections, exploring the influences of the different designs on the mental models that users create when interacting with them. Their work is described in Sections 3.4 and 3.5. An evaluation of the designs (Section 3.6) and their implications (Section 3.7) conclude this chapter.

Parts of this chapter were previously published in:

Vlist, B.J.J. van der, Niezen, G., Hu, J., & Feijs, L.M.G. (2010). *Design semantics of connections in a smart home environment.* In L.- L. Chen, T. Djajadiningrat, L. Feijs, S. Kyffin, D. Steffen, & B. Young (Eds.), Proceedings of design and semantics of form and movement (DeSForM) 2010 (p. 48–56). Lucerne, Switzerland: Koninklijke Philips Electronics N.V.

Vlist, B.J.J. van der, Niezen, G., Hu, J., & Feijs, L.M.G. (2010). *Semantic connections: Exploring and manipulating connections in smart spaces.* In Computers and Communications (ISCC), 2010 IEEE Symposium on (pp. 1–4). Riccione, Italy: IEEE.

Peeters, J. Vlist, B. van der, Niezen, G., Hu, J., & Feijs, L. (2012). A Study on a Tangible Interaction Approach to Managing Wireless Connections in a Smart Home Environment. In L.- L. Chen, T. Djajadiningrat, L. Feijs, S. Fraser, S. Kyffin, & D. Steffen (Eds.) *Proceedings of Design & Semantics of Form & Movement (DeSForM) 2012.* pages 187–196. Wellington, New Zealand: Koninklijke Philips Electronics N.V.

Kwak, M., Niezen, G., Vlist, B.J.J. van der, Hu, J., & Feijs, L.M.G. (2011). *Tangible interfaces to digital connections, centralized versus decentralized.* In Z. Pan, A. Cheok, W. Mueller, and X. Yang (Eds.), Transactions on edutainment V (Vol. 6530, p. 132–146). Springer Berlin / Heidelberg.

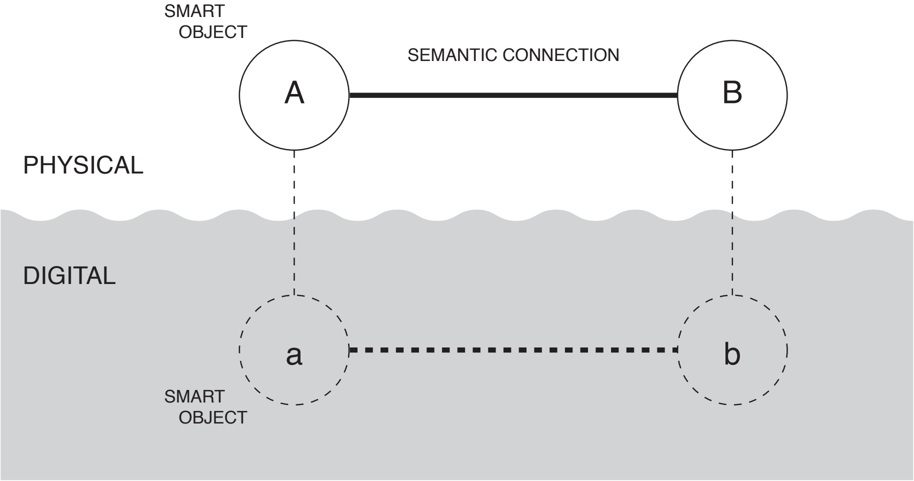


Figure 3.1: A semantic connection between two smart objects

3.2 Semantic Connections

At the time of the first design exploration, our understanding of a semantic connection was the following:

> A semantic connection is a connection between two entities in a smart space. This connection is to be considered a more meaningful connection from a user's perspective as it does not concern the type of connection in terms of the connectivity technology (e.g. Wi-Fi, Bluetooth), but rather focusses on the enabled functionality emerging from the connection. More precisely, the semantic connection that users perceive in the *physical world*, stands for a connection, technically achieved by connectivity technologies and protocols at the lower level, in the *digital domain* (Figure 3.1). In this way semantic connections serve as abstracted models of connections between devices in a smart home environment.

By introducing semantic connections, we aim to enable users to explore and make configurations on a high semantic level without bothering them with low-level details of connectivity technologies such as finding out how the devices can connect (i.e. what technology to use) and installing drivers or selecting the correct settings. This way, users are instead concerned with the additional services that come into being by interconnecting several separate devices. Enabling users to actively explore connections and connection possibilities is expected to contribute to the understanding and transparency of the smart environments and their services.

3.3 Going full circle: A Simple use case

The point of departure for the first design exploration was a use-case scenario of an entertainment setting in a smart home environment. The scenario is simple, yet interesting enough to serve as test case.

3.3.1 The scenario

> *"Mark is relaxing at home when his friend Dries arrives. Dries comes with a portable music player loaded with his favourite songs. He wants to play some of his recent collections for Mark. Mark's home is equipped with a sophisticated surround sound system. They decide to enjoy the music from the music player on the sound system. Dries starts streaming his music to the environment. An object (or several objects) shows possible input and output ports for streaming music in the environment. By interaction with the object/objects, Mark connects the output from Dries music stream to the input of the sound system. Now the room is full with Dries's music and they both enjoy listening to it.*
>
> *The object(s) shows the connection possibilities with a high level semantic abstraction, hiding the complexity of wired or wireless network. By interacting with the object(s), semantic connections can be built, redirected, cut or bypassed.*
>
> *Recently Mark has installed an ambient lighting system that can be connected to the sound system and renders the mood of the music by dynamic colour lighting in the room. Mark uses the objects again to create another connection and now the room is filled with Dries's music and colourful lighting effects.*
>
> *Mark's room-mate Sofia comes back from work and decides she wants to watch a movie on the TV. She seems somewhat annoyed by the loud music. Mark and Dries do not want to bother her and they again use the objects to re-arrange the music stream. Now the music is streamed to Mark's portable music player while also playing back at Dries's. It is also connected to the ambient lighting system directly, bypassing the sound system. They both are enjoying the same music using their own favourite earphones (and the colourful lighting effects), but without loud music in the environment. Now Sofia can enjoy her movie without any disturbing music."*

From this scenario we can see that there are multiple ways and different levels of interacting with the smart devices in the environment. There are high-level semantic interactions with the interaction object(s) (explore/make/break connections) and also lower-level interactions with the music player (play/pause/stop music).

3.3.2 Design challenge

During the first design iteration, we developed an ontology, a software infrastructure[1], software for the various Smart Objects, and a tangible interface for visualising and controlling semantic connections. For this iteration the following requirements were set:

- Semantic connections exist in both the physical and the digital domain. We need ways to visualise these invisible connections and to control them;
- Smart objects need to be able to share their capabilities and content with the other smart objects in their environment.

---

[1] At first, the Jena Semantic Web framework was used. When the first stable version of Smart M3 (Section 2.3.4) became available, we updated our software to work with Smart M3

The design challenge set out, was to design an interface to intangible connections between the devices mentioned in the scenario. To emphasise on physicality, we decided to take a tangibility approach, creating a tangible interface to intangible connections. Moreover, a tangibility approach was preferred (i.e. instead of a GUI) for what we potentially can learn from applying such an approach (e.g. in terms of feedback and feedforward). The tangible interface to be designed should support the following interactions:

- viewing/exploring existing connections;
- viewing/exploring connection possibilities;
- making connections;
- breaking connections.

Important to note, the aim to go full-circle asks for a design that is feasible to implement and evaluate. The design cycle should be relatively quick, and implementing a working prototype is key to better understand the design challenge at hand. Because we are dealing with interconnected smart objects that show behaviour defined by the implementation decisions, we follow a "sketchy" approach, aimed at experiencing the implications of our (design) decisions quickly. Similar to the conception of 4D sketching (Kyffin et al., 2005), to allow for experiencing the behaviour of designs of objects as opposed to 2D and 3D sketches, we have to find an approach that is suited for designing interoperating objects that may show emergent behaviour.

3.3.3 Related work

Over a decade of research has resulted in many proposals for configuring ubiquitous computing infrastructure and solving the interconnection and interoperability issues with consumer electronics. Early work by Siio et al. (1999) explored interactions with connected devices by moving a stylus along paths on a printed map of the infrastructure that is annotated with barcodes. More recent related work presents solutions for simplifying configuration tasks of in-home networks by creating virtual "wires" between physical objects like memory cards (Ayatsuka and Rekimoto, 2005) that can interconnect devices.

Others propose to introduce tags, tokens and containers (Ullmer et al., 1998; Want et al., 1999; Holmquist et al., 1999) for tangible information exchange. Concepts like "pick-and-drop" (Rekimoto, 1997) and "select-and-point" (Lee et al., 2008) are used to manage connections and data exchange between computers and networked devices. The introduction of near field communication, i.e. using a near field channel like radio-frequency identification or infra-red communication, allows for direct manipulation of wireless network connections by means of *proximal interactions* (Rekimoto et al., 2003). Related work is not limited to the network configuration tasks themselves. Visual metaphors to show the progress of making short distance wireless connections, such as Bluetooth pairing, and the affordances and aesthetics of making connections by physical contact between devices are investigated in (Woo and Lim, 2009).

3.3.4 Interaction Tile

The notion of semantic connections described in Section 3.2, and the design challenge as was laid out, resulted in a demonstrator which we called the *Interaction Tile*. Figure 3.2

shows an impression of the process that led to the Interaction Tile design. The Interaction Tile comprises a tile-like interactive object that allows for both exploration of a smart space in terms of connections, and manipulation of these connections and information/data streams through direct manipulation. This is done by making simple spatial arrangements. The Interaction Tile visualises the various connections by enabling users to explore which objects are connected to one another and what can be connected to what. Coloured LED lighting and light dynamics visualize the connections and connection possibilities between the devices, by means of putting devices close to one of the four sides of the tile—a user can check whether there is a connection and if not, whether a connection is possible. By simply picking up the tile and shaking it, a user can make or break the connection between the devices present at the Interaction Tile. A video of the demonstrator is available[2]. Figure 3.3 shows an use case example of how the Interaction Tile can be used.

3.3.5 Design semantics

The design semantics of the demonstrator are intended to be simple and straightforward. The tile-shape shows clear clues about orientation, e.g. what side should be placed up. The four sides clearly show four possibilities for placing objects near the tile; the size of each side restricts the number of objects one can place close to the tile. When an object is placed next to the tile, the light gives immediate feedback when the object is recognized (Figure 3.4c). When multiple objects are placed near the Interaction Tile, it will immediately show the connection possibilities (feed forward) by lighting colour and dynamics. The lights' colour coding is simple and straightforward. Red colour means no connection and no connection possibility (Figure 3.4d); green colour means there is an existing connection between the devices present (Figure 3.4a/e) and green pulsing means that a connection is possible (Figure 3.4b). To indicate that the first object a user places next to the Interaction Tile is recognised, a red colour at the side the object was detected is shown (Figure 3.4c). Placing a second, third and fourth object, the Interaction Tile shows the lighting effect corresponding to their connection capabilities. By simply picking up the tile, and shaking it, the user can make or break the connection between the devices present at the Interaction Tile. The result of this action depends on the connection's current state, and the devices present; if the tile shows a connection possibility, the action will result in a connection event. The same action performed when the tile shows an existing connection will break the connection. Feedback on what action was performed is also available, making a connection is confirmed with a audible beep, whereas breaking a connection is confirmed with a haptic vibration. The shaking action was loosely inspired on a "magic" shake to emphasise the idea that the Interaction Tile makes the connections for you. The result of the action can be checked by users by re-aligning the objects with the Interaction Tile.

We aim to enable users to explore and manipulate the connections within the smart space without having to bother with the lower-level complexity of the actual system's architecture. We envision this "user view" to be a simplified view (model) of the actual architecture of the smart space. Conceptually, the connections are carriers of information; in this case they carry music. Depending on the devices' capabilities (e.g. audio/video input and/or output) and their compatibility (input to output, but no output to output), the Interaction Tile will show the connection possibilities. In our current demonstrator we do not distinguish

---

[2] `http://www.youtube.com/watch?v=vdZcjqfq8RQ`

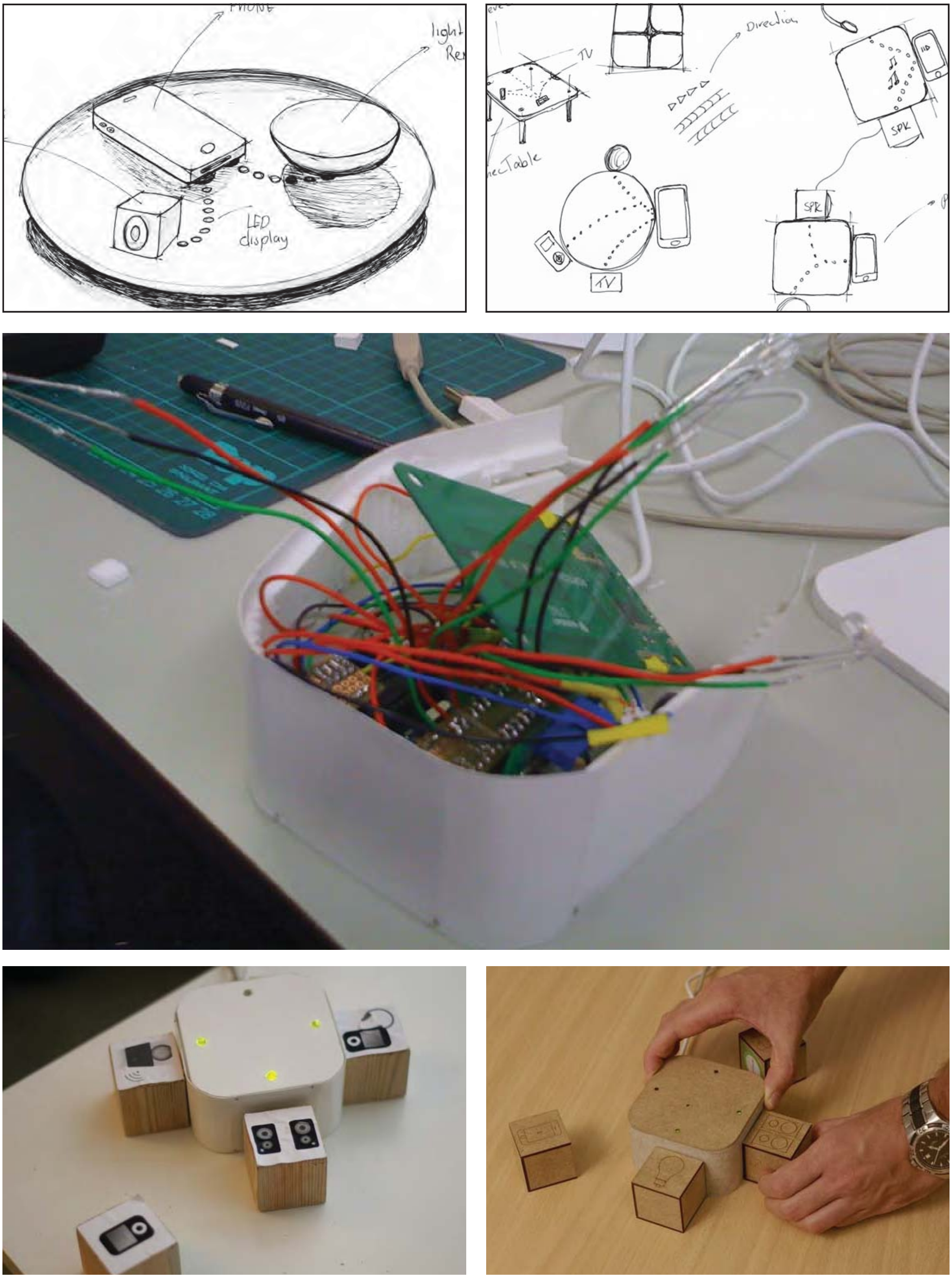
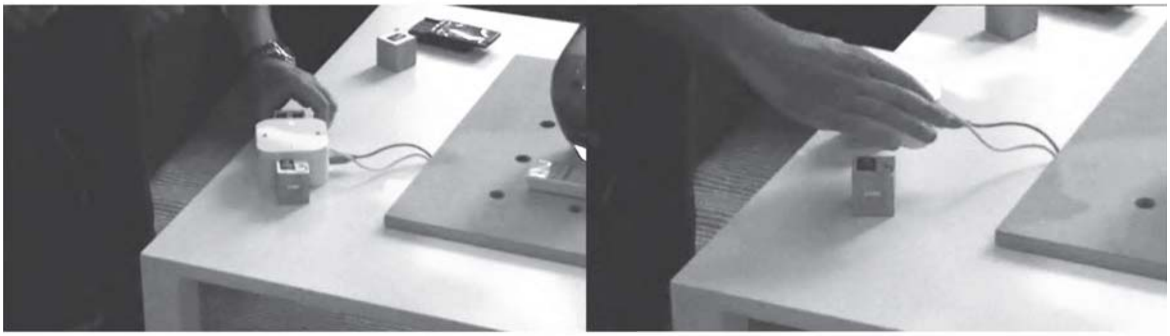


Figure 3.2: Impression of the design process that led to the Interaction Tile design: (top) early sketches and ideas of visualizing connections between smart objects; (centre) building a first functional prototype; (bottom) first functional prototype in action and the final prototype as it was used for the user experiments.

Dries visits Mark with his music player...

...and wants to play some of his music on Mark's sound system



they explore connection possibillities between music player and sound system

they make the connection

they also want to connect the ambient lighting system...

...and explore whether the ambient lighting system can be connected as well

they make the connection...

...and enjoy the music and lighting effects

Figure 3.3: The demonstrator in action

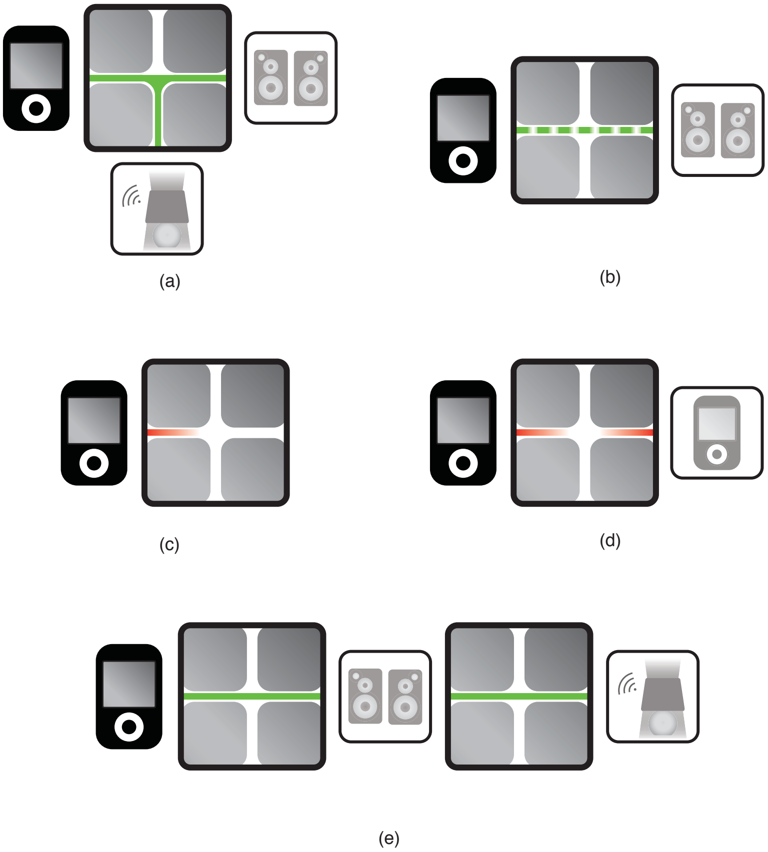


Figure 3.4: Meanings of lighting colour and dynamics: (a) Green solid light means the devices present are connected; (b) Green, pulsing light means the devices are currently not connected, but can be connected; (c) Red solid light means the device was recognised, a second device is necessary to show connections or connection possibilities; (d) Red solid light means the devices are recognised, but no connections or connection possibilities exist; (e) Shows the possibility to use multiple Interaction Tiles to look into connections in a more detailed manner, however both (a) and (e) have the same network structure.

between different types of data since we are only dealing with audio, but that may be necessary in more complex scenarios.

We rely on the symbolic meaning of colour, green colour meaning "proceed" and red meaning the opposite. Using the association of solid colour and pulsing colour with solid and dashed lines we aim at referring to the "existence" of something and the "possibility" of something. This something is a connection, being invisible but with noticeable results (the sound of music out of a loudspeaker that you just connected to your MP3 player). We rely on iconic representation for the cube-like objects representing a stationary non-mobile device and on meaning resulting from direct manipulation of these objects we just described, representing other objects. People seem to be able to work with all these different (in fact rather complex) relationships at the same time, and our expectation is that we need the richness of all these mechanisms to successfully interact with our complex environments and the envisioned smart environments of the future.

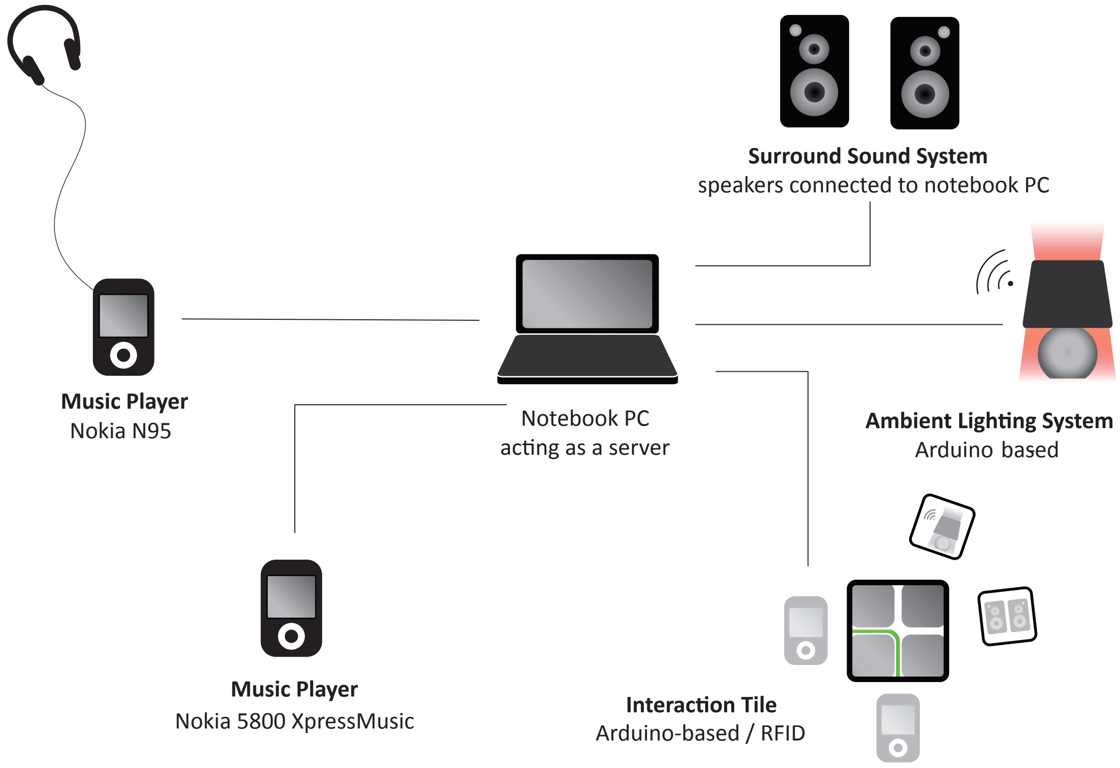


Figure 3.5: An overview of the demonstrator

3.3.6 Implementation

The Interaction Tile acts as an independent entity, inserting events and data into a triple store and querying when it needs information. It hosts an Arduino [3] (micro controller) board and a RFID reader that connect to a PC through USB. The different types of events and the connections between smart objects and their related properties are described in an ontology. The ontology with "is-a" relationships indicated is shown in Figure 3.6. Figure 3.5 shows the architecture of the demonstrator setup. A more detailed description of the Interaction Tile prototype is available in Appendix A

The Interaction Tile consists of the following components:

- Arduino board (Duemilanove);
- 13.56MHz RFID reader (ACS/MiFare) with custom antenna;
- multi-colour LED's;
- accelerometer;
- vibration motor;
- piezoelectric speaker;
- magnetic switches.

The demonstrator consists of the following devices:

---

[3] http://www.arduino.cc

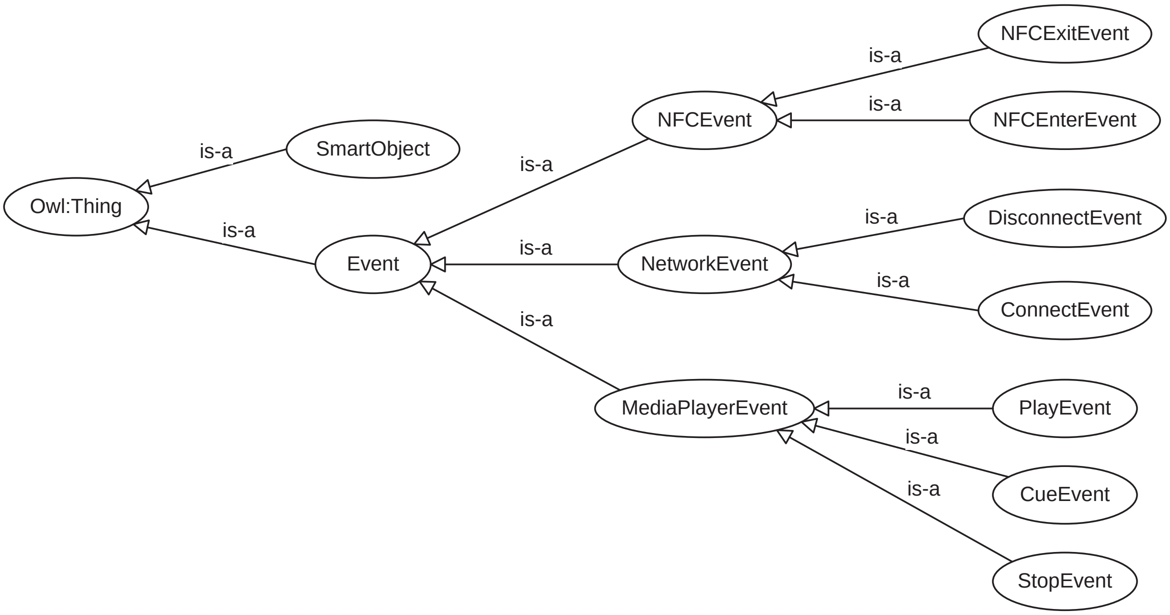


Figure 3.6: Ontology indicating rdf:type relationships

- media players (Nokia N95 and 5800 XpressMusic);
- ambient lighting system (Arduino BT based homebrew lamp with RGB LEDs);
- sound system (speaker-set connected to notebook PC);
- notebook PC (acting as SIB[4]);
- Interaction Tile.

We implemented the demonstrator using the Jena Semantic Web framework, the Processing library for Java, and Python for Nokia's Symbian OS S60. Every interaction with either the music player smart phones or the Interaction Tile results in an interaction event. A reasoner (Pellet[5]) is used to reason about these low-level events in order to infer higher-level results. When a user establishes a connection, two `NFCEnterEvent` events (generated by the RFID reader inside the Interaction Tile) by two different devices not currently connected, will result in a new `connectedTo` relationship between the two devices. Because `connectedTo` is a symmetric relationship, the reasoner will automatically infer that a connection from device A to device B means that device B is also connected to device A. Since `connectedTo` is also an irreflexive property, it is not possible for a device to be connected to itself. A `generatedBy` relationship is also created between the event and the smart device that generated it, along with a timestamp and other event metadata. More implementation details can be found in Chapter 3 of (Niezen, 2012).

3.3.7 A platform for student projects

The demonstrator that was developed (including the Interaction Tile), together with our preliminary definition of a semantic connection, were used as a starting point for student projects within the faculty of Industrial Design (Eindhoven University of Technology). Com-

---

[4] Semantic Information Broker (SIB) is a terminology used in the context of the SOFIA project.

[5] `http://clarkparsia.com/pellet/`

Figure 3.7: Interaction Tile

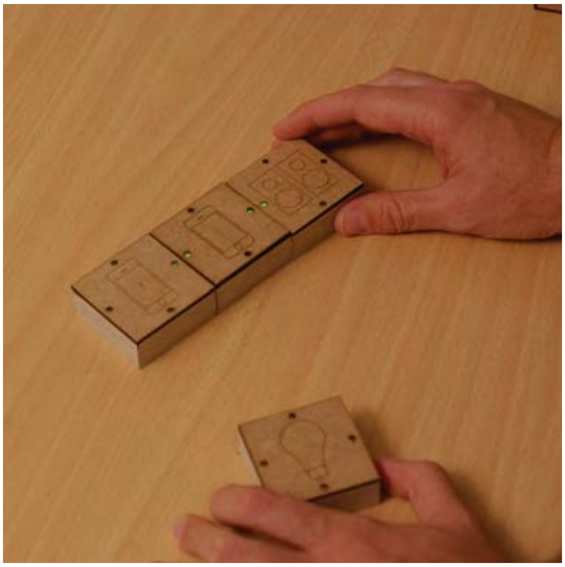


Figure 3.8: Interaction Tabs

bining educational activities with research has showed to be valuable in the past, for both students and the researchers. Our goals for offering these projects to design students were twofold; (1) to teach students the latest developments in the area of ubiquitous computing by offering them to build on the Smart M3 platform, and; (2) by working together with design students we aimed at understanding the needs of current and future generations of designers when designing for interoperability.

Sections 3.4 and 3.5 describe the outcomes of two such design projects. Niezen and me acted as clients[6] for these projects, supervising the students with a focus on the projects' contents. The student work was used for exploration and ideation purposes, and is therefore considered relevant to be discussed.

3.4 Interaction Tabs

Building on the design work described previously, a second design, named Interaction Tabs was developed by Industrial Design masters student Matthijs Kwak (Kwak et al., 2011), to explore alternative possibilities of TUI's. Where the Interaction Tile provides users with a centralized way of exploring, making and breaking connections, the Interaction Tabs provide users with a decentralized way to perform the same tasks.

The Interaction Tabs design was implemented using the same set-up and software, but replacing the Interaction Tile (Figure 3.7) with the Interaction Tabs (Figure 3.8).

3.4.1 Design

Kwak analysed the Interaction Tile using the interaction Frogger framework (Wensveen et al., 2004) and identified improvements that can be considered for the Interaction Tile

---

[6] At the department of Industrial Design, students are supervised by a coach and a client. A client is a stakeholder in the project and the problem-owner, representing either a company or research group/institution.

when placing it in the framework. We decided, though, to stay close to the original design; for the purpose of comparing the designs we considered it to be best to only make small changes, in order to be able to clearly identify what causes a possible change in user behaviour.

By removing the centre tile and moving its functionality into the cubes that represent the devices being connected/disconnected, the interaction with the demonstrator would become simpler, as it would allow for a more direct manipulation of the connections. The (digital) states of the connections are now physically represented, cubes being aligned means that the devices that they represent are connected, and not being aligned means they are disconnected.

Inspired by Siftables (Merrill et al., 2007) the cubes were transformed into tabs, because tabs have a clear top and bottom (Figure 3.8). An LED at each side gives feedback:

Red No connection possible. This occurs when no relation is possible between the two devices of which the tabs are aligned.

Green This occurs when a relation exists between two devices of which the tabs are aligned.

To make a connection, the tabs that represent these devices have to be aligned. To break the connection, the alignment has to be broken.

As a result of removing the central tile, the Interaction Tabs will no longer allow for the exploration of existing connections and connection possibilities without immediately manipulating the connections. Moving towards having a more physical approach influences the scalability and has some other practical implications (e.g. Tabs are smart objects that need to be powered instead of the passive proxy objects that are used with the Interaction Tile, every smart object needs a tab to represent it and connections cannot exist without physically aligning the tabs), which will be described in more detail in Section 3.6.

We expect these differences to also influence the user's mental model, in the way users conceptualize connections and differences in how they imagine devices to be connected, e.g. devices connected in a networked fashion versus connecting devices peer-to-peer.

Kwak also analysed the Interaction Tabs using the Interaction Frogger framework. He viewed the Interaction Tile and Interaction Tabs as being part of the same demonstrator set-up as shown in Figure 3.5, serving as a device to manipulate the connections between the various devices in the set-up. Kwak found that the changes in the design might improve the interaction with regard to:

Direction With the centre tile removed, the direction of making and breaking connections (although done remotely) corresponds better.

Modality With the shaking interaction removed, the modality of making and breaking connections corresponds better.

A more complete design rationale, as well as the analysis Kwak made using the Frogger framework can be found in (Kwak et al., 2011).

3.4.2 Experiment

In order to see which demonstrator would be the easiest to use and allow for a better projection of the users' mental model, Kwak set up and conducted a user experiment to answer the following questions:

> *Are the demonstrators better alternatives, compared to the conventional method of device pairing (i.e. Bluetooth pairing)?*
>
> *Will users be able to work equally well with both demonstrators?*

In the first question, "better" means that exploring, making and breaking connections is easier (more efficient) and more satisfactory (positive user experience). An important aspect is the mental model that the participants develop and how it compares to the system model (see also Section 2.5).

Twelve participants were invited to the experiment; three female and nine male. The participants were between age 21 and 26. With one exception, all participants had a BSc. degree in Industrial Design. One participant also had a MSc. degree in Industrial Design. The setup we used was similar to that described in Section 3.3.6 and depicted in Figure 3.5. For this study, the Interaction Tabs and a simulation of Bluetooth pairing were used in addition to the Interaction Tile. The study took place in the *Context lab* at Eindhoven University of Technology. This lab is furnished to closely resemble a living room, which is the context in which the designs would normally be used.

3.4.3 Measurements and Procedure

Data was gathered about the usability of the demonstrators in comparison to the conventional method of connecting devices, i.e. Bluetooth pairing. We subdivided usability in three aspects; efficiency, effectiveness and satisfaction.

The setup of the test was exploratory, but included two validated methods to gain insights in the participants' mental models and perceived usability, respectively the *Teach-Back protocol* (van der Veer and del Carmen Puerta Melguizo, 2003) and the *System Usability Scale* (SUS) (Brooke, 1996). The action cycle diagram by Norman (1998) (based on the *seven stages of interaction* as is described in Section 2.4) was used as an additional means to elicit the participants' mental models. In addition to these methods, we also collected data about task completion time, errors, recovery from errors and the participants' satisfaction for using the method. A between-subjects design was used and the participants were randomly allocated to one of the three conditions (i.e. Bluetooth pairing, Interaction Tile or Interaction Tabs).

Participants explored, made and broke connections between two mobile phones (a Nokia N95 and a Nokia XpressMusic), a sound system and an ambient lamp, using one of the three interaction methods. Each participant worked through four series of tasks. Bluetooth pairing was tested as a comparative conventional method to measure the usability of the demonstrators. Every session was recorded and notes were made by the moderator.

After a short briefing, the participants worked on the tasks for about 30 minutes (including intermediate discussions). For each interaction method, the participants were given the following assignments: Firstly, users were introduced to the method and given three

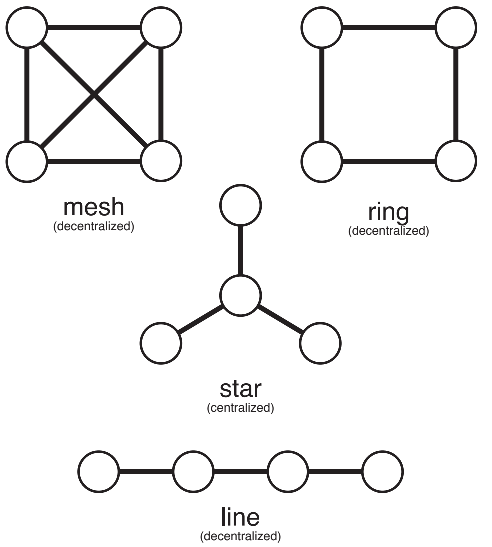


Figure 3.9: Different network topologies users may have reported during the experiments

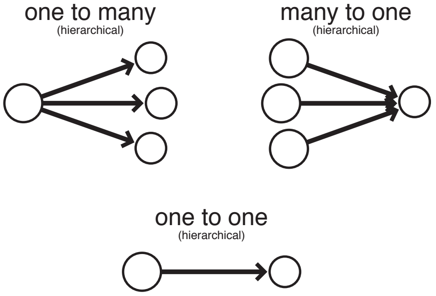


Figure 3.10: Different hierarchical networks

task descriptions. For each description they were asked to connect the devices or configure the demonstrator in such a way as to be able to complete the described tasks (9 minutes). Secondly, users were given a task description and asked to fill in an Action Cycle diagram, listing the seven stages of interaction, needed to complete the task (6 minutes). Thirdly, users were presented with three scenarios. For each scenario they were asked to explain which connections they thought existed in that scenario (9 minutes). Finally, using the teach-back protocol, users were asked to explain to the moderator what they though the method they had used was, and how they thought it worked (6 minutes).

3.4.4 Results

Below, only the most important results of the user experiment are summarized. For a complete description of the user experiment and results please refer to (Kwak et al., 2011).

3.4.4.1 Action Cycle Diagram

The participants reported to follow approximately the same steps to achieve their goals. Eleven out of the twelve participants forgot to specify the breaking of existing connections in "Action specifications". The participants using the Interaction Tabs and Bluetooth noticed this during the execution stage (before thinking they had achieved the goal) and went through another iteration immediately. Of the participants using the Interaction Tile, three out of four participants noticed this after the execution. These participants had to perform this action at a later stage but were still able to achieve their goal.

3.4.4.2 Teach-Back protocol

Although there were differences among the participants within the different conditions, the drawings and explanations of the users' mental models were reasonably similar. None of the

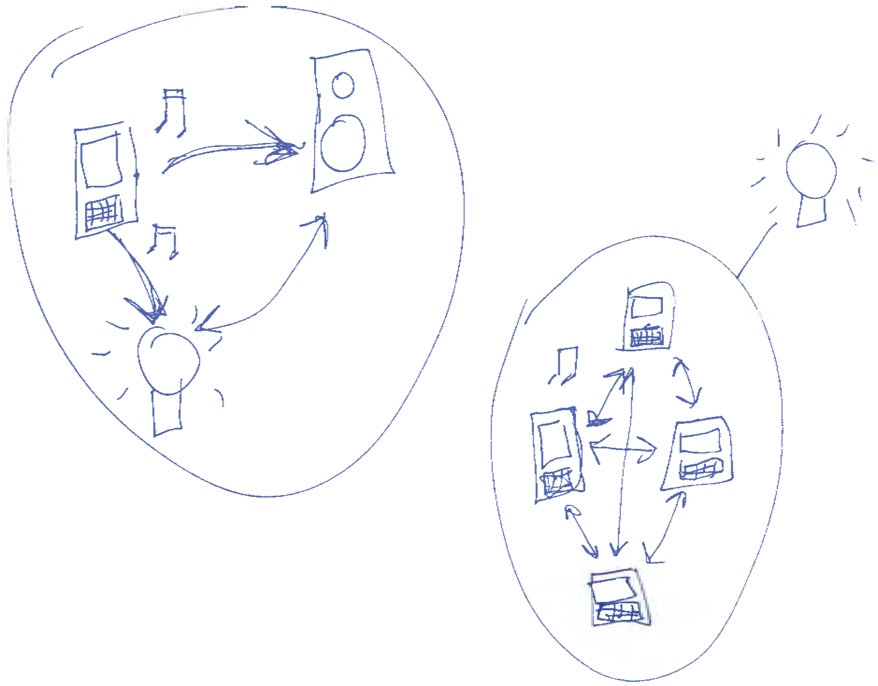


Figure 3.11: Examples of a user's mental model drawing of the network when using the Interaction Tile.

participants described in detail what they thought was happening in the background, but instead focused on the matter "at hand". Three participants (two for the Interaction Tile and one for Bluetooth pairing) mentioned the possibility of extending the current system with more devices (e.g. more mobile phones and a TV). One participant that used the Interaction Tile (Figure 3.11) was able to conclude that the connected devices were networked (i.e. having a mesh topology); the remaining participants described the connections as being either one-to-one or one source to multiple sinks (i.e. star network topology). For an overview of network topologies see Figure 3.9 and 3.10. Figures 3.12 and 3.13 show examples of the mental models of participants from the Interaction Tabs and Bluetooth paring conditions.

3.4.4.3 Observations and post-test discussion

None of the participants using the Interaction Tabs experienced difficulties in performing the tasks. During the post-test discussion they did wonder what was happening in the background. Participants indicated that they expected there was more happening (in the background) than what was visible to them. Using Bluetooth pairing did also not result in any difficulties. Participants mentioned that they were familiar with this way of connecting devices, however, they noted that they had never experienced Bluetooth pairing working so well.

Difficulties did surface for participants working with the Interaction Tile. The first encounter with that interaction device often lead to confusion, as it was not clear to the participants what the relationship was between the central tile and the proxy cubes. The meaning of the pulsing green LEDs to indicate a connection possibility was at first confused with a "working connection". One participant initially thought the LEDs were lasers that

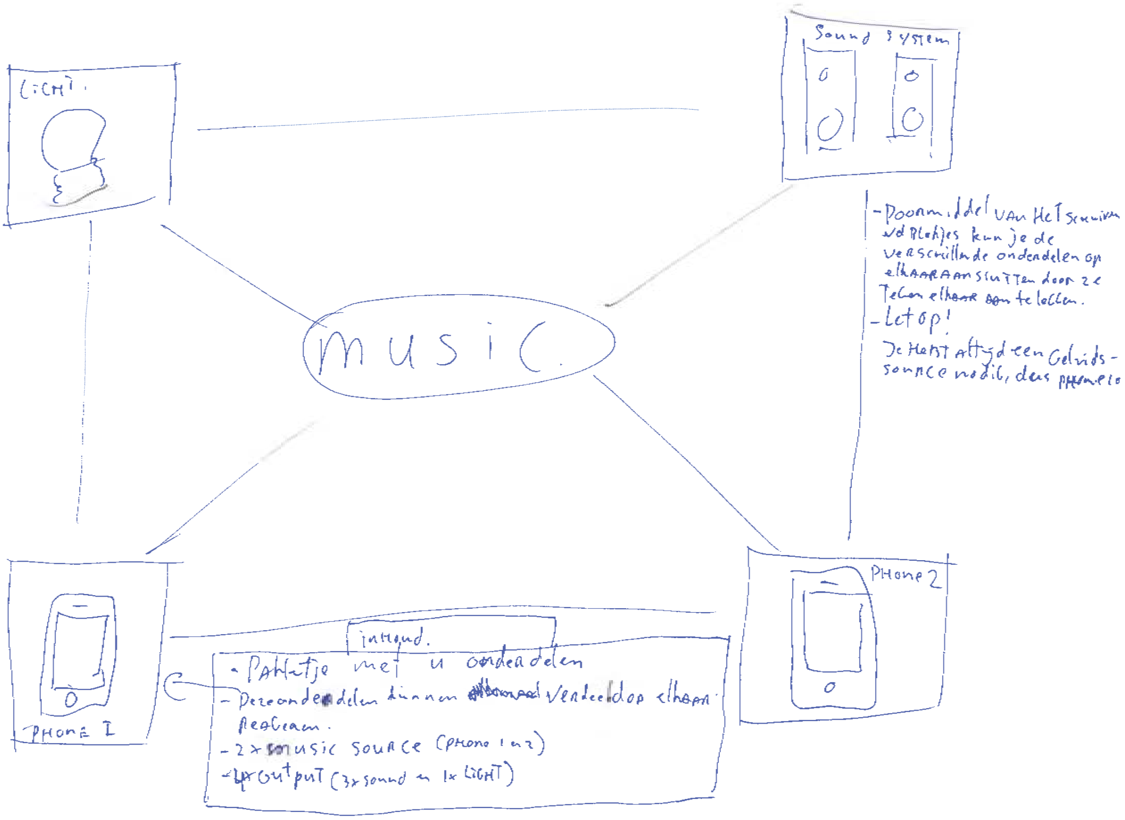


Figure 3.12: Example of a user's mental model drawing when using the Interaction Tabs: "Music can be transmitted from two outputs (phones) to four inputs (phone 1, phone 2, speakers and lamp)".

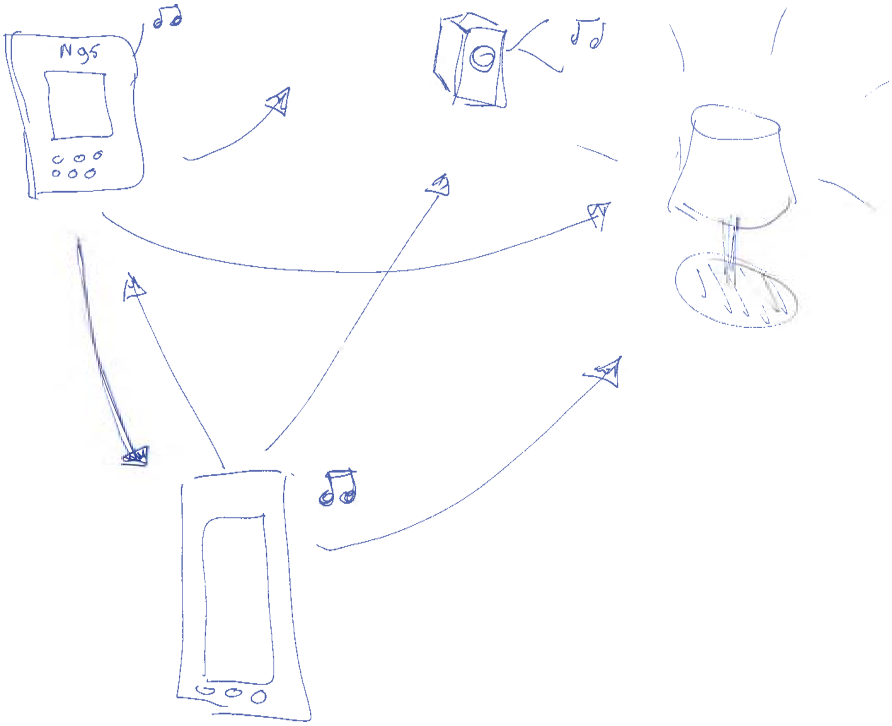


Figure 3.13: Example of a user's mental model drawing of the network when using Bluetooth paring: Data is sent from the phones to the speaker-set, the ambient light or another phone.

could ‘read’ the cubes when placing them on top. Another participant thought it was only necessary to align the cube representing the source device to the central tile and then align the other devices to the source device.

During observations and post-test discussions it became clear that almost all participant (11 out of 12) were not able to derive from the interaction with the system that the connected devices were in-fact networked (i.e. had a mesh network topology). The tasks they performed and the different interface solutions made them conclude that the connections were hierarchical (from source to sink) and participants mainly followed one of two ways to arranging connections:

Line (from one device to the next) - This was observed with the Interaction Tile and Interaction Tabs.

Star (from one device outwards) - This was observed with the Interaction Tabs and Bluetooth pairing.

Some participants arranged connections with the Interaction Tile in a way that indicated they understood the connections to be networked, but they explained verbally to expect the system to make a hierarchy (one source to multiple sinks) out of their arrangement. Interestingly, some participants explicitly mentioned to expect that certain connections should not be possible while in fact they were. For example, they assumed connections to have direction and connections only to be possible from phone to speaker, and from phone to lamp and not between lamp and speaker.

3.4.5 Discussion and conclusions

In one of the examples given in (van der Veer and del Carmen Puerta Melguizo, 2003), the researchers concluded that participants tend to draw little when the system is transparent, as with a less transparent system they are likely to make more detailed drawings to better support their story. The results of this experiment showed a similar level of detail among the interface methods, which may indicate a similar level of transparency for the different interfaces.

The most interesting results came from the observed mental models and post-test discussions with the participants. The observation that most (11 out of 12) thought and acted in a hierarchical way is an interesting one. The Interaction Tile was designed to convey a different way of thinking, but instead participants projected their hierarchical way of thinking on the method. By making connections with the Interaction Tile between no more than two devices at a time they did not use the full capacity of the system, took longer to perform the tasks and were slightly annoyed by the additional effort they had to put in their tasks. Additionally, for those who thought in (star shaped) centralized hierarchies (one device in the centre, the others around it), it appeared difficult to project this way of thinking on the Interaction Tile.

This is where the advantage of the Interaction Tabs showed, because it allows more ways of thinking (e.g. hierarchical, linear, and centralized). The participants found meaning in the arrangement of the tabs and the location of the tabs in relation to each other. For the system this does not matter; a connection is a connection and if devices are connected, they are networked (as symmetric and transitive connections are inferred and automatically inserted, leading to redundant connections).

For such a (simple) scenario, it appeared that the Interaction Tabs performed best. It also seemed that participants were better able to perform the tasks with Bluetooth pairing than with the Interaction Tile but this can be attributed to the fact that they had experience with Bluetooth pairing and connecting devices using a GUI. It must also be clarified that Bluetooth paring was simulated as users were not actually pairing the devices but rather used the interaction style used in Bluetooth pairing.

From this exploratory study, we may conclude that users tend to think in an hierarchical way, indicating a sense of direction in the connections. They also preferred making connections between two devices at a time. These conclusions will be explored further in future design explorations, as are discussed in the following sections and Chapter 4.

3.5 Nodes

The *Nodes* were designed by Industrial Design masters student Jeroen Peeters as a third alternative to explore additional design directions in the same home entertainment setting. Because directionality of the connections was observed to be a natural concept for the users in the previous experiment (Section 3.4.2), this was one of the core concepts that was explored. The Nodes design employs a distributed and localized approach and builds on Gestalt psychology's laws of perception. These hypothetical laws predict perception of visual information in an organized way. In this design they are employed to visualize the otherwise invisible wireless network. In order to gain insights into the use of Gestalt laws to aid in designs that bridge the virtual and physical world, a user experiment was conducted. The Nodes and the Interaction Tile were compared in order to answer the following research question:

> *Is there a difference in the user constructed mental models between the Interaction Tile and the Nodes? And if so, what is this difference?*

It was expected that the Nodes would provide users with a mental model that more clearly resembles the connections between the smart objects, compared to the Interaction Tile. In case of the Nodes, access to the semantic connections is not longer *mediated* through a central entity like the Interaction Tile, but places physical objects that suggest the connections between devices within the system, directly in the environment. This allows users to perceive the network, as it exists within the context, without requiring users to make an additional step in semantic abstraction (i.e. understanding the relation between the configuration made at the Interaction Tile and the semantic connections between the smart objects in the environment themselves, which the configuration *represents*)

3.5.1 Design

Whereas the Interaction Tile is a centralized design—the connections are made by interacting with a central device, irrespective of the location of the actual devices being connected—the rationale with the Nodes design was to explore a different approach to allow users to understand and manage connections between devices in the actual context. As opposed to a centralized solution such as the Interaction Tile, which abstracts the network and takes the connections out of their context, the Nodes are distributed and localized. The Nodes design revolves around physical objects that represent nodes within the invisible

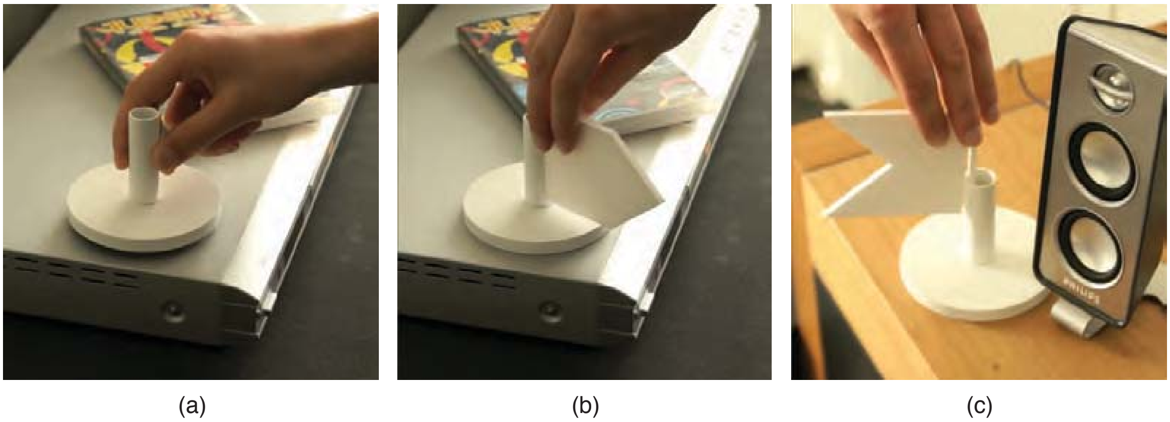


Figure 3.14: Placing a Node on a device (a), placing a network start point on a node (b) and placing a network end point on a node (c). [images provided by: Jeroen Peeters]

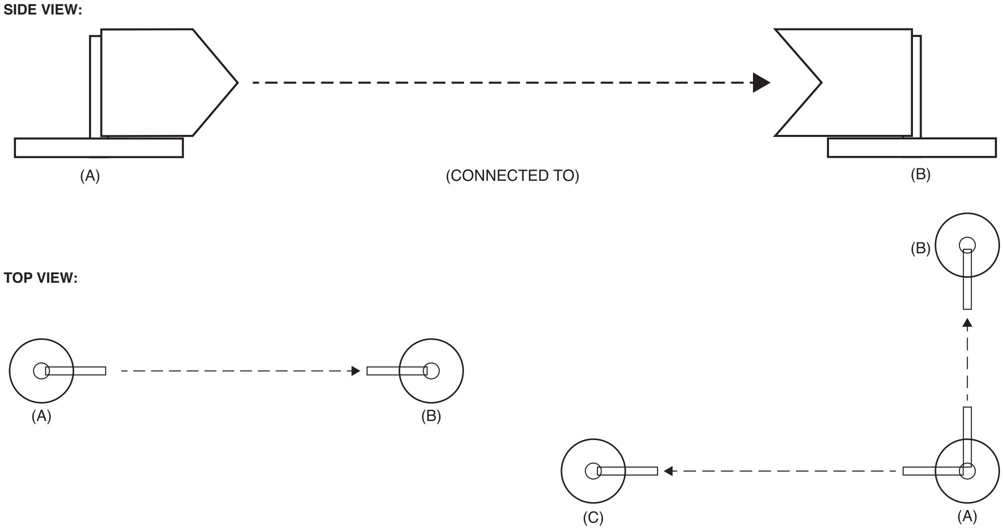


Figure 3.15: Side view: shows how to aim the Nodes to connect device A to device B. Top view: shows two networks: one in which device A is connected to device B, and another in which A is connected to both devices B and C. [image provided by: Jeroen Peeters]

network. The physical nodes are small circular platforms that are distributed in the environment, meaning they are placed close to or onto the actual devices a user wants to connect. Placing the nodes near devices does not yet establish the connections between the devices. To establish connections, users need to determine the start and end points of connections between the nodes. These are determined by placing flat shapes that resemble an arrow (start point) or negative arrow (end point) vertically onto the nodes. (Figure 3.14) By aiming a start point on one node directly at the end point of another node, the connection between two nodes is visualized and established (Figure 3.15).

The Nodes design is based on laws of Prägnanz (also grouping, also see Section 2.6.5), the main principle in Gestalt psychology. Gestalt psychology revolves around the principle that the human mind is holistic and that it has self-organizing tendencies in its perception (Rock and Palmer, 1990). The laws of Prägnanz (Figures 2.16 and 2.15c) are a set of hypothetical laws that allow for prediction as to how visual information is grouped according to certain characteristics. Specifically, the Nodes design builds upon the Law of Closure: The mind has a tendency to complete incomplete forms, effectively seeing something for which it does not receive stimuli. This principle is employed in the design of the Nodes, to visualize something that is invisible (the wireless network) through physical objects that represent parts of it (the nodes and start/end points). The design also employs other Prägnanz laws:

The Law of Proximity objects that are close to one another are perceived to belong together. Used in the design to communicate a node belonging to a specific device.

The Law of Similarity objects that are similar in form are perceived to belong together. Used in the design to communicate the nodes belonging to each other and form networks.

Law of Good Continuation the mind continues visual patterns. Used in the design to communicate connections that cross one another.

3.5.2 Experiment

A user experiment was designed and conducted by Peeters (Peeters et al., 2012), to answer the research question proposed at the beginning of this section. Data was collected by eliciting the participants' mental models to gain insights in the differences between them, when participants were confronted with two proposed design solutions. Fifteen participants in the targeted age of 45 and older were recruited. This age-group was used in order to gain insights into the mental models of users that are expected to be less familiar with the networking of interactive products than generations that grew up with such technologies emerging[7]. In total, eight females and seven males were recruited. All participants reported to use multiple electronic products with varying regularity.

3.5.3 Measurements and procedure

A within-subjects design was used. We employed the Teach-Back Protocol (van der Veer and del Carmen Puerta Melguizo, 2003), asking the participants to explain to an imaginary peer how to perform a specific task with the system, to get insights in how well the participants understood the necessary steps and devices involved to achieve their goal. To support and communicate their mental models, participants were asked to make drawings, schematics or use a textual representation. The data was collected by examining the drawings and descriptions made by participants, as well as from observations and recordings made by the moderator. In the post-test discussion, participants were asked for their feedback and preferences for the two designs.

Video prototypes (Beauduin and Mackay, 2003) were used to convey the interaction and functionality of both designs to participants, using the exact same usage scenario. Video prototypes allow the researcher to have much more control over the behaviour of the system, minimizing the interference of prototyping design flaws or technical instability of the networked devices and networks formed. The use of video prototypes instead of real

---

[7] The students have a high degree of freedom in their projects, and the reasons for their decisions are often different than ours. We acknowledge that in light of the comparison made in this thesis it would have been better to not change the participants' age group. However, the results are still considered relevant

prototypes influences the construction of mental models by the participants, as humans learn differently when seeing as opposed to doing. To minimize this difference, the procedure was adapted: Users interacted with cardboard models of the designs to act out their use of the systems within context and were asked to think-aloud during this step. To do so, we aimed at stimulating users to form their mental models despite of the lack of functionality in the cardboard prototypes.

The experiment was conducted in a controlled environment. An entertainment room at a residency was furnished to resemble a living room, the context in which both designs would be used. Participants were presented with a video prototype of the design and asked to complete a number of tasks using cardboard models as well as writing and drawing. To emphasize the spatiality of the Nodes design, the six devices (VCR, TV, ambient light that reacts to sound, set of speakers, CD-player and a small radio) used in the task descriptions were distributed within the environment. The devices were turned off and to avoid unnecessary confusion they were clearly marked.

Eight different tasks were created for the experiment and every participant was asked to perform all of the tasks in an order that was randomized for each participant. Participants were divided into two groups. One group started the test using the Interaction Tile design, after which they repeated the cycle for the Nodes design. The other group went through the procedure in the reversed order.

Participants were first presented with a video prototype of the design. This video prototype showed a user making and breaking connections between devices using the respective designs. In the videos, the designs appear to be fully functional. Both video prototypes involved the same person, in the same context and managing the same connections. The participant was then asked to use cardboard models of the design to manage connections for two of the tasks. The participant was asked to think aloud and explain what they were doing and why, including how they expected the system to respond to their actions.

Then, employing the Teach-Back Protocol, participants were asked to write down a short general description of the design they were using, as well as to explain to an imaginary friend how they conceptualized the connections in two of the tasks, using drawings.

3.5.4 Results

The collected data was transferred to small cards and analysed using the Affinity Diagram method. Cards were clustered based on their relation to each other, resulting in three categories of interest, presented in the Discussion. The most important results of each technique are summarized in the following sections. For a more complete description of the results please refer to (Peeters et al., 2012).

3.5.4.1 Acting Out

While acting out the tasks with the cardboard models, several participants forgot to perform steps or made small mistakes. With the Interaction Tile, three (out of 15) users forgot to shake the tile, a required action to establish a connection, and instead assumed that simply placing the proxy objects next to the tile would establish a connection. Users were also confused by the icons on the proxies. These mistakes had no observable influence on the mental models that the participants developed of the network.

Using the Nodes design, five out of the 15 participants made mistakes in their use of sender/receiver combinations (e.g. making a connection by pointing two senders at each other, as opposed to a sender and receiver). Two of the participants recovered this mistake quickly, one realized this only when he had to describe the system and the others did not realise this mistake at all.

All participants understood the importance of aiming two senders or receivers at different nodes towards each other. One participant placed two nodes at the same device to establish connections to two other devices (instead of placing two flags on one node as is shown in Figure 3.15), but this did not influence her perception of the network and connected devices. The 14 other participants placed the correct number of nodes at the devices (one for each device). About half (7 out of 15) of the participants required a few moments to decide what nodes were required to send and what nodes were required to receive. The other participants were able to decide instantly.

Two participants were unsure about the placement of nodes relative to the device, i.e. whether they should be on top of the device or whether they could also be in front of the device. Most participants placed the nodes in front of the device, while others mixed placing nodes on top of and in front of the devices. None of the participants expressed worries about height difference in placement of the nodes. Participants were observed to create similar networks in different ways. For example, when performing a task that involved connecting an ambient light that reacted to music, six (out of 15) participants connected the light to the source of the audio (CD player, radio), but most connected it to the speakers that made the audio from the source audible.

3.5.4.2 Written description

Half of the participants expressed that they found it difficult to write down a short general description of the designs. Two participants were only able to describe one system (the one only Nodes, the other only Interaction Tile) and two participants were unable to write a description at all. When participants were observed to be uncomfortable by their inability to describe the system, the moderator skipped this step. For the Interaction Tile, all participants that were able to give a description, described a central entity that is used to connect everything, and which automates the establishing of connections. For the Nodes, all written descriptions mention the existence of two elements: a sender and a receiver (or similar wording, i.e. a device that "delivers a signal" and a device that "needs a signal").

3.5.4.3 Teach-back protocol

All participants were able to use a drawing to explain how they perceived the connections in a certain scenario using a particular design.

For the Interaction Tile design, almost all participants (with one exception) clearly indicated that the connections were mediated by the central entity. They perceived the devices as being connected to the central tile, and that this central entity managed the connections for them (for example see Figure 3.16).Two participants thought the central tile managed the connections by instructing another entity in the network (e.g. a router or hub) to form connections to the other nodes. One participant described the connections as moving around the central tile; i.e. every device connecting directly to another, not mediated by the central unit.

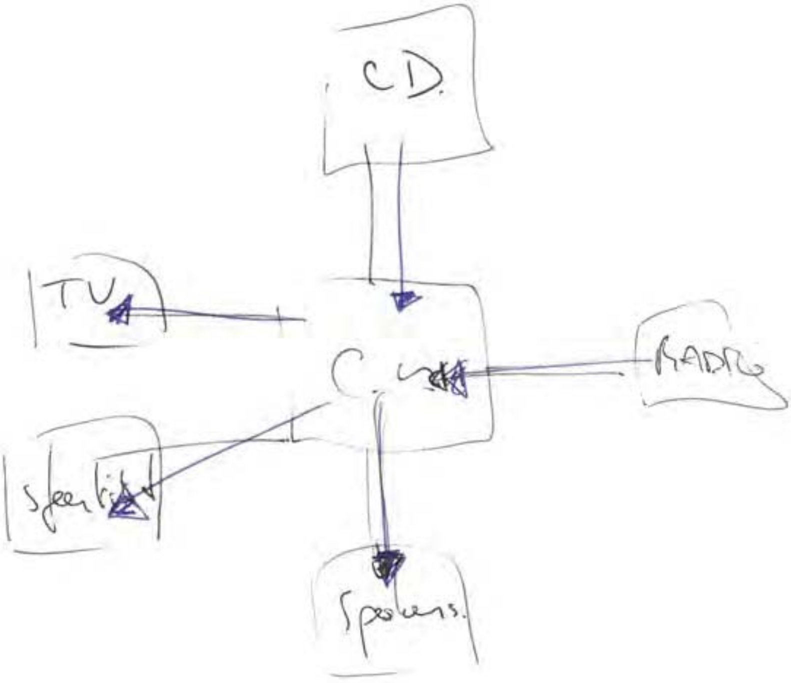


Figure 3.16: A typical drawing of a user's mental model of the network when using the Interaction Tile: all connections go through a central unit.

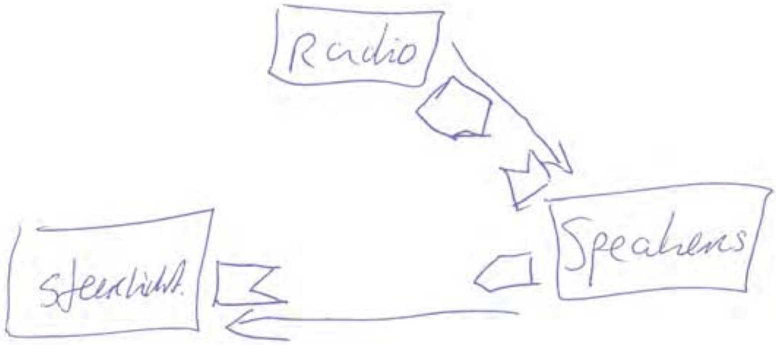


Figure 3.17: A drawing of a user's mental model of the network when using the Nodes design: Data is sent from the radio towards the speaker-set, which acts as a relay to an ambient light.

For the Nodes design, all participants implemented the use of sender/receiver elements correctly and consistently in their drawn explanations of the connections. Every participant drew hierarchical connections between devices, where devices send data and others receive it (see Figure 3.17).

3.5.4.4 Post-test discussion

During the post-test discussion, participants expressed that they wondered what was happening inside the Interaction Tile. They perceived it as being automated. One participant explained that he found it difficult to understand the system because he was unaware of what happened inside the Interaction Tile.

3.5.5 Discussion and conclusions

Analysing the data using Affinity Diagrams, three categories of results emerged from the data. These categories were applicable to both designs. These categories will be used to discuss the differences between the two designs in the following sections.

3.5.5.1 Network structure

For the Interaction Tile, all written descriptions of the system by the participants revolved around a central device that is connected to all devices and manages the connections automatically. Participants often referred to the Interaction Tile as the "central unit", "the interface to all devices" or "a magic box". With one exception, all participants also indicated in their drawings that they perceived all connections to go through the Interaction Tile, where the Interaction Tile "did something" to the signals and created the network. This suggests that users develop a mental model with a centralized network structure when they interact with the Interaction Tile; all devices are connected to and controlled by a central object, the Interaction Tile.

Despite the mistakes that were made by participants in their use of sender/receiver elements in the acting out tasks, all participants implemented this differentiation correctly and consistently in their drawn explanations of connections in the Nodes design. For the Nodes design, all written descriptions report "senders and receivers" (or similar wording) to facilitate the connections. They also indicated that the placement of the nodes near the devices that needed to be connected was necessary to include them in the network. This was also reflected in their drawings. Participants created different mental models of networks with similar functioning, and were able to adapt the use of the system to fit their mental model without compromising the functionality of the network. For example, in a network of three devices, music from a CD player plays on the speakers and an ambient light responds to the music. Most (9 out of 15) participants directed the signal from the CD player to a receiver on the speakers, and relayed the signal from the speakers to a receiver on the light. Six participants sent two signals from the CD player, one towards the speakers and one towards the light. This shows a powerful characteristic of the Nodes design: it supports users in projecting different mental models on the system.

It can be concluded that the results of our experiment supports our hypothesis that the Nodes design provides users with a mental model that more clearly resembles the relationships between the entities in the network, in the sense that devices are directly connected to each other without the network being mediated by a central unit.

Adding direction to the connections, and integrating it explicitly in the design also leads to a better predictability of the functional result of the networks that are established. With the Interaction Tile, connections have no explicit direction (and therefore are symmetric). The results of this experiment (and also that of Kwak (Kwak et al., 2011)) show that users tend to add a sense of direction in their mental models themselves. This may lead to unexpected behaviour. For example: when device A is connected to device B and A is connected to C the behaviour is different compared to connecting A to B and B to C, as in the first case actions on B do not influence what happens on C. This difference is more clearly to be predicted by users when they use the Nodes instead of the Interaction Tile (also see Figure 3.20).

3.5.5.2 Design semantics

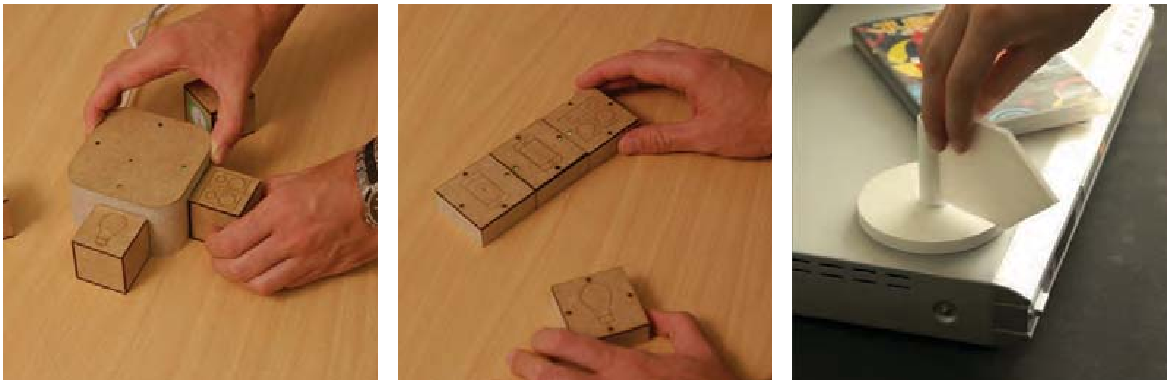
For the Interaction Tile, some participants were confused about the meaning of the graphical icons on top of the blocks (i.e. which device was being represented by which icon). For the Nodes design, similar problems surfaced. It was difficult for some participants to immediately apply the sending/receiving concept in their acting out tasks, and some did not realize the importance of using the right arrow-shape to send or receive a signal. In their drawings, however, all participants used the sending/receiving principle correctly in explaining connections. This indicates that the system could benefit from a better form design to allow differentiation between the sending and receiving shapes. Furthermore, two participants wondered whether the location of the nodes relative to the device was important, although they assumed that it only had to be in close proximity.

These issues occurred equally often for each design and only with a minority of the participants. Identifying these issues or not, did not influence the mental models that were collected using the teach-back protocol. They do however point out important design issues that can be limitations to both systems. This suggests that further research into the semantics of sign and form in both designs could lead to a better understanding of the interaction required for the device, as well as increased usability.

3.5.5.3 Users' preference

From the observations and post-test discussions, statements concerning the participant's preference for either design emerged. When asked about their opinions on both designs, almost all participants indicated they preferred the Interaction Tile over the Nodes. They described the Interaction Tile as being very easy to use, as they only had to add representations of the devices to the Interaction Tile, and did not need to determine what the role of each device was (i.e. in terms of sender or receiver). The system was perceived as being automated and therefore experienced as the most user-friendly one. Furthermore, they liked the fact that they were able to manage the connections remotely, without getting up and moving around.

Most participants indicated they found the Nodes easy to understand, but that it required too many actions (i.e. placing the Nodes in the environment, selecting and adding the required end-points for the connections) . Furthermore, some participants indicated that they would not like having to place additional objects in their living room, for which they did not see specific merit. This suggests that although the Nodes design provides users to develop an accurate mental model of the network, this does not necessarily lead to better usability. Further research could explore design directions that merge the merits of the Nodes design (accuracy and clarity of the mental model, flexibility towards different mental models) with those of the Interaction Tile design (ease of use, perceived value).



(a) (b) (c)

Figure 3.18: A overview of the different designs that are discussed in this section: (a) Interaction Tile, (b) Interaction Tabs, and (c) Nodes.

3.6 Tiles, Tabs and Nodes

In this section the differences between the three different designs are discussed (Figure 3.18). Even though the designs are not directly comparable, as they differ in expressive power and supported functionalities, we can learn important lessons:

3.6.1 Centralized vs. decentralized

If we view the interactions with the three different designs, in relation to the networks the interactions result in and the mental models users may develop, a first distinction we can make is between a centralised and decentralised model. While the network structure at the lower level is centralised (i.e. every smart object is connected to a central SIB and communicates through this blackboard), the first layer of semantics we introduce are `connectedTo` relationships between the smart objects which are connected (decentralised). The mental models that users develop when using one of the designs to interact with the `connectedTo` relationships may, however, be different.

The Interaction Tile approach introduces an additional object (the centre tile) as a physical representation of the invisible connections. However, this additional objects was perceived by some users (as was observed by Peeters) as a central entity like a hub in a centralised network (Figure 3.19). Because the connections in our demonstrator are symmetric, the difference does not lead to other behaviour and the different mental models are thus considered compatible.

Also with the Interaction Tabs, users seemed to project a centralised (or hierarchical) view when they interacted with the Tabs. Participants would for instance take a source object (like one of the phones/media players) and connect the sink objects (e.g. lamp, speakers) to the source. Similar constructions surfaced during the user experiment with the Nodes design, however, with this interface solution daisy-chained (line) connections were more common. With the Nodes design, because of the added notion of direction, different configurations do (potentially) lead to different behaviours (see Figure 3.20).

Interestingly enough, the two user experiments (Nodes vs. Interaction Tile) and (Interaction Tile vs. Tabs) resulted in different trends in the mental models for the Interaction Tile. While Kwak (Kwak et al., 2011) reported mainly peer-to-peer and daisy-chained

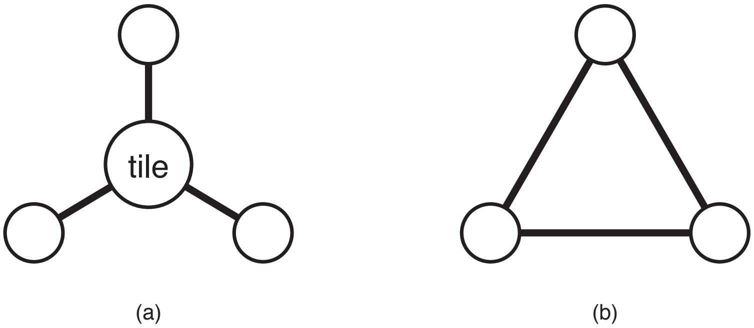


Figure 3.19: (a) A centralised model with the Interaction Tile in the centre and (b) the network as it is created when the connections are made.

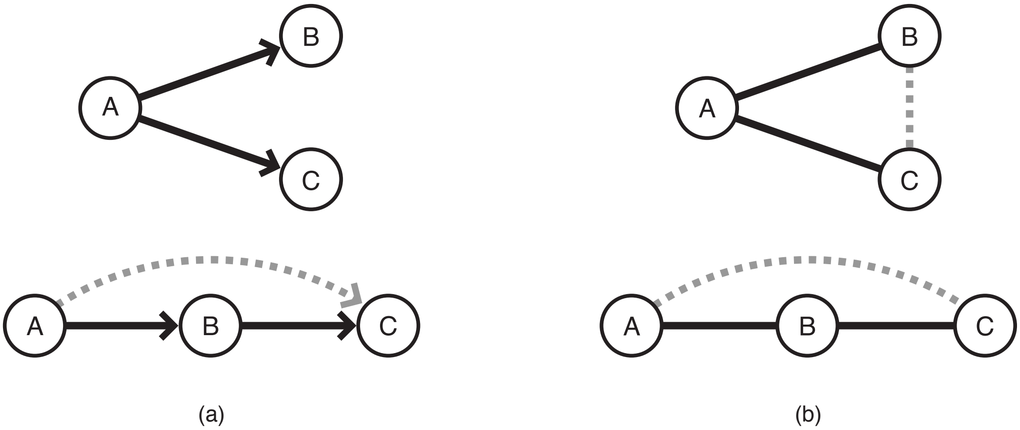


Figure 3.20: (a) with (one) directionality, both configurations may lead to different behaviour as in the top configuration, B and C are not connected (b) with symmetric connections (due to transitivity) both configurations have the same result.

models for the Interaction Tile, Peeters (Peeters et al., 2012) found that the majority of the mental models showed a centralised structure, with the Interaction Tile as a central entity. This difference may be attributed to the different age groups for the experiments. Alternatively—and more likely—it may have been caused by the differences in the methods and procedure. Peeters showed the participants video prototypes having the participants watching someone else performing the interactions first, before they were asked to perform these actions themselves. In Kwak's experiment (Kwak et al., 2011), participants were confronted with the demonstrators without any explanation. Therefore, participants in Kwak's experiment may not have been encouraged to explore the different ways of interacting with the Interaction Tile, and simply continued connecting the objects one-by-one after they found out that this particular way worked.

3.6.2 Directional vs. symmetric

As is shown in Figure 3.20 and briefly described in the previous section, directionality is a key concept. Although in the demonstrator setup all semantic connections are symmetric, many users appear to project their ideas of direction onto the Interaction Tabs and Interaction Tile. Even though we decided to simplify the concept of connections for the demonstrator, most participants think to observe directionality in the system, as music (and possible light) seems to travel from one place to the other. In the Nodes design the concept of (one) directional connections was added, giving the user more expressive power. This makes the interaction with the system more complex, as users have to manually create the inverse connections wherever a symmetric connection is desired.

3.6.3 Digital state != physical state vs. digital state = physical state

The designs are physical interfaces to the intangible connections and employ different mappings between the physical states of the interfaces and the digital states of the connections. The Interaction Tile uses *augmented* information (i.e. red and green lights) to reveal the status of the digital connections. As a consequence, semantic connections can exist even when the Interaction Tile is not in use (e.g. turned off, stored away or not having any proxy- or smart objects aligned to it). Connections can thus be accessed on demand, or can be explored/manipulated with other interfaces (such as the Spotlight Navigation described in Chapter 4).

The Interaction Tabs and Nodes view connections differently, as the physical state of these objects is directly coupled to the digital states of the connections. Even though this way of coupling digital and physical states is powerful and easy to grasp by users, it has practical implications. The Interaction Tabs are more suitable to make ad hoc connections between devices to perform a task that is temporary. It is impractical for situations where configurations should be made that need to be maintained over a longer period of time as the physical configuration should be kept aligned.

While the Nodes are more suitable to make persistent configurations, it does imply that one has to place physical objects around the home environment, also connecting mobile devices (that are gaining popularity in peoples' homes) is difficult.

Besides these more practical implications, that may be omitted as the designs were never intended as product designs but rather as research tools, these different design decisions may also influence the mental models users develop as the physical dimensions of the configurations do not always match the digital connections they are signifying. As an example, people may believe that the arrangements of the Interaction Tabs in Figure 3.21 result in different network configurations, while in-fact, they are the same.

Similar examples can be given for the Nodes design. While people may or may not infer transitive connections between smart objects, they are inferred and created in the digital domain. These automatically inferred relations are observable in the physical world as the smart objects behave accordingly but are not directly perceivable (Figure 3.22). With Interaction Tile, these connections can be visualised and may help users to better understand more complex configurations.

Physical shapes also restrict meanings. When connections are not transitive, the rectangular shape of the Tabs prevents the creation of a ring network structure with three entities.

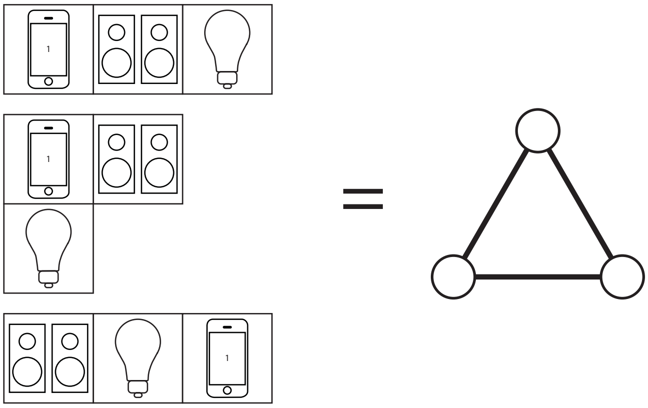


Figure 3.21: Although the three arrangements of the Interaction Tabs appear different, they all result in the same network structure.

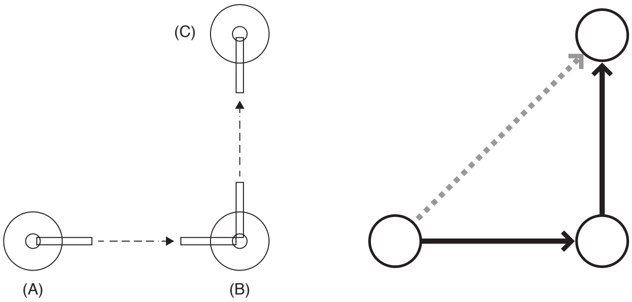


Figure 3.22: Transitive connections exist in the digital domain and are not physically represented by the Nodes. They are, however, perceivable and may be inferred by people as well. As a result, when undesired behaviour occurs, they cannot be removed (as they are not physically represented).

Therefore, the physical shape of the Tabs may intentionally or unintentionally restrict this use (Figure 3.23).

3.6.4 Remote interaction vs. interaction distributed in physical space

Another key difference between the designs—in particular between the Interaction Tabs and Interaction Tile on the one hand, and the Nodes at the other hand—is that of remote interaction, in contrast with interaction distributed in space. Both ways of interaction require some sort of representation of, either the network (connections) or the smart objects that are connected.

The Nodes design uses network nodes i.e., the end points of the semantic connections and are distributed throughout the environment. By placing these nodes next to, or on-top of the devices, endpoints are linked with the devices. For the Interaction Tabs, the devices are physically represented by the Tabs themselves. The Interaction Tile also uses physical representations of the devices, although mobile devices may be directly aligned with the tile. These differences also have practical implications. The Interaction Tile and Tabs can be used remotely as for the Nodes users have to get to the devices and appliances they

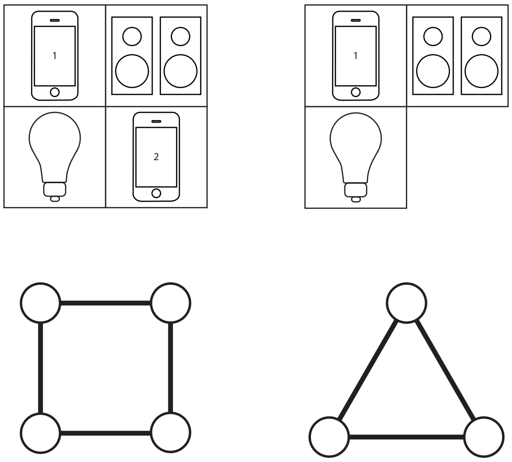


Figure 3.23: Two (similar) network structures are physically not possible due to physical restrictions of the Tabs. Tabs only (physically) allow for ring structures with an even number of nodes.

wish to connect. Users may have preference for one of the two ways of interacting, but also the context of use will influence this. As for connections that are often manipulated, remote access may be highly appreciated, for connections that are only changed occasionally preferences may be less strong.

3.7 Discussion

With the designs presented in this chapter we made the invisible wireless connections visible. We also took it a step further by enabling users to physically explore and manipulate the connections. Implementing the demonstrators helped with defining more specific questions and identifying key issues in using product semantics theory to design for bridging the digital connections with their physical counterparts. Even though the demonstrator simplifies reality, it shows that making high-level semantic abstractions of normally low-level tasks has potential to allow for *semantic interaction* in home-network configuration tasks.

Building the demonstrators also identified possibilities for improvements and extensions. Like discussed before, it currently does not distinguish between different types of information exchanged, nor do the Interaction Tile and Tabs show directional properties of the connections. In the Nodes design, we added direction to the connections by integrating it explicitly in the design. The results of the experiments by Peeters (Peeters et al., 2012) and Kwak (Kwak et al., 2011) show that users tend to add a sense of direction in their mental models, even when this is not reflected in the interaction with the connections.

Besides these observations, the demonstrator shows that even the slightest and simple ways of giving feedback (lighting colour and dynamics) can reveal meaningful information. To what extent users can extract meaningful information from the interactions with the smart space and how they can use it to build a suitable mental model for understanding, was evaluated in two user experiments. The results show that people are capable of constructing mental models that are *compatible* with the system model, which should be considered a positive outcome. Additionally, direction was observed to be a natural property of the connections and something that users used in their mental models, even with its absence.

The mental models in users seem to be influenced by various variables. The user studies showed that potential variables that influence the mental model are: age group, medium (video prototype, tangible prototype), the design itself and previous experience with similar products/systems.

The last decade(s) of HCI research shows an ongoing pursuit of making digital information and content physical, to allow for a natural way of accessing and controlling such data. Although there is rich potential in tangible interaction concepts, trade-offs are inevitable. One problem of tangible computing is the introduction of many new physical objects into the environment (as was shown in particular by the Nodes design). Leaving information in the digital world has advantages—we do not always want to have physical representations of all the information that we generate in the digital world, which would mean overcrowding the physical space.

A relatively unexplored approach is to use the existing physical (electronic) objects and devices in the interaction with the virtual world, going beyond using their (touch) screen and or buttons to interact with the information world. We propose to use the physicality of the objects e.g., their context, relative position and proximity to each other to generate new interaction concepts. We also expect that the physicality of the objects itself and the context in which they are used, are the main providers of meaning. The connection created between a MP3-player and a stereo set has a meaning in itself because of the resemblances in meaning of the two (being able to play music). Putting a photo camera close to a smart phone could mean the user would like to exchange all or maybe the most recent image between the two. It is our challenge to not only make it happen technically, but to enable users to express their wishes in terms of information exchange in a more meaningful way.

3.8 Concluding remarks

This chapter described a first design exploration of the meaning of connections between interoperable objects in smart spaces. The demonstrator that was built and the three interaction designs show examples of how the notion of semantic connections can be used in user interaction with a smart space. We identified the following questions that are explored in the remaining part of this thesis: How do we handle increased complexity? How should information about the information/content that is exchanged or the functionality of the connections be revealed? How is control over the content that travels over the connections provided? How can the design of physical objects enhance the creation of suitable mental models in users? Furthermore, we need to identify whether this way of interaction can be generalized and applied in different contexts in the home.

Where smart systems or environments in some sense try to understand or predict what the user is trying to accomplish, by being adaptive and anticipatory, we aim to identify ways to give users appropriate means to express their intentions. The possibilities, available services and information that exist in the smart environment needs to be communicated in a meaningful way. Only if this is done correctly will users be able to build helpful mental models of the functionality the environment has to offer, set goals and make plans on how to act.

Work was also done on making the match from the other side—the side of the smart space. By using technologies originating from the Semantic Web and ontologies (see Section 2.3) to define common concepts and relations, we aim to make a better match between system's internal models of interaction and the user's mental model. We see a vital role here for the theory of product semantics, the study of how artefacts acquire their meaning, and use its theories to define common concepts and describe the meaning devices may have for their users. Please refer to (Niezen, 2012) for a description of the work done.

Chapter 4

Design Exploration II

4.1 In this chapter

Based on the findings that are discussed in the previous chapter, this chapter introduces a second design iteration. First, we introduce a refined definition of semantic connections and describe a theoretical framework which grounds our ideas. This refined definition is then used to build a new demonstrator, building on the SOFIA IOP. In this new use-case, we worked together with project partners to demonstrate interoperability between devices running on different platforms, from different manufacturers. The use-case is build around our ideas of semantic connections, and new interface solutions to interact with semantic connections are introduced. We developed the Connector, a tangible interaction approach, enabling users to physically select devices in their environment and directly view and manipulate the connections in a simple, universal way. The Connector is described in Section 4.4. In cooperation with Conante[1], a second prototype was developed, based on the Spotlight Navigation device (Rapp, 2010) to access semantic connections using a projected augmented reality approach (Section 4.5). Both designs were evaluated in a user experiment which is described in Section 4.7. The results (Section 4.8) and a discussion (Section 4.9) conclude this chapter.

4.2 Semantic Connections: defining a framework for design

We defined the term *semantic connections* (Section 3.2) to refer to meaningful connections and relationships between entities in a ubiquitous computing environment. In this section we redefine semantic connections, incorporating insights we gained in the first iteration.

Semantic connections make up a structural layer of inter-entity relationships on top of the network architecture. The connections can represent the real, physical connections

---

This chapter is largely based on:

Van der Vlist, B., Niezen, G., Rapp, S., Hu, J., & Feijs, L. (2013). Configuring and controlling ubiquitous computing infrastructure with semantic connections: a tangible and an AR approach. *Personal and Ubiquitous Computing, 17*(4) Springer-Verlag London. pages 783–799.

[1] one of the partners in the SOFIA project

(e.g. wired or wireless connections that exist between devices), or conceptual connections that seem to be there from a user's perspective. Their context (what things they connect) is pivotal for their meaning. The term "semantics" refers to the meaningfulness of the connections. We consider the type of connection, which often has the emphasis now (e.g. WiFi, Bluetooth or USB) not to be the most relevant, but that what the connection can do for someone—its functionality—even more.

Semantic connections have informative properties, i.e. they are perceivable in the physical world and have sensory qualities that inform users about their properties and uses. However, these physical qualities might be hidden at some times, or only accessed on-demand, by a special purpose interaction device. The digital counterparts of semantic connections are modelled in an ontology. There may be very direct mappings, e.g. a connection between two real-world entities may be modelled by a `connectedTo` relationship between the representations of these entities in an ontology.

Semantic connections have properties like directionality and are transitive. The rationale behind semantic connections is to rely on:

- the semantics of existing objects to provide meaning for the relationships between the objects and the resulting meaning of the *networked* objects.
- the power of natural mapping and locality, using real objects and locations to provide meaning for the connections that are created between the objects and (object) locations.
- inherent, augmented and functional feedback and feedforward to strengthen the meaning of the connections and the emerging functionality.

Crucial to our approach is to make the gap between user goal and action smaller. If we consider streaming music from one device to another, "streaming" now consists of multiple actions that do not necessarily make sense. In our view, this single high-level goal should have one (or at least as few as possible) single high-level action(s). That single action should carry the meaning of its goal. By using the physical world as interaction space and using the real location of the objects, we aim to reduce the need to identify the devices from a list with names or rely on other forms of textual representation.

4.2.1 Semantic connections interaction model

A user interaction model for semantic connections is shown in figure 5.3. It describes the various concepts that are involved in the interaction in a smart space and shows how these concepts work together. The interaction model was inspired by the Tangible Interaction model (MCRpd) by Ullmer and Ishii (2000) which in turn was based on the Model View Controller (MVC) model (see also Section 2.4.4).

We distinguish between the physical part of the user interaction and the part that takes place in the digital domain. A user cannot directly observe what is happening in the digital domain (and should not) but experiences the effect it has in the physical world, by interacting with the various smart objects and the (semantic) connections that exist in between them. In doing so, users create a mental model of the objects/system they are interacting with, which only partly (or not at all) includes the digital part. Digital information manifests itself in the physical world *through* the smart objects as information, media and services.

When a user interacts with a smart object, connected to the smart space, he/she senses feedback and feedforward, directly from and inherent to the controls of the device (inherent

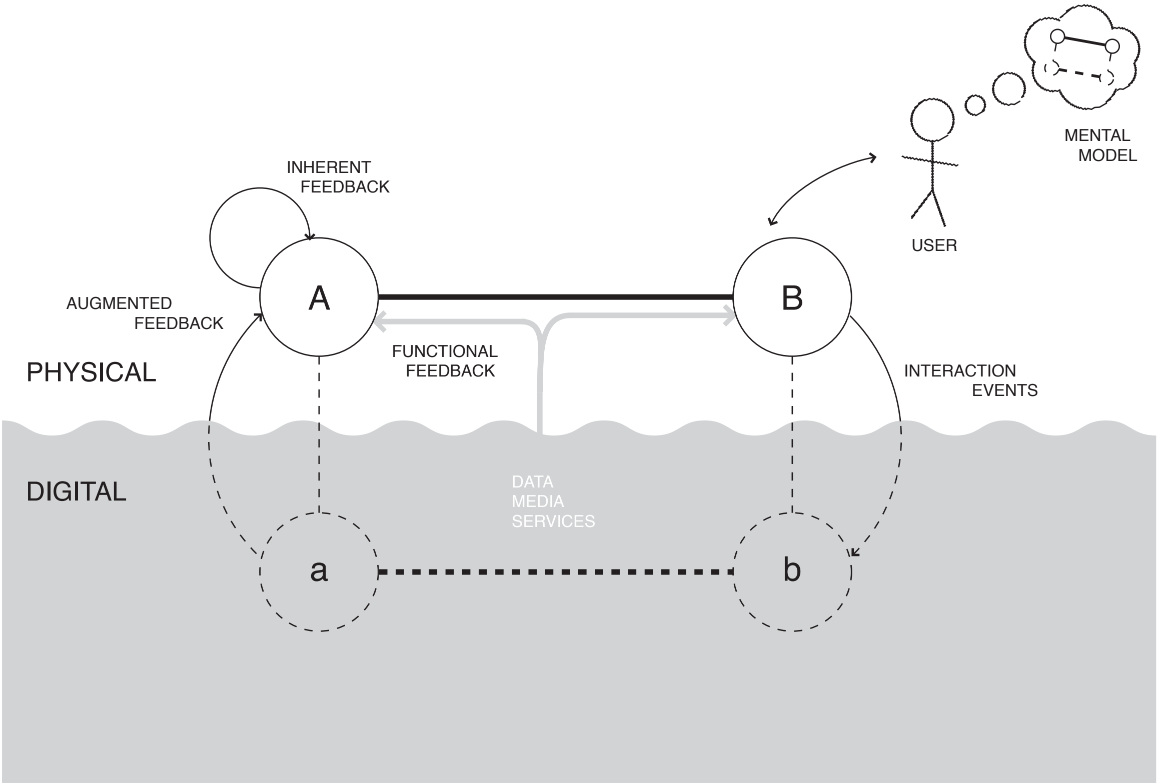


Figure 4.1: Semantic Connections user interaction model

feedback), digital information augmented onto the physical world (augmented feedback) and perceives the functional effect of the interactions (functional feedback). The terminology; inherent, augmented and functional feedforward and feedback is adopted from (Wensveen et al., 2004) as is described in Section 2.6.1. Figure 4.2 shows examples of different types of feedback as they are used in our first iteration as described in Chapter 3.

The user actions in the physical world are transformed into interaction events and events/state changes, using semantic transformations. This interaction data in terms of user intentions, possibly together with user preferences, defaults and context information is stored in the smart space (also called SIB; Section 2.3.4).

4.3 Related Work

The past decade of research has resulted in many proposals for configuring ubiquitous computing infrastructure and solving the interconnection and interoperability issues with consumer electronics. In this section, a selection of work is described which is directly related to our work in general or to one of the designs described in Sections 4.4 and 4.5.

Early work by Siio, Masui and Fukuchi (Siio et al., 1999) explored interactions with connected devices by moving a stylus along paths on a printed map of the infrastructure that is annotated with barcodes. More recent related work presents solutions for simplifying configuration tasks of in-home networks by creating virtual “wires” between physical objects like memory cards (Ayatsuka and Rekimoto, 2005) that can interconnect devices.

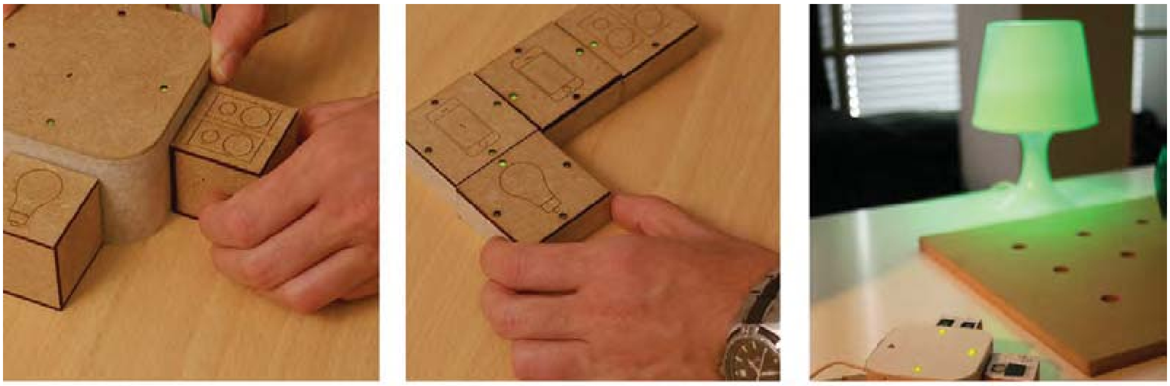

(a) (b) (c)

Figure 4.2: Examples of the different types of feedback: (a) Augmented feedback; (green) lights showing a connection currently exists. (b) Inherent feedback; the feeling of a "snap" when two tiles are aligned. (c) Functional feedback; a light rendering the mood of the music when a music player is connected to it.

Others propose to introduce tags, tokens and containers (Want et al., 1999; Holmquist et al., 1999) for tangible information exchange. Concepts like "pick-and-drop" (Rekimoto, 1997) and "select-and-point" (Lee et al., 2008) are used to manage connections and data exchange between computers and networked devices. The introduction of near field communication, i.e. using a near field channel like radio-frequency identification or infrared communication, allows for direct manipulation of wireless network connections by means of *proximal interactions* (Rekimoto et al., 2003).

Fitzmaurice (Fitzmaurice, 1993) states that we should browse, interact, and manipulate electronic information within the context and situation in which the information originated and where it holds strong meaning. He adopted the notion of *situated information spaces* to avoid being overwhelmed by too much information. This means that the information associated with physical objects should be collected, associated, and collocated with those objects. The physical object anchors the information and provides a logical means of partitioning and organizing the associated information.

Rukzio and Holleis (Rukzio and Holleis, 2010) discuss the design space of interactions and applications enabled by pico projector units integrated into mobile devices. Three interaction concepts that are directly related to our prototype are: changing the location and orientation of the projecting device; projection acting as a magic lens, revealing a part of a virtual information layer which is much bigger than the actual projection (magic lens metaphor); and projection showing information related to the object on which the projector currently focusses (augmented reality).

Ballagas et al. (Ballagas et al., 2006) surveyed interaction techniques that use mobile phones as input devices to ubiquitous computing devices. The survey is based on Foley's taxonomy (Foley et al., 1984), where input devices are structured according the graphic subtasks they can perform: position, orient, select, path, quantify and text entry. During our design phase we considered and implemented a number of these subtasks for making and breaking connections.

Three physical mobile interaction techniques were evaluated in (Rukzio et al., 2006). *Touching* refers to bringing the user's mobile device into contact with the object the user wishes to interact with. Using *pointing* it is possible to select a smart object with a mobile device by aiming at it. *Scanning* allows users to get a list of nearby smart objects by using a wireless mechanism. We incorporated variations of the first two techniques in our prototypes.

In (Hinckley et al., 2004) an interaction technique called *stitching* is described, that allows users to combine pen-operated mobile devices utilizing wireless networking, by using pen gestures that span multiple displays. For example, a user can transfer photos from one device to another by drawing a path from the photo collection on one device to another device, skipping the bezels of the screens. An advantage of this technique is that since there is natural order implied by the gesture, stitching establishes which device is the sender and which is the receiver. In our augmented reality and tangible UI approach we made use of implied natural order in a similar manner.

A projector-camera system was used to augment smart objects with additional information by Molyneaux et al. (Molyneaux et al., 2007). The system is capable of locating and tracking the objects and projecting information onto the objects themselves, by aligning the projection with the object's surfaces.

Beardsley et al. (Beardsley et al., 2005) developed a handheld projection system that lets users create opportunistic displays on any suitable nearby surface. Fiducial markers and a camera are used to calculate and transform the projected image plane. They also described the interaction technique of selecting a physical region of interest, based on the "hold-and-drag" motion of a projected cursor.

Another line of research investigates how to combine functionalities of several devices or services in a fully automatic way, to solve explicitly stated user goals without involving the user in the combination (Heider and Kirste, 2002; Torge et al., 2002). In contrast to these efforts, the Speakeasy project (Newman et al., 2002) follows the recombinant computing approach, and allows end-users to specify *connections* between entities to transfer data. In direct connection mode, users can discover, control and connect any available component on the network. Recombinant computing is a system design philosophy that follows a bottom-up approach to creating computing environments, where individual entities form part of an elastic, always-changing whole. These entities are designed to be able to interact and interoperate with one another *without having prior knowledge of one another*. They expose simple programmatic interfaces called recombinant interfaces, which govern how they can be made to interoperate.

4.3.1 Near-field interaction

The Connector, one of the designs that is described in the next sections, uses an RFID reader to *physically* identify and select smart objects. Building on RFID technology, which involves RFID readers and passive *tags*, Near Field Communication (NFC) was developed. NFC is a set of standards for mobile devices to enable communication over the near-field channel by touching two devices together or holding them in close proximity. The technology has been implemented in various smart phones and found its way into interaction design (also see Section 2.2.2). Currently, NFC is often associated with contactless payment systems and simple data exchange, but it may also be used for simplified setup of other communication technologies such as Wi-Fi and Bluetooth, eliminating the nuisance of Bluetooth pairing or connecting to Wi-Fi networks with passwords.

NFC technology reintroduces the physical qualities of earlier near-field communication technologies like IrDA (Infrared Data Association). When using infrared communication between devices, they had to be close to each other and have a line of sight. Because of these physical restrictions, IrDA was (reasonably) secure and did not need passwords, it were the physical restrictions that made people trust that their phones would not be transmitting anything when they were in your hand or pocket. NFC has similar requirements, as it needs physical proximity (even though accidental physical alignments of two NFC-enabled devices may still cause issues, they can be eliminated by proper implementation).

Not least due to the physical characteristics of NFC, it has captured the attention of interaction designers. Arnall (2006) created a graphical language for touch-based interaction with NFC enabled devices (Figure 4.3). Designing with RFID and NFC has been explored by (Martinussen et al., 2007; Martinussen and Arnall, 2009; Nordby, 2010a; Johansson, 2009) and resulted in an approach, taking RFID technology as a design material (Nordby, 2010b). By using the physical properties of this technology, similar gestures as we see in contemporary multi-touch interaction design (e.g. swipe, tap, double-tap, tap-and-hold) or scanning (i.e. holding devices in close proximity), can be used for interaction between (mobile) devices or devices and tags. These physical qualities have been explored and some of them were used in the interaction design of the Connector, which will be described in the next section.

4.4 The Connector: a Tangible Approach

In our previous iteration, three tangible UI approaches were explored. We continued this line of work by developing a new smart object to interact with the semantic connections, which we call the *Connector*. The Connector has similar functionality to the three tangible UI approaches which were explored in the previous iteration, and the Spotlight Navigation device, which is to be described in the next section. Figure 4.4 gives an impression of the design of the Connector.

The following design considerations that were explored in Chapter 3 are also used in this iteration because we found them effective:

- Digital state != physical state: even though the physical representation of the connections' states showed to be easy to comprehend, they introduce unwanted "clutter" in the environment and have practical implications. On-demand access to connections is considered the way forward;
- Using the physical location of the smart objects in their environment for identification (as was employed in the Nodes);
- Using coloured light to indicate the connections' states.

The innovations we want to explore in this iteration as compared to Chapter 3 are:

- Direction of the connections (as was explored in the Nodes);
- Using the physical qualities of the actions involved in making and breaking connections (such as aligning or breaking the alignment between the Interaction Tabs);
- Using the physicality of the smart objects and proximity between smart objects to explore, make and break connections.

The Connector can be used to explore and manipulate semantic connections between different devices in the home environment. It is a hand-held device that identifies devices, by scanning RFID tags that are located on the devices themselves. By holding the Connector

Figure 4.3: A selection of icons for touch-based interactions designed by Timo Arnall. [image source: Arnall, T., (2005) "A graphic language for touch" available from: `http://www.elasticspace.com/presentations/graphic_language_touch_rfid_nfc.pdf`]

on top of the tag, users can explore the connection possibilities that are visualized with lights on top of the Connector. After holding the device in the RFID field for a moment, the device-ID is locked and the other device to be connected can be selected in a similar fashion. With a push-to-click action a connection between two devices can be established. For removing an existing connection, the ring on the lower part of the device should be pulled until it clicks.

4.4.1 Design

The cylindrical shape of the Connector (Figure 4.4) is loosely inspired on that of a loupe or hand lens. The top of the connector is a lens or display, the bottom part is intended to invite for touching other objects. By moving the Connector over a tag, the connection possibilities can be "read" from the top of the cylinder. The display consists of two rings that can light up, each divided into four segments. The connector is designed to afford several actions. You can move it over an object or tag to see whether it is active (scanning). A device or object can be selected by holding the Connector close to or on a tag until the selection sequence is completed (scan- or tap-and-hold). The Connector can be compressed by pushing the top and the lower part together, and it can be pulled, by pulling the lower part and the top part away from one another until it clicks.

When the tag is in range of the Connector's RFID field, it reads the tag and the first (yellow) light segment on top of the Connector will light up, serving as feedback indicating that the Connector recognises the device. After holding the Connector over a device tag for a moment, a sequence starts, lighting up the second, third and fourth segment of the inner ring (Figure 4.6a). This can be seen as feedforward to hold the Connector over the tag until it has been selected and all 4 segments are lit. After the device is recognized and selected, another device may be selected in a similar fashion. Now, the second ring of lights will start lighting up in sequence and one should wait until both rings are fully lit (Figure 4.6b). Removing the Connector from the tag prematurely cancels the selection process. Figure 4.5 shows someone scanning a coloured lamp with the Connector.

When a connection between the selected devices is possible, both rings start flashing green. When no connection is possible, they will turn red. When a connection between the devices you scanned already exists, the rings will turn green (Figure 4.6c). To make the connection, the Connector is compressed by pushing the top and lower part together (direction of the action is the same as pushing a connector in a socket), or by pushing the Connector down on the device it is touching, until it clicks. To remove an existing connection between two scanned devices, the ring on the lower part of the Connector should be pulled (using two hands) until it clicks (direction of the action corresponds to pulling a connector out of a socket). The rings will show a red light to indicate that the connection has been broken. The segments will turn off once the Connector is moved away from the device. Performing the opposite action of what is required to make or break a connection, cancels the procedure. Figure 4.6 shows an overview of the actions, feedback and feedforward given by the Connector. A movie[2] is available of the Connector, as it was used in the smart home pilot (also see Section 4.7)

---

[2] http://youtu.be/D9aEEEZLtFI

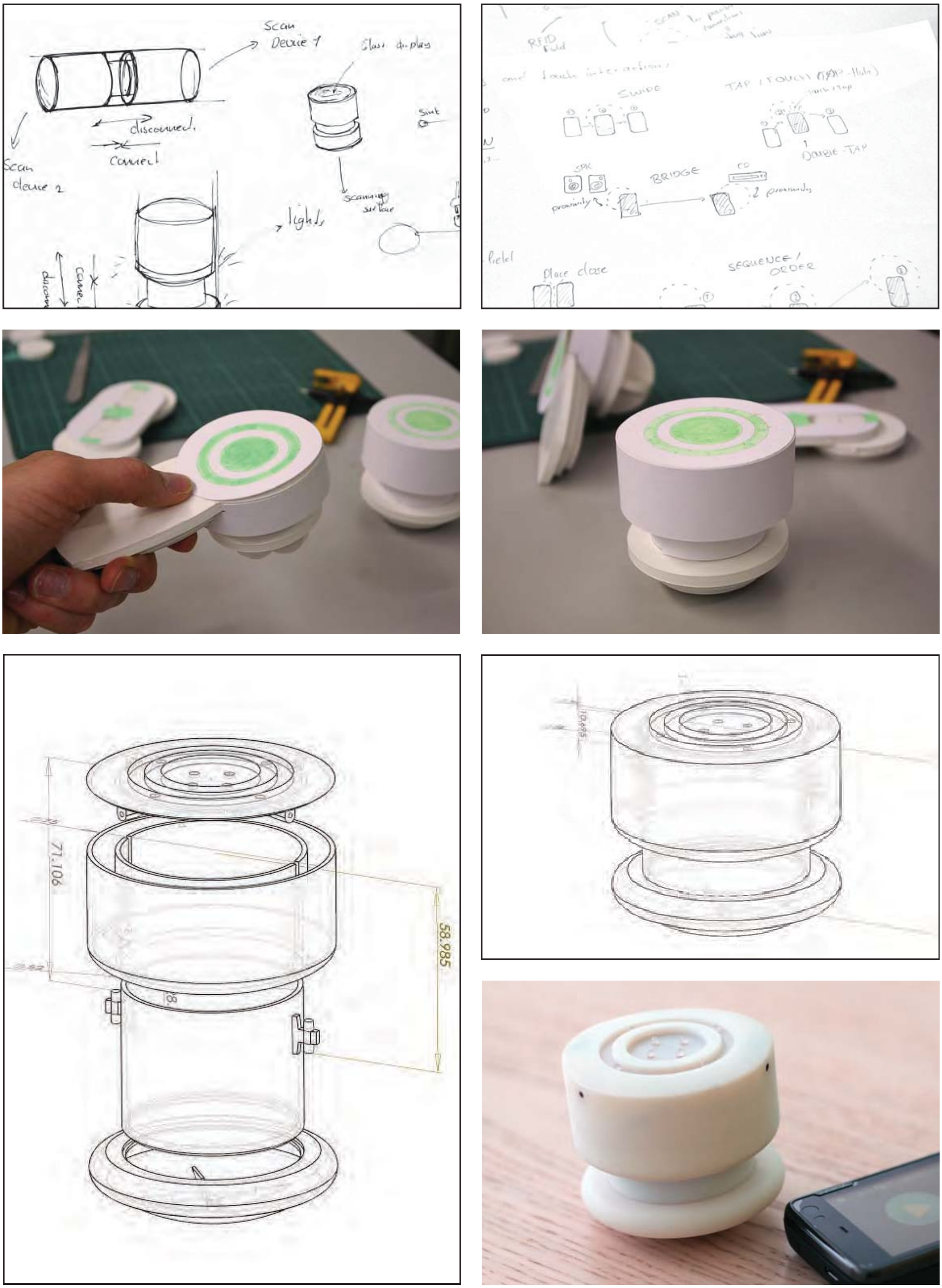


Figure 4.4: Impression of the design process that led to the Connector: (top) early sketches of the connector design concept and an exploration of the action possibilities with an RFID field; (centre) images of cardboard models to explore possible shapes; (bottom) the CAD designs and the 3D-printed result.

Figure 4.5: Image showing the Connector scanning a coloured lighting lamp.

4.4.2 Prototype

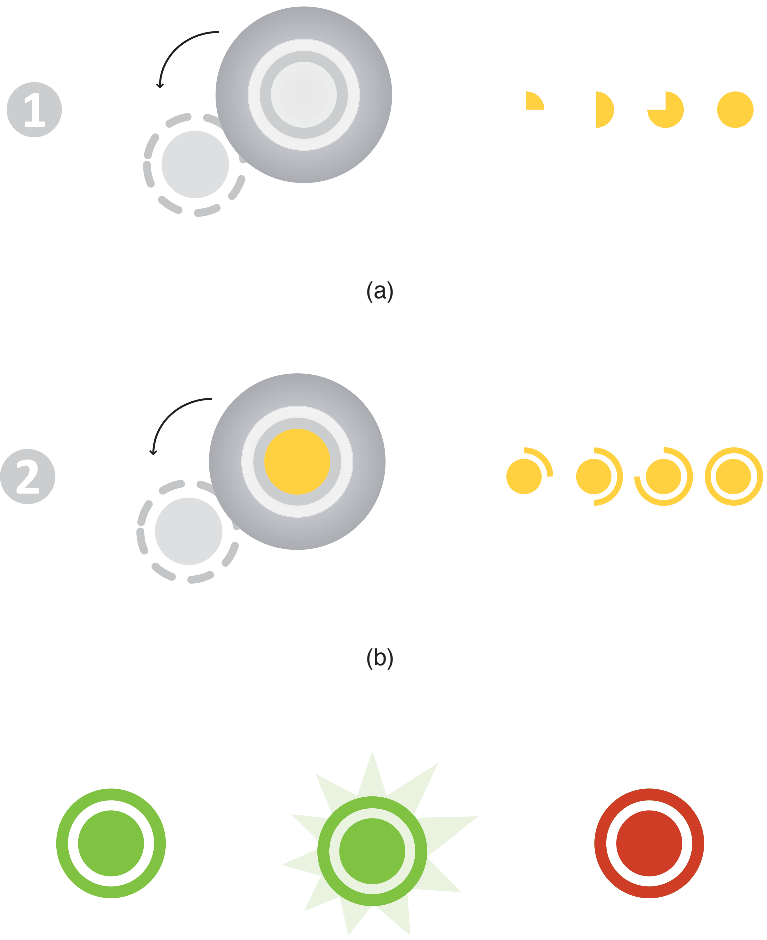
The Connector prototype is made out of four separate 3D printed pieces (as is shown in Figure 4.4, bottom left). The lower part and the top part of the Connector can be moved inward and outward serving as a two-way spring-loaded switch. The prototype packages all the necessary components into one integrated device which is wirelessly connected to a computer using a Bluetooth connection. For a detailed overview of the Connector prototype an its components please refer to Appendix B.

Here we only mention that the Connector contains the following main components:

- Arduino Stamp 02
- Innovations ID-12 125kHz RFID reader
- SparkFun BluetoothMateGold
- 8 bi-colour LEDs
- Switches
- 3.3v LiPo battery (850 mAh)

4.5 Spotlight Navigation: an Augmented Reality Approach

Spotlight Navigation (Figure 4.7) can be used to explore and manipulate connections between smart devices. With Spotlight Navigation, connection information contained in the smart space is projected into the real world, augmenting the real environment with virtual information, making it intuitively perceivable for users. Spotlight Navigation projects icons close to the actual devices in physical space. It allows for the creation of new connections simply by drawing lines between these icons, using a “pick-and-drop” action with a push-button on the prototype (press and hold the button when pointing at one device, move



(a)

(b)

(c)

Figure 4.6: An overview of the actions, feedback and feedforward given by the Connector. (a) scanning first smart object, (b) scanning second smart object, (c) feedback on connection's status.

over the second device and release the button). Additionally the connection possibilities are projected between devices that allow for a connection, by changing the colour of the projected line (while the connection is being drawn) from yellow to green when the line's end is moved over the frame of the targeted device. When a connection is impossible, the connecting line will turn red and disappears as soon as the button is released.

4.5.1 Design

Spotlight Navigation was originally invented by Rapp (Conante, one of the SOFIA project partners) as an intuitive way of accessing large data spaces through handheld digital projection devices (Rapp, 2010). Rather than directly projecting the equivalent of a small LCD display, Spotlight Navigation continuously projects a small portion of a much larger virtual pane or data space. It is the device's orientation that defines which part of the larger pane

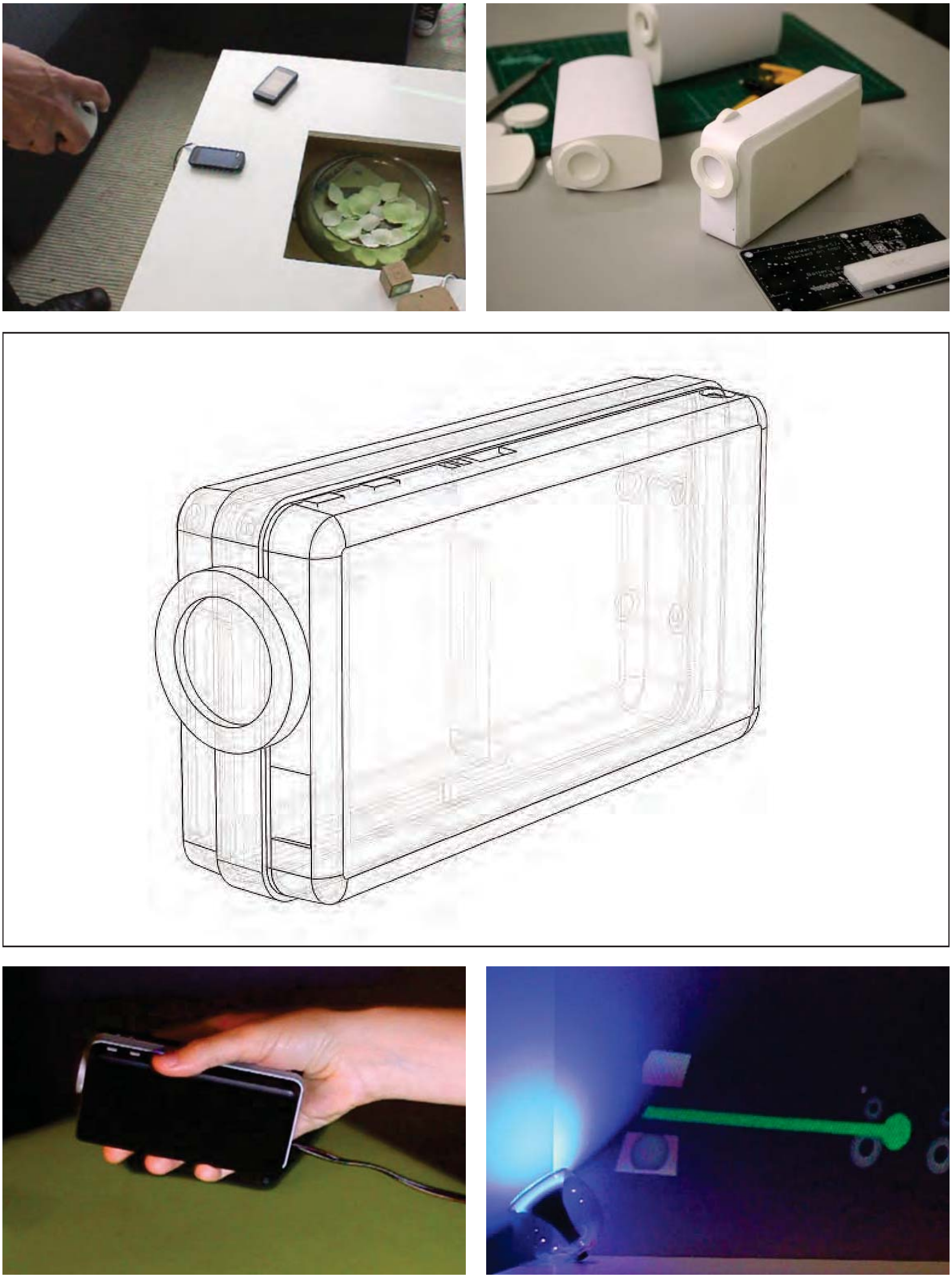


Figure 4.7: Impression of the Spotlight Navigation design: (top) explorations with an early functional prototype and a form study; (centre) Spotlight Navigation CAD design (CAD model by Stefan Rapp); (bottom) Spotlight Navigation prototype and example of projected display when two devices are connected together.

is selected for display. This is done in such a way that the virtual data appears to have a fixed location in the real world. By moving the projector's light spot over the wall, users make portions of the data space visible through intuitive, direct pointing gestures. This intuitiveness stems from the fact that the projected content always stays roughly at the same physical place, regardless of the orientation of the device. It becomes visible, depending on whether it is in the projector's light cone or not. In other words, users have the impression that they are illuminating a part of a conceptually unbounded virtual data space, just as if they would be looking at hieroglyphs on a huge wall in a tomb with a flashlight. As people are familiar with operating flashlights, the operation needs no or little training. When accessing a data space with the device, users can zoom in and out of the data space by using a scroll wheel control, resulting in a pan-and-zoom user interface.

The UI principle of the original Spotlight Navigation device was considered a viable approach to visualise semantic connections, by augmenting physical space with connection data. One of the main goals of our semantic connections approach is to make information and functionalities universally available to every connected smart object. As the data describing the semantic connections (`connectedTo` relationships between devices) is also available in the smart space, using Spotlight Navigation to visualise and manipulate this data was considered a good test case to see whether the different UI approaches could be used interchangeably. Together with Conante, we developed a prototype system that could fulfil this role. Figure 4.7 shows the design and development process, as well as the prototype that was created.

To visualise the semantic connections in physical space, we rely on the symbolic meaning of colour, where green colour means "proceed" and red means the opposite. Using green, yellow and red lines we aim at referring to the "existence" of a connection, the "possibility" of a connection or to indicate that a connection is not possible. Figure 4.7 (bottom right) shows the projection when connecting two devices together.

With Spotlight Navigation, devices are identified by their physical location, relying strongly on *natural mapping*. Connections are created simply by drawing lines between the devices. An erasing gesture with the Spotlight Navigation device pointed at an existing connection, breaks the connection.

4.5.2 Prototype

On a technical level, the operation is achieved through continuously measuring the orientation, and optionally also the position, of the device. Our prototype is using an inertial navigation module, also called an Inertial Measurement Unit (IMU), that directly measures the orientation by means of accelerometers, gyroscopes and an electronic compass.

The Spotlight Navigation prototype is a fully embedded setup integrated into a 3D printed casing (Figure 4.7, centre image). The design of the casing was targeted at getting the smallest possible setup that could run on the integrated batteries. A dummy ring was added to the prototype to strengthen the semantics of a mobile projector. Our current setup consists of the following components:

- OMAP3530 board (IGEP module)
- Pico projector (Microvision SHOWWX)
- Orientation sensor (Sparkfun 9DOF Razor IMU)
- Scroll wheel (with button press functionality)

- Two additional buttons
- Two 3.7v li-ion batteries (Nokia BL5J)

The OMAP3530 processor contains a 3D graphics core (PowerVR) that is capable of rendering the connection visualizations and device icons in real-time. Our current prototype still requires the object positions to be manually configured in space, as it did not contain a camera. By using a camera, as is a possibility for future versions, our ideal is to recognize the identity and physical location of each device, so that it is no longer necessary to align the projected object icon with the location of its associated device. The visualization software is derived from the original PC version of Spotlight Navigation[3] and uses OpenGL ES 2.0 to interface with the graphics hardware. An advantage of the laser-based SHOWWX projector over DLP/LED-based projectors is that there is no need to re-adjust the focus for different projection distances.

4.6 Hardware Infrastructure and software components

In this section, the hardware and software is described which was used to create a prototype system. It implements the interaction model (Section 4.2.1), the Semantic Media and Semantic Interaction ontology (Section 4.6.2) and two interaction approaches (i.e. the Connector and Spotlight Navigation) described in previous sections. The prototype system consists of many different hardware and software components, built on-top of the SOFIA interoperability platform (IOP). The SOFIA IOP is based on a blackboard architectural model (as explained in section 2.3.4) that implements the ideas of space-based computing (Honkola et al., 2010). It consists of two main components: a SIB that acts as a common, semantic-oriented store of information and device capabilities, and KPs, virtual and physical smart objects that interact with one another through the SIB. Various SIB implementations exist that conform to the M3 specification. Smart-M3, the first open source reference implementation released in 2009, was already mentioned. The SIB implementation used in the prototype system is called ADK-SIB (Application Development Kit SIB) and was developed within the SOFIA project.

The ADK-SIB is a Jena-based[4] SIB written in Java and runs on the OSGi (Open Services Gateway initiative) framework. Some modifications were made to the standard ADK-SIB provided by the SOFIA project, such as reasoning support added using the TopBraid SPIN API 1.2.0[5]. Reasoning on information contained within the SIB was performed using SPIN[6] (SPARQL Inferencing Notation).

The KPs, which are software components running on the various devices, communicate with the SIB through SSAP (Smart Space Access Protocol) messages (Honkola et al., 2010) over TCP/IP. SSAP consists of a number of operations to insert, update and subscribe to information in the SIB. These operations are encoded using XML.

The setup was built in an environment that approximates a real-world home environment for these kinds of devices. Two wireless routers were placed in two different rooms, bridged with an ethernet network cable. One router was configured to act as a DHCP server, while the other acted as a network bridge. All components were connected to the network

---

[3] the software was designed by Stefan Rapp

[4] `http://jena.sourceforge.net/`

[5] `http://topbraid.org/spin/api/`

[6] `http://wwwspinrdf.org`

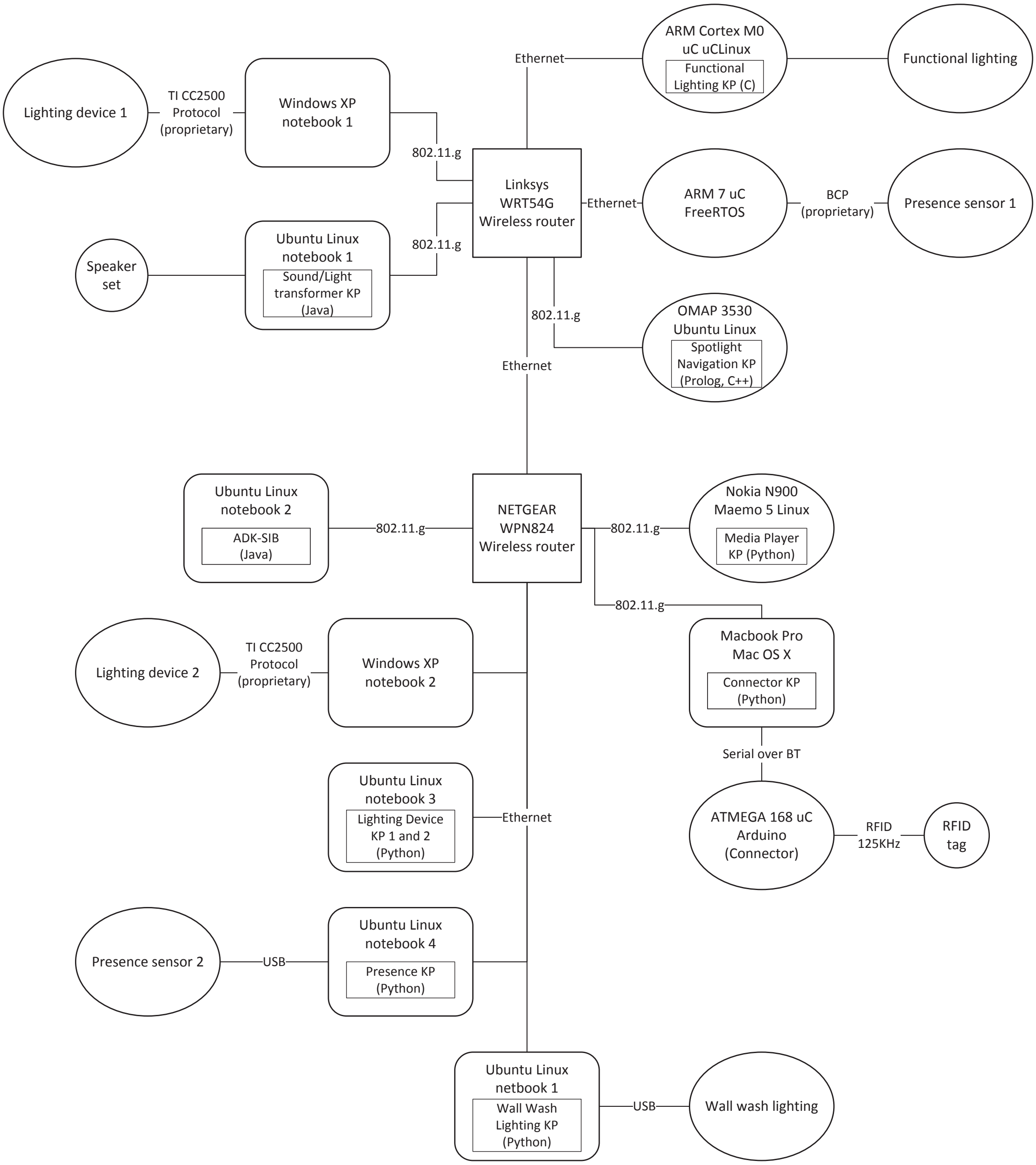


Figure 4.8: A schematic overview of the hardware and software components in the pilot study setup

via 802.11g wireless. In Figure 4.8, a schematic overview of the hardware and software components in the pilot study setup is given.

A *Connector KP* was developed to control the Connector device over the Bluetooth interface, while a *Spotlight Navigation KP* controls the Spotlight Navigation device. The Connector was placed in a living room (downstairs), together with wall wash lighting (controlled by a *Non-Functional Lighting (NFL) KP*), a presence sensor (controlled by a *Presence KP*), a mobile phone with media capabilities (controlled by a *Music Player KP*), and a coloured lighting lamp (controlled by *Lighting Device (LD) KP 1*). The Spotlight Navi-

Table 4.1: System specifications of components used in prototype system

Component	CPU	Operating system	Language
SIB	Intel Core 2 Duo 2.8GHz	Ubuntu Linux	Java
SLT KP	Intel Core 2 Duo 2.2GHz	Ubuntu Linux	Java
Connector KP	Intel Core 2 Duo 2.6GHz	Mac OS X	Python
Music Player KP	ARM Cortex-A8	Maemo Linux	Python
Presence KP	Intel Pentium M	Ubuntu Linux	Python
NFL KP	Intel Pentium M	Ubuntu Linux	Python
Spotlight Navigation KP	OMAP 3540	Ubuntu Linux	Prolog, C++
Functional Lighting KP	ARM M0, ARM 7	uCLinux, FreeRTOS	C
LD KP 1 & 2	Intel Pentium	Windows, Linux	Python, C++

gation device was placed in an upstairs room, together with another coloured lighting lamp (controlled by *LD KP 2*) and a Functional Lighting device with presence sensor (controlled by a *Functional Lighting KP*). A virtual *Sound/Light Transformer (SLT) KP* was developed to transform the audio signals produced by the mobile phone into lighting information that could be rendered by the coloured lamp (for semantic transformers please refer to Section 5.3.5 and (Niezen et al., 2011). The system specifications of each component used in the prototype system is shown Table 4.1. For implementation details of the KPs please refer to (Niezen, 2012).

4.6.1 Knowledge processors

As was explained before, KPs (Knowledge Processors) are virtual and physical smart objects that interact with one another through the SIB. KPs communicate with the SIB through SSAP (Smart Space Access Protocol) messages over TCP/IP. SSAP consists of a number of operations to insert, update and subscribe to information in the SIB. Information is exchanged with the SIB in *triple* format (subject, predicate, object), and this is also the level at which developers of KPs handle inserts, queries and notifications.

To give an impression of how communication with the SIB is handled, we use the SoundLightTransformerKP (SLT KP) as an example. The SLT KP was written in Java and makes use of the Minim audio library for beat detection, in order to generate meaningful lighting patterns that can be sent to the Living Colour lamp. The KP listens for media player events from devices it is connected to, and generates RGB values based on the rhythm of the music. These RGB values are then sent to another computer over TCP/IP which forwards the values to the Living Colour lamp using a propriety protocol.

Part of the event handler that handles subscriptions from the SIB is shown in the following code fragment:

```
/**
 * Handle the events in case a triple we have subscribed to changes
 */
@Override
public void kpic_SIBEventHandler(String xml) {
 String eventId = null;
 String eventOutput = null;

 println("Subscription notification!");
 //Get new triples as triples are added or updated in the SIB
 Vector<Vector<String>> triples = xmlTools.getNewResultEventTriple(xml)
     ;

 if(triples!=null){
  for(int i=0; i<triples.size() ; i++ ){
   Vector<String> t=triples.get(i);
   eventId=xmlTools.triple_getSubject(t);
   eventOutput=xmlTools.triple_getObject(t);
  }
  ...
 }
}
```

Listing 4.1: Code example of the SLT KP software

When the SLT KP is connected to another device KP using a `connectedTo` relationship, we respond to the the interaction events generated by that device. To do so, we subscribe to `connectedTo` events from of all devices that have a `connectedTo` relationship to ourselves (*surroundSoundSystem*). This code fragment is shown below as an example of how subscriptions are created using the Java KP interface:

```
//Subscribe to connectedTo
print ("Subscribe to ConnectedTo\n");
xml=kp.subscribeRDF( null , sc + "connectedTo" , sc + "
    surroundSoundSystem" , URI);

if(xml==null || xml.length()==0){print("Subscription message NOT valid
    !\n"); break;}
print("Subscribe confirmed:"+(this.xmlTools.isSubscriptionConfirmed(xml
    )?"YES":"NO")+"\n");

if(!this.xmlTools.isSubscriptionConfirmed(xml)){break;}
subID_2=this.xmlTools.getSubscriptionID(xml);
print ("RDF Subscribe initial result:"+xml.replace("\n", "")+"\n");
```

Listing 4.2: Code example of the SLT KP software

4.6.2 Ontology development

During the second design exploration, two new ontologies; the Semantic Media ontology and Semantic Interaction ontology, were created. These ontologies extended our ideas

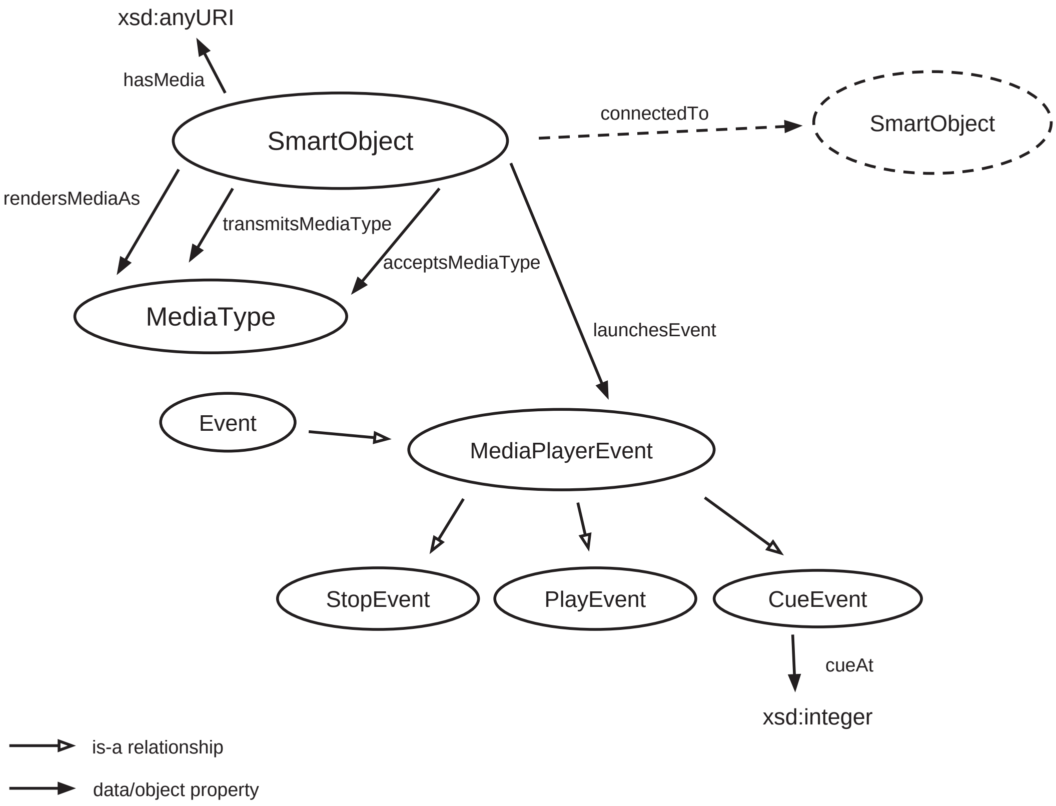


Figure 4.9: Semantic Media Ontology

developed during the first design exploration, and were aimed to enable interoperability between the devices supplied by the different partners involved in the Smart Home pilot[7].

4.6.2.1 Semantic Media ontology

The Semantic Media ontology, shown in Figure 4.9, is a domain ontology that allows for describing media-specific device capabilities and related media content. A mobile device may be described using the following triples (`subject predicate object`):

```
MobileDevice rdf:type SmartObject
MobileDevice acceptsMediaType Audio
MobileDevice transmitsMediaType Audio
MobileDevice hasMedia "file://media/groove.mp3"^^xsd:anyURI
MobileDevice rendersMediaAs Audio
```

The system configures itself through semantic reasoning based on these media type descriptions. A media player event of type `PlayEvent`, that would be generated when the mobile device starts playing music, is described as follows:

---

[7] The Smart Home pilot served as one of the pilots for the SOFIA project and was a joint effort of the participating project partners: Philips, NXP, Conante and TU/e.

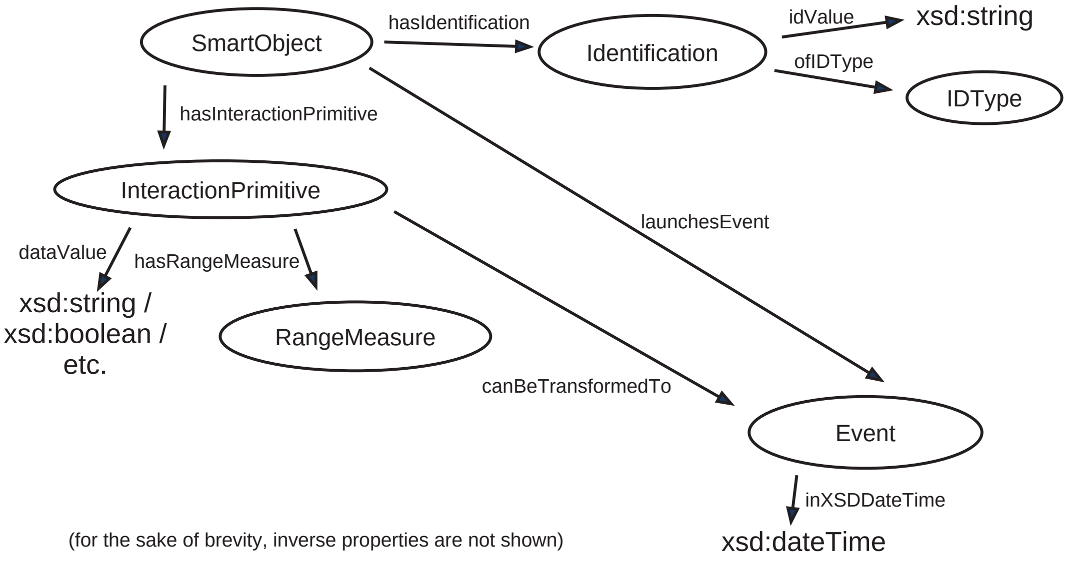


Figure 4.10: Semantic Interaction Ontology

```
event1234-ABCD rdf:type PlayEvent
event1234-ABCD semint:inXSDDateTime "2001-10-26T21:32:52"^^xsd:dateTime
MobileDevice launchesEvent event1234-ABCD
```

Smart objects may be connected to one another using the `connectedTo` relationship. When a device receives an event notification, it first verifies that it is currently connected to the device that generated the event, before responding to the event. Smart objects may be connected to one another directly if there is a semantic match between transmitted and accepted media types. Otherwise a semantic transformer will have to be introduced to transform the shared content, while still preserving the actual meaning of the connection.

A semantic transformer is defined as a service that transforms information shared between devices from one type to another, while preserving the meaning of the information. The concept of a semantic transformer is considered an important part of the theory developed in this thesis and is discussed in more detail in Section 5.3.5.

4.6.2.2 Semantic Interaction ontology

The Semantic Interaction ontology that was developed is shown in Figure 4.10. A device, defined as a `SmartObject`, is uniquely identified by some kind of Identification, for example a IP address and port number, RFID tag or barcode. Identification is discussed in Section 5.3.2.1. Different ID types can be defined as required. Devices can then launch events, for example a media player can generate a `PlayEvent` when music starts playing.

A smart object is described in terms of its interaction primitives. This new concept, as well as the other new concepts introduced in this section, refine the ontology that was developed in the first design exploration. An *interaction primitive* is defined to be the smallest addressable element that has a meaningful relation to the interaction itself. Interaction primitives are described in more details in Chaper 5 in Sections 5.2 and 5.3.2.2.

As an example of how the ontology may be used, we start off by defining a smart object and its interaction primitives. It is only necessary to describe interaction primitives of a device if we use that device's interaction primitive to control another device through the smart space. We can, for example, describe the volume control rocker switch on a smart phone as an interaction primitive:

```
SmartPhone rdf:type SmartObject
PhoneRockerSwitch rdf:type InteractionPrimitive
SmartPhone hasInteractionPrimitive PhoneRockerSwitch
```

We now need to define the properties of the interaction primitive. We start by describing the range measure, or the range of values that the interaction primitive can produce (e.g. the rocker switch can produce `Up`, `Down` or `Neutral` values). The range of values that an interaction primitive can take on is specified using a `RangeMeasure`. These range measures are similar to the measure of the domain set used by Mackinlay et al. (1990), as was described in Section 2.4.6.

Using the range measures, we can then infer which transformations may be used to map the input values to other interaction primitives or events. The ontology could be extended to also describe the different manipulation operators of the interaction primitive, e.g. rotation on the z-axis or movement along the y-axis. Note that the model of MacKinlay et al. has only been applied to GUIs. Similar models for ubiquitous computing have so far not given comprehensive taxonomies of input devices. Our approach of using interaction primitives to describe input devices is an attempt at providing such a taxonomy. The actual data value of the interaction primitive is described using the `dataValue` property. Data values may be strings, boolean values or other datatypes, e.g.:

```
PhoneRockerSwitch dataValue "neutral"^^xsd:string
```

When PhoneRockerSwitch is pressed, the data value is updated with:

```
PhoneRockerSwitch dataValue "up"^^xsd:string
```

This enables other devices to make use of the user input on the`PhoneRockerSwitch`, irrespective of the interaction events generated. In fact, using `Transformation`, it becomes possible to map the physical, generic button presses from interaction primitives like `PhoneRockerSwitch` to specific high-level events like `VolumeUpEvent` or `VolumeDownEvent` using the default transformation `AdjustLevel` as is described in Table 5.1.

By specifying the transformation using the proper OWL 2 semantics, the reasoner should be able to infer which user inputs can be mapped to which specific high-level events. This shows up as a `canBeTransformedTo` property between an interaction primitive and an event. In our example, this means that the following relationship will be inferred:

```
PhoneRockerSwitch canBeTransformedTo VolumeEvent
```

where the "up" data value may then be mapped to `VolumeUpEvent` and the "down" may be mapped to `VolumeDownEvent`, which are both sub-classed from `VolumeEvent`. This prevents situations where arbitrary mappings causes some of the semantics of the interaction to disappear.

4.7 Evaluation

The prototypes based on our two interaction approaches were evaluated in a user study, using the prototype system as described in the previous section. This user study (which served as a pilot for the SOFIA project[8]) was composed of demonstrators made by the different partners in the SOFIA project and was conducted with users in a setting that resembles a real home.

The full setup connects several devices in the home together, constituting an example case of a Smart Home. The pilot was aimed at testing the functionality of the system and evaluating whether people understand the system and the connections between its components. More specifically, the goals of the pilot were to:

- evaluate the mental model users have after interacting with/experiencing the setup, with an emphasis on if and how users understand and conceptualize the relations between the smart objects and the information that is exchanged.
- evaluate the differences in mental models of the relationships between smart objects, depending on the interaction method used (Spotlight Navigation vs. Interaction Tile)
- evaluate the feeling of control users have while interacting with/experiencing the setup
- evaluate perceived user appeal of the main idea behind the setup
- identify perceived user concerns (such as privacy issues, and defined boundaries) about the system
- evaluate the overall functioning (usability and user experience) and stability of the full setup.

In order to collect sufficient insights to improve the system and to verify our ideas, seven groups consisting of three people each, were asked to interact with the system, during which their experiences were recorded. The two interaction prototypes presented in sections 4.4 and 4.5 were part of a larger test setup which was evaluated during a full week of experiments. In addition to the measurement tools that are described in the *measurements* section (4.7.3), we also performed several performance measurements of the software architecture, and held a questionnaire among the developers that worked on the Smart Home pilot. The performance measurements show, among other things, that the software architecture is fast enough to respond to a user's inquiry request (e.g. checking whether a certain connection is possible or already exists, or other user actions). The response time was measured to be well within the maximum allowable value of 2 seconds, as is defined in (Miller, 1968). A full description of the performance measurements, as well as the outcomes of the developers questionnaire can be found in (Niezen, 2012). Here, we focus on the results which are relevant for evaluating the interaction concepts.

During the pilot, users experienced a smart space with various automated and interactive appliances and devices (which we refer to as smart objects). The appliances in the smart space are interoperable, sensitive to changes in their environment and exchange information

---

[8] The pilot was a joint effort of the participating project partners: Philips, NXP, Conante and TU/e. Credits should be given to: Gerrit Niezen, Stefan Rapp, Aly Syed, Riccardo Trevisan, Sriram Srinivasan, Hans van Amstel, Jettie Hoonhout and Jolijn Teunisse for their contributions to the smart home pilot.

with one another. There exist several explicit and implicit relations between the smart objects, of which some can be explicitly viewed or manipulated with the Spotlight Navigation device (available in the study room of the pilot setup upstairs), or the Connector device (available in the living room of the pilot setup downstairs).

4.7.1 Participants

Twenty-one participants were recruited in seven groups of three friends. Selection was based on age (between 20 and 35), availability during the week of the pilot and their mutual friendships. Of the recruited 21 participants that successfully completed the trials, 13 were male and 8 were female. Their age ranged from 23 to 34, with an average age of 28,5. Nine participants were living alone and 11 were living together. The participants were mainly highly educated. Among the participants was one student, three PhD candidates and the remaining participants were employed. Reported fields of occupation included: design (7), medical (1), IT/engineering (4), managerial (4) and services/consultancy (5). The median score of self-report familiarity with interactive systems was 6 on a 1-through-7 scale.

4.7.2 Materials

Figure 4.11 shows a brief overview of the different parts of the system. The experiment took place in two rooms, the study and the living room of the Experience Lab on the High Tech Campus in Eindhoven. The facilities and infrastructure of the Experience Lab were used to set up the prototype system and to collect observation data (video and audio recordings).

In the smart home pilot, media content is shared among several devices in a smart home setting. Music can be shared between a mobile device, a stereo speaker set and a Philips Living Colours lamp that can render the mood of the music with coloured lighting. The music experience is also shared remotely between friends living in separate homes through the Living Colours lamp. Other lighting sources, like the smart functional lighting (FL, fig. 4.11) and the smart wall wash lights (NFL, fig. 4.11) are sensitive to user presence and the use of other lighting sources in the environment. The full setup was built using the SOFIA software platform as described earlier in sections 2.3.4 and 4.6. The smart home pilot follows the following scenario:

> *"Mark and Dries enter their home. The intelligent lighting system detects their presence, switches the lights on, and notifies the smart space about user presence. The decorative wall-wash lights are in turn notified of user presence by the smart space, and turn on. Mark and Dries start listening to music. They would like to try to render the music on a lighting device to also create some visual effects accompanying the music. They query the smart space and find out that the lighting device can render such lighting effects. They make a connection between the music player and the lighting device using the Connector. The light starts being rendered on the lighting device. To put the focus on the lighting device, the decorative wall-wash lights in the room automatically dim down. At the same time, the light pattern also starts being rendered on the remote lighting device, where Mark's sister Sofia can observe the same light effects in her own house.*

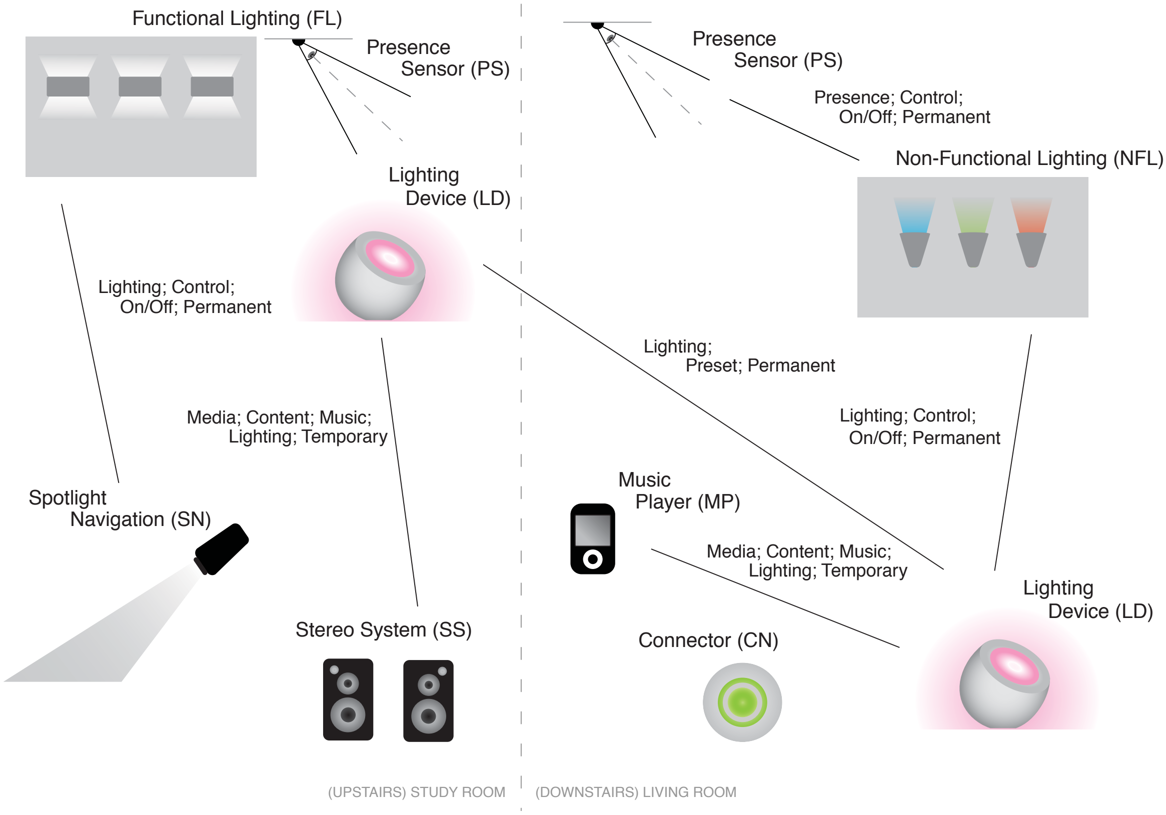


Figure 4.11: The devices and their connections as used in the system

> *At another location: after a while, Sofia is curious and wants to listen to the music that Mark and Dries are listening to. She connects her lighting device to her stereo using Spotlight Navigation, and the same song plays on her surround sound system."*

4.7.3 Measurements

During the pilot, several measurement instruments were employed. Participants were asked to rate the pilot setup on three different scales; the HED/UT scale (van der Heijden and Sorensen, 2003), the Perceived Control scale (Hinds, 1998) and a questionnaire developed by the SOFIA project for internal use (SOFIA Questionnaire). The mental models that users developed during their interaction with the system were recorded using the Teach-Back protocol (van der Veer and del Carmen Puerta Melguizo, 2003), and the participants attitudes towards the system were recorded with a semi-structured interview. Because the HED/UT scale, Perceived Control scale and the SOFIA Questionnaire were targeted at the entire pilot system and not specifically to the semantic connections prototypes, we do not discuss their results in this thesis[9].

---

[9] The smart home pilot was an official deliverable of the SOFIA project. The results of all the measurements were used to report on the: (a) technical feasibility and performance of the pilot setup, (b) the responses of prospective end users to the setup, and (c) the feedback of stakeholders from within the partner organisations. Full reports are available in project deliverables D2.14 and D2.15 (`http://www.sofia-community.org/files/SOFIA_D2+14_D2+15_v1.0.pdf`).

4.7.3.1 Mental models

Mental models were extracted using the teach-back protocol (van der Veer and del Carmen Puerta Melguizo, 2003). Because users' mental models consist of both semantic and procedural knowledge about the system they were interacting with, teach-back questions can be subdivided into "what is?" questions focusing on semantic knowledge, and "how to?" questions focusing on procedural knowledge (van der Veer and del Carmen Puerta Melguizo, 2003). Using such questions, adjusted to our specific situation and research goal, we aimed to extract the semantic and procedural concepts that were relevant for our users.

Participants were asked to explain to the researcher what they thought the system was and what it could be used for, including listing all the components and the relationships and connections between the components they thought made up the system. By asking for the perceived connections and relationships between the components, we aim to gain a better understanding of how users conceptualize the cause-and-effect relationships between their actions and the responses of the various devices in the smart home. This includes the information that is exchanged between these devices. By asking the participants to explain to the researcher how to perform a specific task with the system, we aim to get insights in how well the participants understood the necessary steps and devices involved to achieve their goal. To support and communicate their answers to both types of questions to the researchers (and for recording purposes), participants were asked to make drawings, schematics or use a textual representation.

4.7.3.2 Interview

In order to gain a deeper insight in the things that occurred during the experiment sessions and record the users general opinion, a semi-structured interview was conducted. Using a list of open questions as a structure, participants were evoked to share their experiences with the test setup and think along for possible improvements. During the interview, the researchers also asked questions based on specific behaviour or actions of the participants that they observed during the trial.

4.7.4 Procedure

Participants had already received written information about the experiment together with an official invitation by e-mail. After the participants were welcomed in the Experience Lab and were briefed, they received and signed an informed consent form and were asked to fill out a pre-experiment survey. This survey included demographic questions and a self-report scale of familiarity with interactive systems like (tablet) PC's and smart phones.

The groups of three participants were split up in two groups of which two participants were lead to the living room area to fulfil the role of Mark and Dries (which were using the Connector), and one participant was taken to the study to fulfil the role of Sofia (who was using Spotlight Navigation). These names will be used later to identify the different treatment groups. All participants were introduced to the devices which they had to interact with before the experiment started.

During the experiment, the participants were asked to perform a series of predefined tasks that revealed the functionality of the system. Every participant received these tasks on paper and was asked to think aloud, or for the participants in the living room (Mark and Dries), to share and discuss their thoughts during the whole experiment. After they performed the tasks, they were asked to freely explore the system to deepen their understanding and check their assumptions of its operation. They were allowed to continue this free exploration until they thought they understood the system's operation and would notify the researcher that they had finished. The researchers (one in the living room and one in the study) sat down in the back of the room during the entire session and were available in case anything went wrong.

After interacting with the demo, the participants were asked questions to elicit their mental models and were interviewed. The Mark and Dries characters were interviewed together, and they could openly discuss their opinions and mental models. Some of the participants agreed on their answers and agreed on one drawn representation of their shared mental model. Others disagreed, and created their own representation. The duration of each trial was approximately 50-60 minutes, including briefing, instructions, filling out the questionnaires and the closing interview.

4.8 Results

Of the 21 participants who participated in the pilot, we collected 18 mental models. The teach-back protocol with the Sofia characters (n=7) resulted in seven unique mental models. For the Mark (n=7) and Dries (n=7) characters we obtained 11 mental models, of which three were shared. The mental models consist of drawings (typically one containing the semantic information and one containing the procedural information) and oral explanations. The oral explanations were recorded by audio recorder and written down in transcripts.

The interviews were also recorded by audio recorder and written down in transcripts separately. This resulted in a total of 14 transcripts, 7 for the Sofia characters and 7 for the combined responses of the Mark and Dries characters. In case the Mark and Dries characters had different opinions, the responses were labelled with the characters' names respectively.

The mental model drawings, explanations and interview transcripts were carefully analysed to identify the main themes in the responses. Firstly, the mental model drawings (containing semantic information) were spread over a large wall, organising them by treatment group and grouping them by network structure. Secondly, the drawings were completed with details from the oral explanation and interview transcripts (written down on Post-it notes), that were unclear or not present in the drawings. Thirdly, the mental models were analysed in terms of the themes we identified. Identifying the themes was an iterative process, going through the raw data in detail to identify interesting events and comments, and then cycling through the rest of the data to find similar occurrences. Every occurrence was scored to get a measure of how many times they occurred. Finally, the mental models were abstracted to get a clear view on each of the mental models, and to enable a better mutual comparison. The abstraction step is explained in more detail in Appendix C. The abstraction was done after the previously described analysis steps to make sure that the richness of the raw data was maintained.

The main themes and trends are discussed in the following sections, in the categories: completeness, semantic connections concepts, organisational layout, network structure, semantic knowledge, procedural knowledge and other remarks. The mental models recorded from the Mark and Dries characters are treated as one group.

Table 4.2: Completeness

Component missing	Mark & Dries	Sofia	total
presence detection	8	7	15
non-functional lighting	2	-	2
functional lighting	-	2	2
connection to non-functional lighting	2	-	2
connection to functional lighting	-	3	3

Table 4.3: Semantic connections concepts

Observed concept	Mark & Dries	Sofia	total
transitivity	2	1	3
directionality	6	3	9
priority	2	-	2
permanent/temporary connections	2	-	2

4.8.1 Completeness

One of the themes that came forward during the analysis was the varying completeness across the collected mental models. Out of all the mental models, 15 did not report that presence detection was used; seven out of seven for the Sofia characters and eight out of 11 for the Mark and Dries characters. Of the three that included presence detection in their drawings, one was a shared model and the other two were from the same session.

A few other components of the system that were in the study and the living room were occasionally not included in the mental models. This includes the non-functional lighting (NFL) in the living room, the relation between the NFL and the the Living Colour (LC) light (the NFL would dim down when the LC was active), the functional lighting (FL) in the study upstairs, and the dimming of the FL when the Spotlight Navigation was in use. The NFL was missing in two mental models, as was the connection between the NFL and the LC. These two mental models were from the same session. For the mental models of the Sofia characters, two out of seven missed the FL and three were missing the connection between the FL and the Spotlight Navigation. An overview of the number of missing components per group is listed in Table 4.2.

4.8.2 Semantic connections concepts

During the user experiments some of the participants noticed and discussed interesting networking concepts such as transitivity and directionality. These concepts were also considered in the semantic connections interaction model, but were not implemented in the pilot. Despite the absence of these concepts, participants did intentionally (or sometimes perhaps unintentionally) draw them in their mental models or discuss them. For example, participants drew arrows indicating the direction in which they thought information was exchanged. Among the concepts that emerged from the mental models are directionality, transitivity, priority and the temporary or persistent nature of the connections. An overview of the semantic connections concepts that were mentioned are listed in Table 4.3.

> **Sofia 7:** *"So, their music installation is wirelessly connected to my lamp (draws an arrow in the line) and then when I turn on the music there is a wireless connection between my music and their lamp (draws an arrow in the line). Oh, so this connection (between their lamp and their music) is not needed anymore (crosses out the line)."*

Transitivity was noted in three of the mental models (for an example see quote above) and directionality in nine of them. Directionality was mainly observed from the drawings but was also mentioned during the oral explanations (for an example see quote below).

> **Sofia 1:** *"I understand that the interaction is made in such a way that it seems that I suck up the music with the device from the lamp and put it in the stereo..."*

Two participants indicated a notion of priority in their mental models and were concerned whether one out of multiple conflicting connections would have priority over the others:

> **Mark/Dries 1:** *"I would like to be able to indicate direction of a connection. What if you connect two mobile phones with one lamp?"*

Two persons discussed the persistence of connections, wondering when connections would stop existing (for instance when the person would take a mobile device out of the house):

> **Mark 1:** *"When does a connection end? When you make a connection when you are at home, and you start listening to music in the train, what happens to the connection? Will the lamp also start blinking?"*

Others indicated connections, which they described as a permanent connection, distinctively from the other non-permanent connections (e.g. dashed line versus solid line).

Participants also stretched the idea of making connections between devices towards making connections to other meaningful entities. Making connections to personal belongings of people, items that reminded them of a particular person or other representations were mentioned (also see Section 4.8.6). Moreover, participants indicated (indirect) connections to exist between places (homes) and people, namely between Mark and Dries, and Sofia. For an example, see Figure 4.12.

4.8.3 Organisational layout

Another theme that emerged was a difference between the mental model drawings in terms of organisational layout. We identified three types of organisational layouts (i.e. physical/spatial, logical or a combination of the two which we labelled hybrid layouts) in the way people draw their mental models. Table 4.4 gives an overview of the observed layouts per character group. The majority used a physical/spatial way of describing their mental model, of which we identified eight as being fully spatial (the main structure of the network is based on the physical location of the components) and another eight mental models have what seems to be an arbitrary mapping, using the physical appearance of the components to identify them in the drawing. Some of these representations include spatial information but it is not used as their main structure. We label these hybrid layouts. There are two mental models that show a logical way of representing the network and its components using blocks and labels

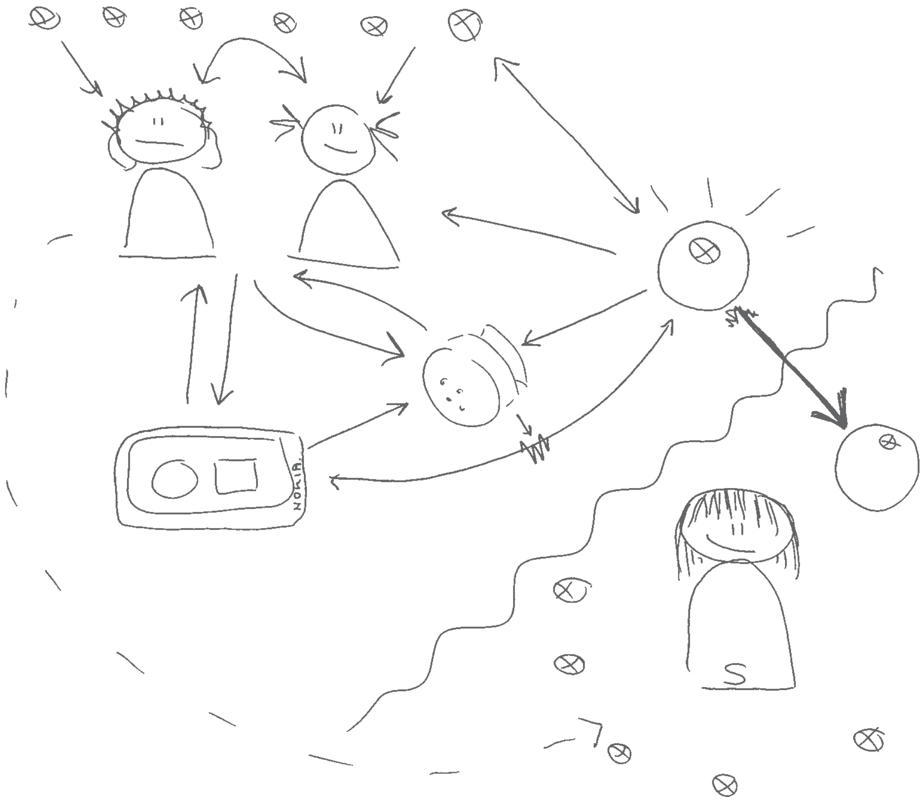


Figure 4.12: Mental model drawing of Dries character 7 showing an indirect relationship between Mark/Dries and Sofia

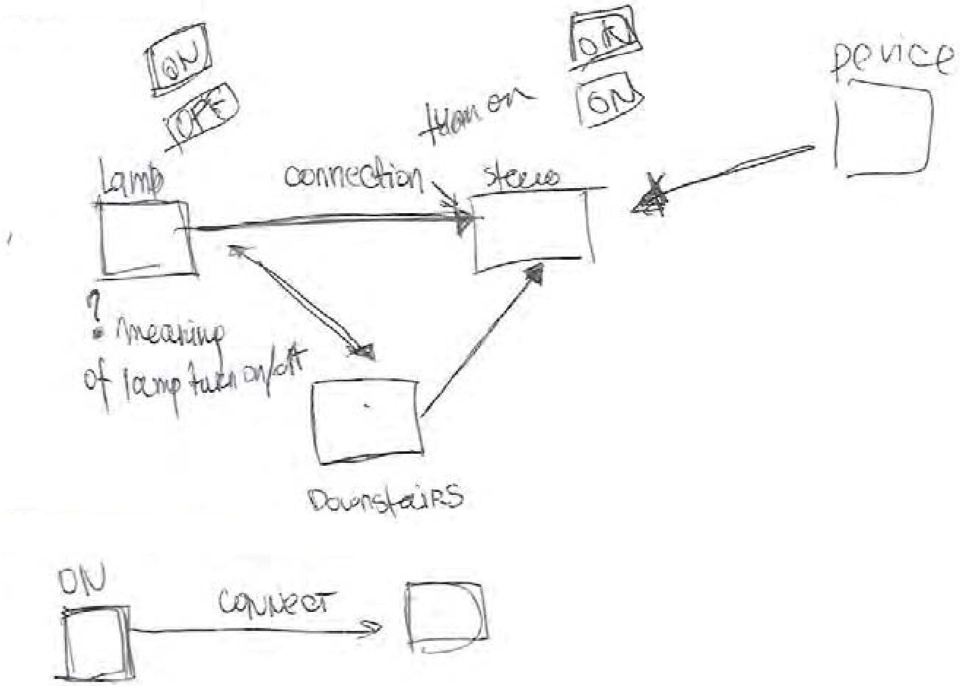


Figure 4.13: Mental model drawing of Sofia character 3 with a logical representation

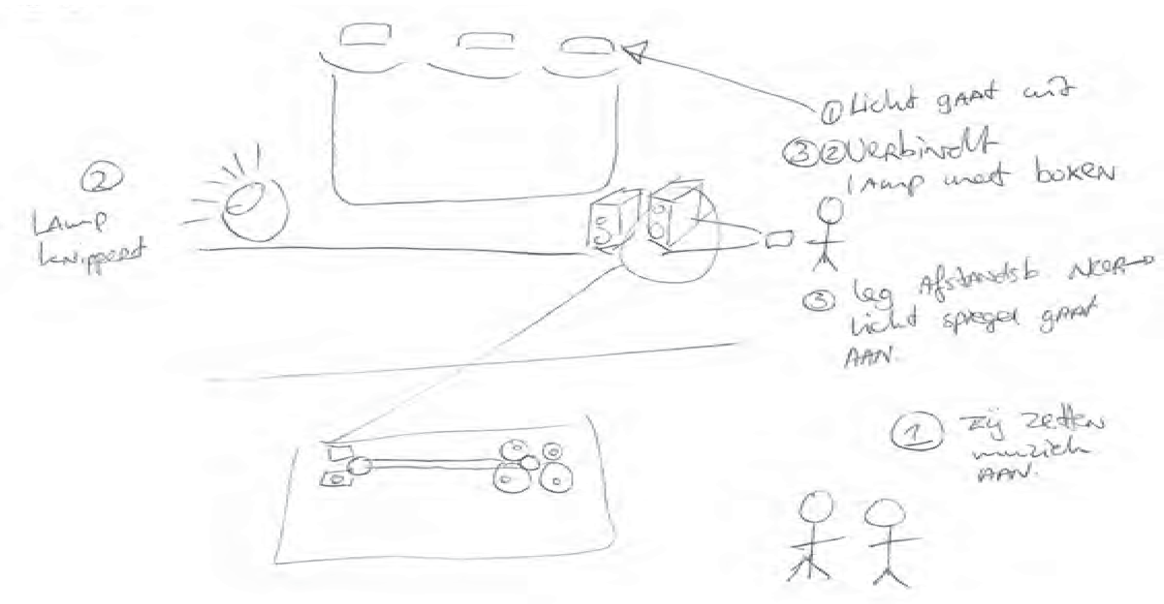


Figure 4.14: Mental model drawing of Sofia character 5 with a spatial representation

Table 4.4: Organisational layout

Organisational layout	Mark & Dries	Sofia	total
spatial/physical	3	5	8
logical	-	2	2
hybrid	8	-	8

to identify the components. Figure 4.13 shows an logical organisational structure, while figure 4.14 shows a spatial one. Figure 4.15 shows an example of a hybrid layout. Similar ways of organising mental models were found in (Poole et al., 2008).

4.8.4 Network structure

Similar to the results that were obtained from earlier user experiments (as are described in Chapter 3), the results of this evaluation also show that participants conceptualise the network's structure differently. For the mental models of the Mark and Dries characters, we observed three main trends in the structure of the networks they drew. We distinguished between network structures that define a central entity (which is close to the actual network architecture), network structures that have a mainly peer-to-peer structure, and a mixed infrastructure which both have peer-to-peer connections and connections going through a central entity (e.g. the Connector object). The participants' oral explanations of the mental model also included references to invisible central network entities. Two participants mentioned (Mark and Dries from one session, with a shared mental model):

> **Mark/Dries 2:** *"There is also a third or fourth device. There has to be another device, otherwise you would run out of battery (from the connector) within an hour when everything would be handled by the connector".*

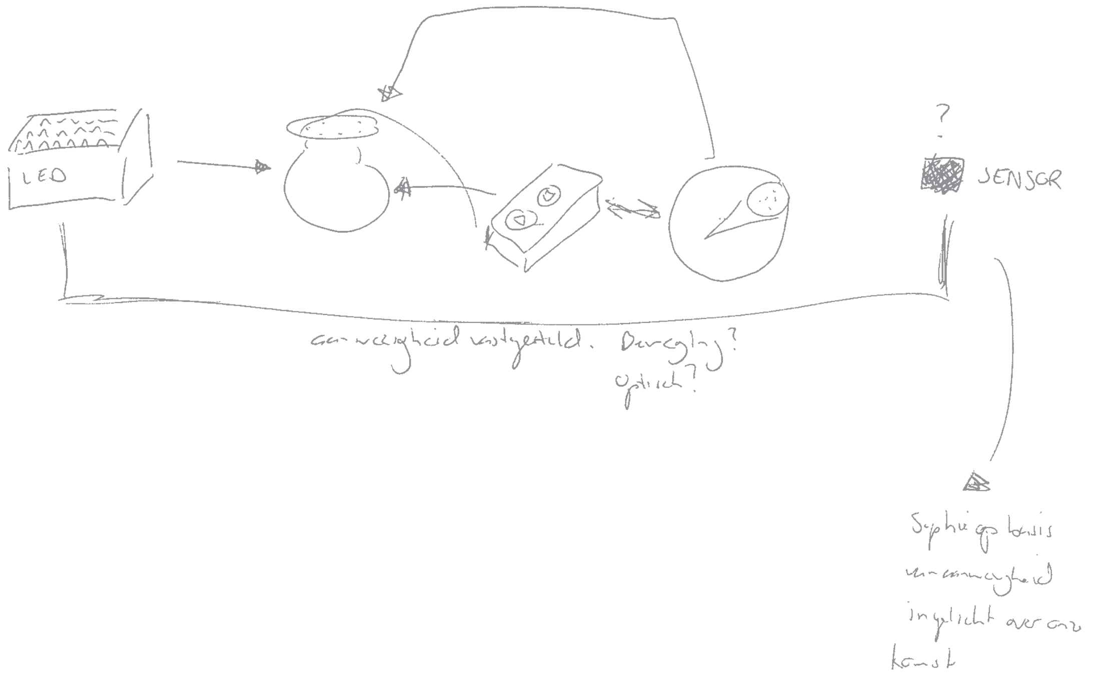


Figure 4.15: Mental model drawing of Mark/Dries character 4 with the Connector as a central entity. Example of a hybrid organisational layout.

and:

> **Mark/Dries 2:** *"You connect the two devices (phone and LC). Device X then knows that they are connected and takes care of the connections. Connections between devices remain. The Connector only indicates what should be connected and makes the connections. Device X then takes care of the connections."*

Two other participants stated (Mark and Dries from one session, with two different mental models):

> **Mark/Dries 6:** *" There must be a central unit. We thought of that when we saw the lighting turn on automatically. So there must be something that controls everything centrally, the Connector only indicates which connections you want to make, the central unit then takes care of the rest."*

All of these mental models of the network are compatible with the actual situation in the pilot. Table 4.5 gives an overview of the observed network structures.

We observed five mental models with a central entity, four with mainly peer-to-peer connections and two with a mixed structure. Figure 4.16 shows a network structure with a central, invisible entity, while figure 4.15 shows a network structure with the connector as the central entity.

For the Sofia characters we mainly observed two different network structures: A daisy-chained, line shaped one (every component connects to one or two others in a serial manner)

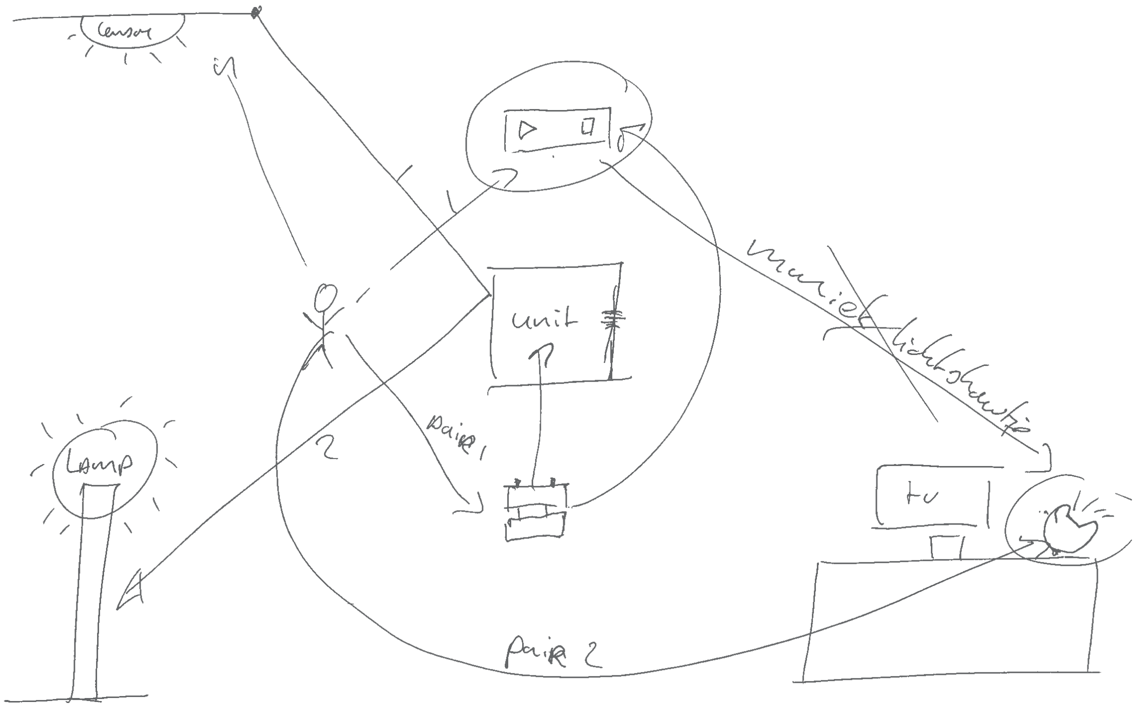


Figure 4.16: Mental model drawing of Mark/Dries character 6 with an invisible central entity

Table 4.5: Network structure

Organisational layout	Mark & Dries	Sofia	total
central entity (centralised)	5	-	5
peer-to-peer (line)	4	5	9
mixed (peer-to-peer & centralised	2	-	2
parallel	-	2	2

found in five mental models, and a parallel structure (where connections had a more parallel nature) which occurred two times. What is interesting to note was that the Spotlight Navigation device was often seen as an entity that was not connected to the network, while the Connector object was in all cases considered part of the network (and in some cases even as being the central entity).

4.8.5 Semantic knowledge

Semantic knowledge that users developed about the system, can be observed from the mental model drawings and their oral explanations. In order to get a clear view on each of the mental models, we made abstractions of the mental model drawings and their recorded explanations. An example of such an abstraction is shown in Figure 4.17. Appendix C shows an example of how a mental model drawing and its oral explanation are interpreted and abstracted. The abstractions show the smart objects, the relationships between them and other relevant entities that were identified by the users. The abstractions were made in such a way that they are directly comparable; eliminating differences due to the different

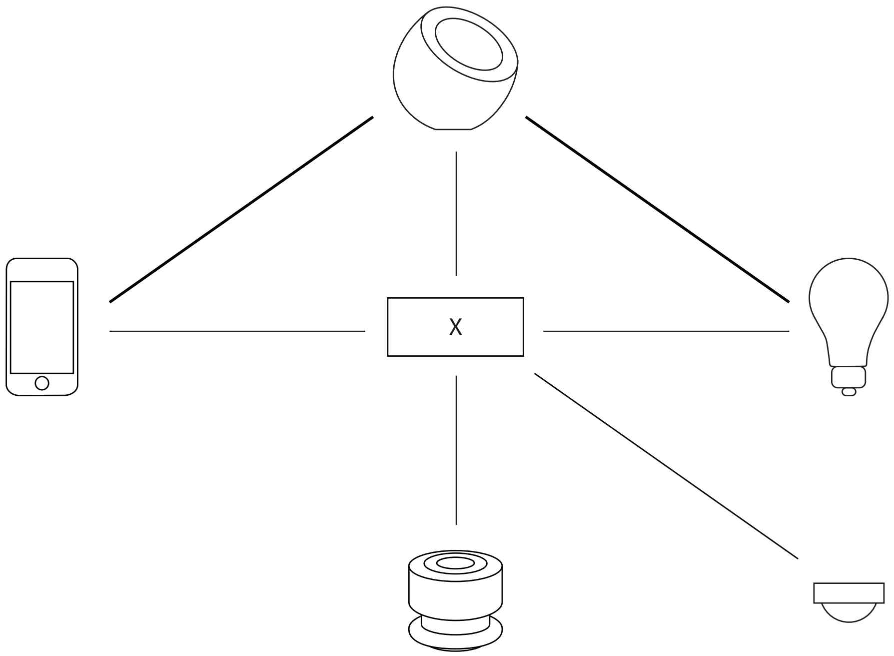


Figure 4.17: Mental model abstraction of a Mark/Dries character 6

drawing styles and different organizational layouts. Arrows, used by participants to signify connections were replaced by lines as not all arrows may have been used purposefully to indicate direction. The cases where participants explicitly talked about the connections as if they were directional, are listed in Table 4.3

Doing so, the 11 mental models collected for the Mark and Dries characters were further reduced to six unique mental models. Appendix D gives an overview of the six different mental models and how often they occurred. For the Sofia characters, the seven different mental models were reduced to six unique mental models, of which two mental models showed full overlap (Figure 4.18). One other mental model showed to be different from the aforementioned overlapping mental models, at only one connection (the connection between "music" and the Living Colours lamp downstairs). Important to note: the Sofia characters did not know the situation downstairs, but guessed the situation based on what they could observe upstairs. Appendix E gives a full overview of the six different mental models for Sofia.

4.8.6 Procedural knowledge

We observed that in both situations (upstairs and downstairs), participants were able to work with the system and were able to explain what steps were necessary to achieve a certain goal. For the Sofia characters, situations occurred where the participant would know how to perform a certain task, but was unable to explain why the system functioned like it did. For example; a participant would understand that dragging a line between the Living Colour (LC) lamp and the speaker set would turn on the music, without understanding where the music came from. Similar situations occurred two times for Sofia.

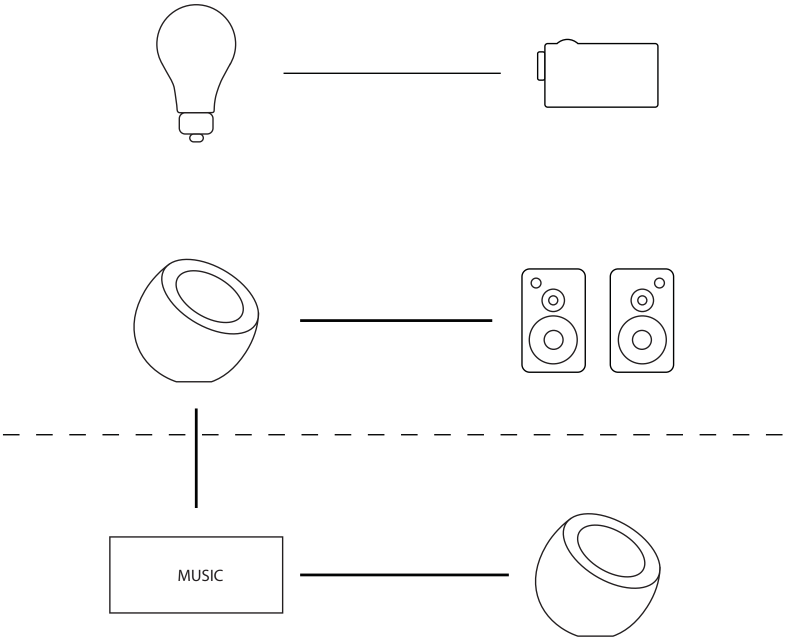


Figure 4.18: Mental model abstraction of Sofia character 5 and 6

> **Sofia 5:** *"But what I do not get The music comes from there (speakers) and I have the impression that the light follows the music, but it already follows it even when the music is not playing yet or is not here yet."*

An important observation was made with respect to procedural information and the feedback, supporting the acquiring of such information during interaction. We observed that some participants explored the system's functioning in a very systematic manner. They observed what the responses of the system were on their actions, made assumptions and acted according to these assumptions in the successive tasks. To confirm these assumptions they relied very much on feedback provided by the system. In few cases where the system would respond slowly, or would behave unexpectedly due to errors or glitches in the system's performance, users responded confused. Sometimes this lead them to re-evaluate their (correct) assumptions. This clearly shows the importance of providing users with the right feedback, at the right time.

To test whether the participants could use their gained knowledge about the system to apply it in a new situation, participants were given a new task. The Mark and Dries characters were asked to set-up the system in such a way, as to share their full music experience with Sofia, who was at a distant location. All the participants were observed to have the necessary procedural knowledge, when they were asked to explain what steps were necessary to perform for making such a configuration. What did lead to interesting result was the final step that was thought to be required to share the experience with Sofia. The Living Colour lamp was considered by the majority to be the device used to share the music with Sofia. How this was done exactly was unclear. Only one participant was convinced that sharing the music experience was done by making a connection between the Living Colours lamp and the music, without any additional actions. The other participants stated that an additional action was necessary to share the music with Sofia. Their solutions for this final step included:[10]

- making a connection between the LC and a phone contact;
- making a connection between the LC and a tagged item (e.g. photo frame portraying Sofia, something personal from Sofia) representing Sofia;
- making a connection between either the LC and a tagged item (representing Sofia) to share only the light, or the music player and a tagged item to share only the music;
- using the smart phone music player to select Sofia from a contact list.

4.8.7 Other remarks

From the interviews we observed several trends that are not directly related to the mental models. Some of which were to be expected, while others were more surprising. Many participants were disappointed in the limited functionality of the current setup. Although the participants were enthusiastic about the ease of which the connections between devices could be made, they were disappointed that they could only control the connections between two devices, despite the fact that there were many more devices and appliances available (especially in the living room e.g. TV, stereo set, other light sources and luminaries):

> **Mark/Dries 5:** *"It would be nice if more devices could be connected with the system. I'd like to explore and make different configurations. When I connect this and that (points at drawing) the system could create a atmosphere to watch TV... if I for instance connect the TV with the lamp. But if I make connections between the stereo set, the lamp and the coffee table, I get a coffee drink atmosphere ... or when I connect the lamp, the music and a reading lamp, the system knows I'm reading a book."*

Most participants were enthusiastic about the "simple way" of making connections. However, they did indicate that they wanted to be more in control of what would actually happen when the connection was made. A Sofia character stated:

> **Sofia 1:** *"I like it that I physically drag the content to the right context, the stereo. This posture is a little limited (referring to sitting on the chair), but I can imagine that I would be able to drag content all across the room, to my laptop for example. This is what I expect of such a device, it has a cable now but I expect it to be wireless in the real system."*

In another (Mark/Dries) group a participant stated:

> **Dries 1:** *"I would like to have more control over what happens when you make a connection ... you could of course just try, there is not much that can go wrong. That is also fun ... trying out things, it would make me curious into what the effects will be."*

---

[10] Similar to the situation for Sofia, the participants playing Mark and Dries did also not get an explanation about Sofia's situation. It was only explained to them that Sofia was at a distant location and was doing a similar experiment at the same time. We decided to leave Sofia's situation unclear to see how they would apply the knowledge of the system (their mental model) they gained during the experiment to solve the task they were given.

Some participants indicated that this lack of control was not crucial, because they figured that the connections could be undone in the same fashion when they did not like the effects of the connection. With regard to the overall functionality participants also indicated that they would like to see more "practical applications" that would make their daily life easier. These remarks were mostly in the direction of concepts known from the home automation domain.

Several remarks were made concerning the user interaction with the Spotlight Navigation and Connector device. For the Spotlight Navigation, remarks were often made about the icons that were projected. The icon for the Living Colour lamp was not always clear to users. Remarks were also made about the (mis)alignment of the icons and the physical devices, and many participants indicated that the icons could be omitted or be replaced by boxes around the physical objects. Additionally, remarks were made about the inaccuracy of the pointing gesture and difficulties in operating the button on the device.

For the Connector device, the low speed of the selection procedure was often mentioned. The effort required to physically select a device was often mentioned as a downside, while others mentioned it as a positive point as it was considered playful:

> **Mark/Dries 2:** *"It is a lot of work to connect the devices every time ... It is, however, easy that you can make the connections between all devices in the same manner. But it is tedious to get up and walk towards the devices every time you want to connect them."*
>
> **Dries 7:** *"The selecting procedure is somewhat slow ... it also has something positive that you have to actually do something to make a connection. It is nice that you get feedback as soon as you use the Connector to connect something ... That the rings show a sequence and turn green when you have selected both devices works well. It is also nice that the lights turn red when you disconnect a connection. Compressing and pulling the Connector also works well for connecting and disconnecting. It is playful, physical."*

Remarks were also made about the limited pairwise selection - participants indicated that they would want to have the possibility to select and connect more devices at the same time.

4.9 Discussion

Spotlight Navigation and the Connector are two user interface approaches to configuring ubiquitous computing infrastructure. Although we cannot directly compare the mental models elicited during the user experiment, which would have asked for a more controlled setting (e.g. having the same setup and having an equal number of participants for both treatments), we did make interesting observations.

The most striking difference between the way users described the setup, was the perception of the users that the Connector was part (if not the central part) of the system, while the Spotlight Navigation was often considered outside of the system. We hypothesise that this is due to the different roles that the Spotlight Navigation and the Connector have in the interaction with the connections. The Connector is used to conceptually "carry" the content between the two devices and, in itself represents the relation between these two artefacts. The Spotlight Navigation is, in contrast, perceived as a "remote control" that visualises the connections in physical space. This might lead the users to conclude that the projected lines are the connections, directly between the devices, and leave the Spotlight Navigation device itself outside of this network.

A clear example of the Spotlight Navigation being considered a remote control, can be seen in Figure 4.18 which represents the mental model of two Sofia characters (Sofia characters 5 and 6). Other examples of the Spotlight Navigation being considered a remote control can be found in Figures E.1, E.3 and E.6 in the Appendix. As discussed in section 4.8.4, Mark and Dries conceptualised the connections as either initiated by or contained within the Connector. Examples of Mark and Dries characters conceptualising the connector as the container of the connections are available in e.g. Figure 4.15 and in Figures D.4 and D.3 in the Appendix. Even though both flavours of mental models (i.e. that of a remote control for Spotlight Navigation and that of a physical representation of connections for the Connector) seem to work fine for users, these differences may have implications for environments where multiple UIs (to semantic connections) are presents that can be used interchangeably. For such environments, mental models of users that conceptualise semantic connections to exist independently from the UI that manipulates them, are preferred. Such a view on connections fits the concept of semantic connections better, as semantic connections exist in the digital domain, and can be viewed and manipulated by any smart object that has the right interaction capabilities and that can access the smart space.

The results show that devices and appliances that automatically act and react to people's behaviour (i.e. incidental interactions) are often not considered in the mental models, compared to the devices and relations that users interact with explicitly. Examples of the aforementioned, can be found in the absence of the devices with implicit interactions in the mental models, e.g. the absence of functional lighting, non–functional lighting and presence detection. Section 4.8.1 describes such cases, and the number of missing devices in the mental models can be found in table 4.2. The absence of such devices is not surprising (as users do not explicitly interact with them), and indicates that designers should pay extra attention to raising awareness of such devices.

However, the results also show that some participants noticed these relations, and incorporated them in their mental models. More interestingly, some of the participants expected that they could manipulate these relationships (e.g. between the presence sensor and non-functional light) in the same way as they could manipulate the other connections. This result is promising and might indicate that people project their experience with one part of the system to the rest of it. This also became apparent when participants started looking for tags on other devices of which they thought could also be connected.

An interesting observation is the rather direct impact of the interaction device's *design* on the mental models. For instance, the design and interaction of the Spotlight Navigation reminded them of a remote control, and consequently they used and described it as such. One participant even thought it was connected to the speaker set because it controlled the music (i.e. making or breaking a connection between the Living Colours lamp and the speaker set started and stopped the music playback).

It also showed that giving proper feedback (the right feedback, at the right time) is crucial when users are developing their mental model (see Section 4.8.6). For example, participants indicated that the LED selection sequence was confusing at first, as they thought lighting up a single LED was feedback for success, waiting for all 4 LEDs to light up was not always understood directly. Another example: feedback was perceived as inconsistent (due to technical restrictions, a pause between selecting the first and second object of about 3 seconds was needed). When users acted faster than those 3 seconds, selecting the second object was unsuccessful. As users were not aware of this 3 second threshold, they looked for other things in their use to account for this inconsistency, which could influence their mental model. When feedback is inconsistent or too late, the participants showed to be confused and change their assumptions of the system's internal working accordingly. Only when feedback is understood correctly (in terms of its meaning), is consistent and is provided at the right time, designers can assume that the users will develop suitable mental models. A performance measurement that was performed during the user experiments (as was already discussed at the beginning of Section 4.7) showed that the software architecture is fast enough to respond to a user's inquiry request (e.g. checking whether a certain connection is possible or already exists, or other user actions). The average response time was measured to be well within the maximum allowable value of 2 seconds, as is defined in (Miller, 1968). Even though this should be fast enough, observations showed indications that faster feedback was expected by the participants.

Another observation was the complexity of conceptions that the participants held about the connections and their properties. Although there was no explicit direction in the interactions or the connections, participants conceived the connections that for instance carried music, to have a direction, travelling from its source to a destination. Directionality was also indicated where one device seemed to control the behaviour of another device. By allowing users to use this sense of directionality in their interaction to establish the connection, we can easily give them more control over the connections' properties.

Transitivity was another, less obvious, concept that emerged from some of the mental models. Transitivity of a connection is a logical property that emerges when a network node A is connected to B, and in turn B is connected to node C. Transitivity then defines A to be connected to C as well. We observed participants to erase connections they indicated to exist before because they "were no longer needed" because of transitivity. And, in another case, worried about (hypothetically) removing a device from the network when it was in a chain of multiple connected devices, because it would lead to removing the transitive connections as well.

Participants also worried about the persistence of the connections. They wondered whether they had to make the connections every time they wanted to use them, or whether they would be remembered. Moreover, participants worried about whether or not connections would continue to exist when they would carry a device along when leaving the house. This shows that the persistence of a connection is a relevant concept for users. The observation that users show an understanding of the semantic connections concepts, indicates that they can potentially be used in the interaction with the connections, to allow for more advanced control over the connections as has currently been implemented.

When the participants were asked to perform tasks they were not familiar with (i.e. Mark and Dries sharing their music experience with Sofia), participants came up with solutions that seem to extend their experience with establishing connections between devices, towards making connections between devices and people (or objects representing people). These observations are promising and confirm the validity of our ideas about semantic connections: they may also exist between devices and people, in-between people and between devices/people and locations.

4.10 Concluding remarks

This chapter introduced a second design iteration, building on results from our first exploration. The concept of semantic connections was developed further and used to create a new use-case demonstrator. In this new use-case, we worked together with project partners to demonstrate interoperability between devices running on different platforms, from different manufacturers. The use-case was built around our ideas of semantic connections, and new interface solutions to interact with semantic connections were introduced.

We have shown that semantic connections, together with the SOFIA IOP provide a platform and therewith the possibility to improve the interoperability among devices. In this context, two prototypes were developed to experiment with tangible and augmented reality approaches to manage semantic connections. Both show their potential in moving the interaction with devices from a device-oriented paradigm towards a more task-oriented paradigm (combining the capabilities of multiple devices to perform a task) with increased interoperability.

During the evaluation, participants responded positive towards the semantic connections, as an easy way to connect devices together. They appreciated the fact that it provides a possibility to interconnect devices that would normally not, however more practical applications were desired. The mental models that were obtained showed interesting trends in perceived network structure. Moreover, mental models showed networking concepts like directionality and transitive connections. These results are promising and confirm our hypotheses that such concepts are important and users can potentially deal with these concepts in their interactions with smart objects and their connections.

Although the exploration described in this chapter was still aimed at exploring the possibilities of our approach, promising results and insights have been achieved already. The results obtained during this evaluation are used to further define our semantic connections interaction model and define a semantic connections theory, which is introduced in Chapter 5. The takeaways from this chapter are summarised below:

- the relevance of the semantic connections concepts: directionality, transitivity and persistence have been confirmed;
- the software architecture and ontologies implementing our ideas on user interaction within smart spaces showed to be viable;
- the pilot demonstrates that interoperability between devices running on different platforms, from different manufacturers is possible;
- the Connector and Spotlight Navigation are both viable UI approaches to interact with semantic connections;
- users show to be able to understand the relationships between smart objects in a smart environment and develop their understanding into a suitable mental model.

Part III

Semantic Connections Theory and Framework

In this final part of the thesis we define a theory and framework, in which meaning, action and function are coupled. The theory and framework are aimed at helping designers and developers of interoperable smart objects to deal with the challenges in contemporary interaction design.

Chapter 5

Semantic Connections Theory

5.1 In this chapter

This chapter introduces a theory to describe semantic connections. Before we introduce our *semantic connections theory*, *interaction primitives* are introduced, which are part of our theory. We continue with describing the semantic connections theory, which grounds our interaction model, and use finite state machines (FSMs) to model and explain the different concepts. To evaluate our theory we implemented it in a new use case scenario, which involves sleep, i.e supporting going to sleep rituals and the waking-up experience. Next we describe the additions and changes we made to the theory based on the implementation, ending in a discussion and conclusion.

5.2 Interaction Primitives

For our semantic connections approach to work well technically, modelling the physical world in the digital domain is equally important (as was briefly discussed in Section 3.8). Product semantics theory and the notion of affordances may be used to help define concepts and model physical properties of devices that have meaning to their users. A product is more than a set of capabilities and functionalities. Where Niezen (2012) has defined ways to describe device capabilities, e.g. (Niezen et al., 2010, 2011), a smart object should also be classified in terms of *what it is*. Purely looking at device capabilities two devices may seem identical based on the descriptions, however, from a user's perspective they may have very different meanings.

But also for interaction elements and controls, we need the ability to describe interactions and objects for interaction in such a way that the physical and contextual meaning

---

This chapter is largely based on:

Vlist, B.J.J. van der, Niezen, G., Hu, J., & Feijs, L.M.G. (2011) *Interaction Primitives: Describing Interaction Capabilities of Smart Objects in Ubiquitous Computing Environments.* IEEE Africon 2011, September 13–15, Livingstone, Zambia.

Niezen, G., Van der Vlist, B. J. J., Hu, J., and Feijs, L. M. G. (under review). Semantic Connections Theory: Enabling Interaction Designers and Developers to Create Interoperable Smart Objects. ACM Trans. Interact. Intell. Syst. (TiiS). 32 pages.

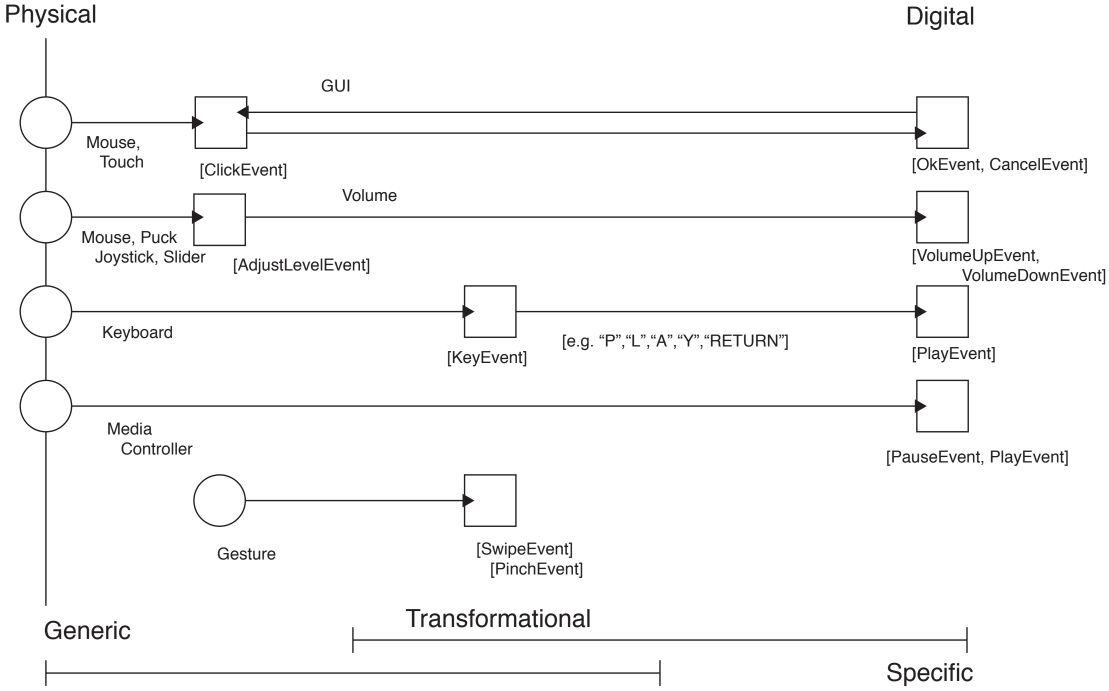


Figure 5.1: User Interface Model

of the interaction is preserved. For this purpose we developed a framework to model the essential elements of an interaction, which we call *interaction primitives*, the smallest addressable interaction element that has a meaningful relation to the interaction itself. We hypothesize that user intentions can be inferred when the interaction capabilities (in addition to device capabilities in general) of devices are described semantically, and the relationships between the devices or the interaction elements of the devices are described using semantic connections.

We model these interactions in a mutually understandable way by describing them in an ontology. The actual ontology is described by Niezen in (Niezen, 2012) and in (van der Vlist et al., 2011). We explain the use of our framework with a volume control example. Interaction primitives are part of our semantic connections theory (as is described in Section 5.3.2.2) and were implemented by (Bartolini et al., 2011).

5.2.1 User interface model

To enable user interaction in smart spaces on the level that was sketched in the introduction of this thesis, the developer community needs to agree on a way of describing the various elements involved in the interactions. These interaction elements or controls (i.e. buttons, touch-screens, sliders etc.) are physical by nature (i.e. they are material parts or at least directly perceivable in the physical world), which means that their physical meaning and some of their physical properties need to be preserved while describing them. Later in section 5.2.2 a simple example will be described, explaining why the previous statement is important, especially when considering that this user interaction data is shared and used by other devices.

Figure 5.1 shows our proposal to model user interfaces in terms of their physical, real-world interaction properties (like position, movement, rotation, force and torque) and their transformation towards the consequences they have in the digital domain (e.g. triggering interaction events, changing states).

On the left-hand side of the figure we plot entities that sense *physical* properties like position, movement or pressure. We consider these properties to be very *generic*, as they do not report a user's intention directly. The inputs first need to be transformed into an intentional event (events that express user intention). This can happen directly, for example pressing a play button, which is transformed into an `PlayEvent`. It can also follow a series of intermediate transformational steps, where a sequence of interaction events (possibly happening on different devices) may be used to capture the user's intent. This sequence of events is then transformed into a single intentional event.

On the right-hand side of the figure we have the *digital* entities that represent the intentional events. We consider these entities to be very *specific*, as they communicate the (assumed) intention of the user's actions directly.

Entities and their relationships in an interaction together form an *interaction path*. The interaction exchange or action between elements in the path is conducted via one or more *interaction channels* along which information or action is communicated (Dubois and Gray, 2008). As an example, a typical interaction path in Figure 5.1 would be:

`Keyboard` → `KeyEvent` → `PlayEvent`

showing that an interaction channel exists between `Keyboard` and `KeyEvent`.

During the transformation from physical to digital, the interaction devices (or their interaction primitives) also move from generic to more specific, where generic user interfaces start off very generic and stay generic or transformational (meaning they have been transformed but still need further transformation). This means that such interaction devices or interaction primitives can still be transformed into many different events or states. When an interaction primitive travels from generic to specific with a single transformation (like the media controller buttons) it means that that these interaction primitives are very specific UI elements (i.e. have one single function). An example of such an interaction primitive would be a hardware button with a specific label that is only used for one function. As another example, consider a gesture; i.e., a (less) generic interaction primitive that transforms from a physical movement that is sensed in a certain way, to the digital representation of that gesture, being a "pinch", "swipe" etc. The pinch and swipe are still considered transformational because they still need to be transformed further to result in a certain interaction event. However, in the initial transformation, some meaning is preserved (i.e. the physical characteristics of the gesture). These characteristics limit the number of actual events the gesture can still be transformed into, e.g. a "swipe right" gesture should not be transformed into a "navigate forward" action, as this is the way we usually navigate backwards. Table 5.1 shows some of the possible transformational events we consider to be applicable to smart home environments in which multimedia and lighting devices are connected for certain applications.

Although describing user interaction capabilities of devices according to the user interface model is valid for user interaction in general, it is specifically relevant when we consider the notion of a smart space through which this interaction data can be exchanged. To

Event	Entity this event can be performed on
AdjustLevel	Volume, Lighting
switchOnOff	Lighting, any SmartObject
Navigate	Playlist, Menu, SequentialData
Undo/Redo	Any interaction event
Stop/Start	Application, Media
DragAndDrop	Media
Query	Media, other events

Table 5.1: Examples of transformational events in a smart environment

achieve this, all events that need to be shared must be modelled in a mutually understandable way. A good way of modelling them would be an ontology, as is described by Niezen in (Niezen, 2012) and in (van der Vlist et al., 2011).

When modelling, only that which is meaningful to be shared with other devices is considered. It is not necessary to describe interactions that are internal to the device and that are not shared. An accelerometer, for example, may be modelled as a separate device, sharing the raw accelerometer data to be used by other devices. However, when integrated into smart phones, the accelerometer's data can often be abstracted as part of an interaction path, e.g. to only share the orientation of the device, or specific gestures measured with the accelerometer. In this case, the raw values may only need to be available locally on the device, to be used by the developers of other device-specific applications.

5.2.2 Example: Volume control problem

To underline the relevance of describing a device's interaction capabilities according to the user interface model we proposed, let us consider an example. We take a fairly simple example of sharing a "rocker switch" or a group of two buttons that can often be found on smart phones or media players to control the volume of various (local) audio sources (e.g. music, ringer, movie, etc.). It is likely to assume that, when considering a scenario where many devices are interconnected and user interaction information can be shared, controlling the volume of music playing remotely with the rocker switch on your smart phone will be desirable.

Rocker switches to control volume come in different versions, but to keep it simple, we consider rocker switches that are labeled with (+) and (-) (like we find on the iPhone 4) and versions that are not labelled (like we find on the Samsung Galaxy S and iPhone 3G(S)). The way the labeled buttons appear in the physical world, prescribe that the part labeled (+) should be mapped to `adjustLevel` "up" or directly to `VolumeLevelUpEvent`, and `adjustLevel` "down" or `VolumeLevelDownEvent` for the part labeled (-). For the `AdjustLevel` transformation also see table 5.1.

For the unlabeled versions it is more difficult, as just choosing an arbitrary mapping might result in mappings that are not expected by a user. If we take for instance the unlabeled rocker switch of the Samsung Galaxy S, we find that they mapped the top part of the rocker switch to "volume up" and the lower part to "volume down" (when holding the phone in upright/portrait position; Figure 5.2a). When using the phone in landscape position, the right button is mapped to "volume up", and the left to "volume down",

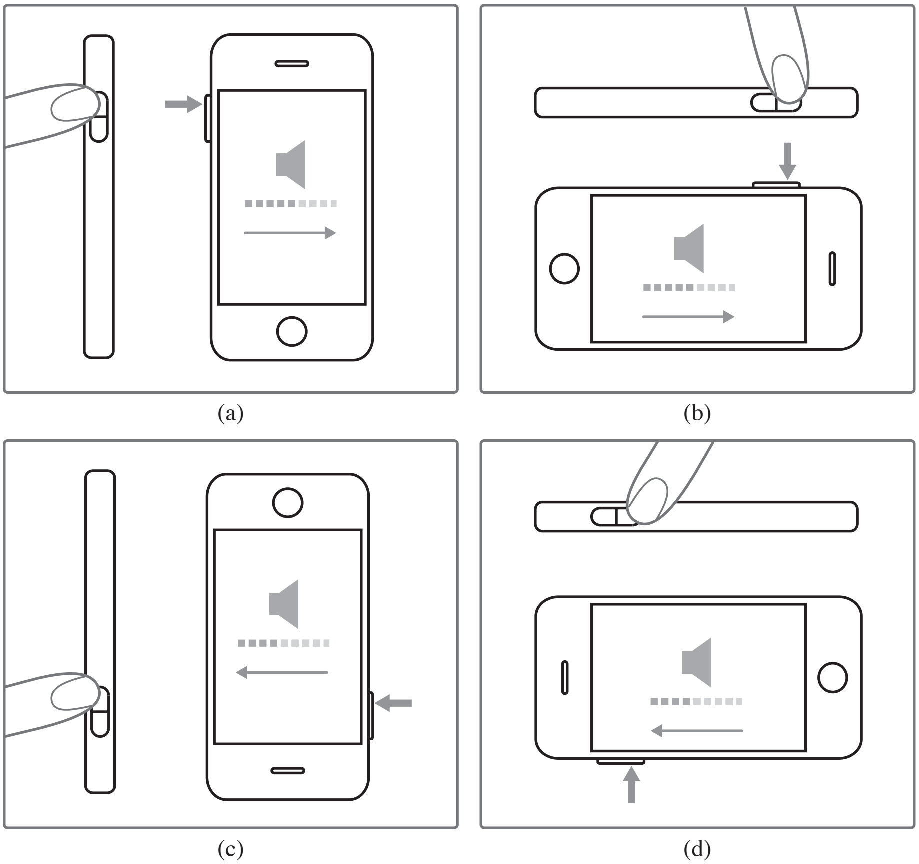


Figure 5.2: Volume adjust problem: (a) portrait position, volume up; (b) landscape position volume up; (c) portrait position (rotated 180°), volume down; (d) landscape position (rotated 180°), volume down.

which still makes sense (Figure 5.2b). When rotated 180 degrees, from both portrait and landscape position, the chosen mapping becomes a little confusing as the mapping now seems reversed (Figure 5.2c/d). Now, suddenly, the mapping that was chosen appears less natural. Interestingly, the designer/developer of the phone and its software decided to make the screen and its content *context aware* i.e. matching the orientation of the phone, but not the volume controls.

With something as trivial as controlling the volume, this is not much of a problem. But when the concepts behind the mappings become more complicated, issues like those just described become more problematic.

This simple example shows the importance of *semantic mapping*, where leaving out physical properties that are meaningful to a user may result in unexpected behaviour. What we propose as an interaction primitive for an unlabelled rocker switch, is thus not only mapping the rocker switch up and down positions, to volume up and down, but also taking the orientation into account. Thus making the mapping context (orientation) dependent, and sharing the more meaningful, contextualized mapping.

5.3 Semantic Connections Theory

At the start of chapter 3 semantic connections were defined for the first time, and they were redefined in chapter 4. In this section we combine our previous definitions, observations

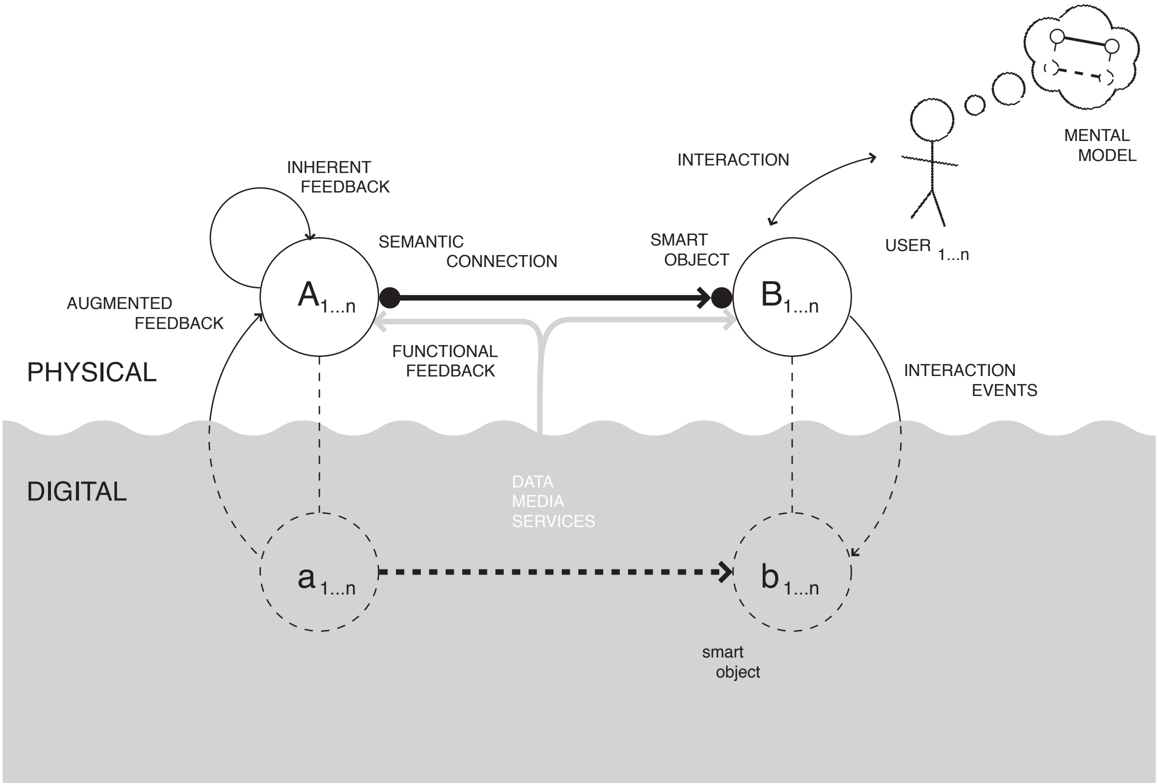


Figure 5.3: Semantic connections user interaction model

from the experiments and existing user interaction models (as are described in Section 2.4) to define a semantic connections theory. When we look at many of the existing interaction models, they offer models and a vocabulary to describe user interaction. However, these models are mainly focussed on describing interactions with a single device. Our theory may be used to analyse, i.e. understand, explain and predict what happens with information and interaction events when devices are interconnected and form an ecology of smart objects.

Before we extensively introduce our semantic connections theory, we first introduce the semantic connections user interaction model.

5.3.1 Semantic Connections Interaction Model

A user interaction model for semantic connections is shown in Figure 5.3. It describes the various concepts that are involved in the interaction in a smart space and shows how these concepts work together. The interaction model was inspired by the Tangible Interaction model (MCRpd) by Ullmer and Ishii (Ullmer and Ishii, 2000) which in turn was based on the Model View Controller (MVC) model, which are described in Section 2.7.

We distinguish between the physical and digital parts of the user interaction. A user does not directly observe what is happening in the digital domain, but experiences the effect it has in the physical world by interacting with the various smart objects, and the semantic connections that exist between them. In doing so, users create a mental model of the objects/system they are interacting with, which only partly (or not at all) includes the digital part. When a user interacts with a smart object connected to the smart space, he/she senses feedback and feedforward, directly from and inherent to the controls of the device (inherent feedback), digital information augmented onto the physical world (augmented feedback) and perceives the functional effect of the interactions (functional feedback). The terminology, inherent, augmented and functional feedforward and feedback is adopted from (Wensveen et al., 2004) as was introduced previously in section 2.10. The user actions in the physical world are transformed into interaction events and device state changes. These interaction data in terms of user intentions is stored in the smart space[1]. Note that in Fig. 5.3 the arrow on the left shows the feedback (or output) of the smart object, and on the right it shows the input (the interaction event). The reason for this difference between the two sides of the figure is to avoid repetition, as they may and in most cases will, occur on both sides.

5.3.2 Smart objects

Smart objects are the devices that are connected to the smart space, enabling them to share information with one another.

Definition 5.3.1 Smart Object: A smart object is a device with both computational and network communication capabilities that can be uniquely identified in both physical and digital space.

According to our definition, an NFC-enabled smart phone is a smart object. A WiFi-connected lamp is also a smart object, given that it can be physically identified by e.g. proximity based on signal strength, RFID (Radio-Frequency Identification) or similar technologies [2].

In terms of our definiton, an RFID-tagged light switch is not a smart object. A software agent running on a GUI, even though it is visually perceivable (e.g. Microsoft Office's Clippy[3]) is not a smart object. Despite its apparent physical existence (physically and digitally identifiable), it is mediated by a computer. In such cases the computer is considered the smart object and not the agent running on it, as the agent is not *primarily* a physical entity. In the following subsections we describe the concepts specific to the smart objects themselves.

5.3.2.1 Identification

For semantic connections to be realized, a smart object needs to be uniquely identifiable in both the physical and the digital domain. A device that is tagged with an RFID tag can be identified in the physical space, and the unique identifier read from this tag must also be linked to the digital representation of the smart object. Near Field Communication (NFC)—using a near field channel like RFID or infrared communication—is an interesting case, because it allows for direct manipulation of wireless network connections by means of proximal interactions (Rekimoto et al., 2003).

---

[1] the notion of smart space means that data is stored centrally in an information broker, and can be accessed by the various smart objects in the smart space. For more information on these concepts please refer to (Niezen et al., 2010).

[2] Note that our definition does not define where a KP (SOFIA terminology; section 2.3.4) should run. For simple smart objects such as a smart light bulb, the actual KP that is communicating to the information broker may be a virtual entity running somewhere on the network.

[3] `http://en.wikipedia.org/wiki/Office_Assistant`

Of course, there are many ways in which a smart object may be identified. An IP address makes a device easily identifiable in the digital space, but it is difficult to create a physical representation of this identity. Consider the case where IP addresses are printed on stickers and stuck on computers to make them easier identifiable to IT service personnel.

5.3.2.2 Interaction primitives

We first defined interaction primitives as a way to describe the user interaction capabilities of smart objects in ubiquitous computing environments in (van der Vlist et al., 2011), based on the work of Foley, Card and others, which was introduced in section 2.4.6.

Definition 5.3.2 Interaction primitive: An *interaction primitive* is defined to be the smallest addressable element that has a meaningful relation to the interaction itself.

The key on a keyboard labelled "A" is an interaction primitive, as pressing it not just changes the key's `Up` state into a `Down` state, but carries the meaning to *produce* a character "A". A gesture `SwipeLeft` on a touchpad is also an interaction primitive, as this is the *smallest* addressable element to still have meaning. Describing the input on a lower level would cause it to loose its meaningful relation to the interaction (i.e. in terms of its physical qualities such as direction, dynamics etc. and their relation to the functional result[4]). A touchpad itself is not an interaction primitive but rather an input device. Data obtained from the touchpad, annotated with its meaning, can be an interaction primitive. A GUI is not an interaction primitive, but a GUI element can be.

Interaction primitives are described in terms of their physical properties that are meaningful to a user. For example, an unlabelled button should not just be represented in terms of its `On/Off` states, but also whether it is in a `Up` or `Down` state. This enables the mapping of physical, generic interaction primitives like a rocker switch to specific high-level events like `VolumeUpEvent`.

An interaction primitive also has a range measure, that describes the range of possible values that it can take on. This makes it easier to determine if and how they can be mapped to specific *interaction events* (described in more detail later in Section 5.3.4).

Interaction primitives and interaction events together form an *interaction path* (Dubois and Gray, 2008). As an example, a typical interaction path would be:

$$\texttt{VolumeSliderLeft} \rightarrow \texttt{SlideLeftEvent} \rightarrow \texttt{VolumeDownEvent}$$

where the `VolumeSliderLeft` is an interaction primitive mapped to `SlideLeftEvent` interaction event. Based on the available context information (e.g. orientation of the slider or smart objects the slider is connected to), this can in turn be mapped to a more specific `VolumeDownEvent`.

When modelling, only that which is meaningful to be shared with other devices is considered. It is not necessary to describe interactions that are internal to the device and that are not shared.

---

[4] I acknowledge that feeling and aesthetics of an interaction are also important, but for now such qualities are not modelled, if at all possible

5.3.2.3 Device states

When describing the capabilities of a smart object, not only the interaction capabilities are important, but also the device states. For users, device states may be hidden or they can be perceived as *modes*, e.g. a media player playing back music (or not), a lens-cap covering a camera's lens or a part of a GUI currently displayed on a screen. With Universal Plug-and-Play (UPnP), two types of documents are used to describe a device and its capabilities. A *device description document* describes the static properties of the device, such as manufacturer and serial number (Jeronimo and Weast, 2009). UPnP describes the services that a device provides in *service description documents*. These XML-based documents specify the supported actions (remote function calls) for the service and the state variables contained in the service.

The state variable descriptions are defined in a similar way to how we define our interaction primitives, with a unique name, required data type, optional default value and recommended allowed value range. The UPnP Forum has defined their own custom set of data types, with some similarity to the XML Schema data types used by OWL 2 (that we used for our implementation as described later in Section 5.4.1.1). As an example, consider a state variable to describe the darkness of a piece of toast, where `ui1` is defined as an unsigned 1-byte integer:

```
<stateVariable sendEvents="no">
<name>darkness</name>
<dataType>ui1</dataType>
<defaultValue>3</defaultValue>
<allowedValueRange>
<minimum>1</minimum>
<maximum>5</maximum>
<step>1</step>
</allowedValueRange>
</stateVariable>
```

The `sendEvents` attribute is required for all state variable descriptions. If set to `"yes"`, the service sends events when it changes value. Event notifications are sent in the body of an HTTP message and contain the names and values of the state variables in XML.

Let us consider these device states in terms of user interaction. There are four key concepts in an interaction - actions, states (internal to the device), indicators, and modes (physically perceivable device states) (Thimbleby, 2007). The user performs actions, which change the device state, which in turn control indicators (augmented feedback). Users may not know exactly which state a system is in; they only know (if at all) what mode it is in. If we want to fully capture the capabilities of the device, we need to specify the device states, the transitions between these states, the interaction primitives which can cause these state changes, as well as the default and current states of the device. When this device is then connected to another device, we also need a way to describe the connection between these devices.

5.3.3 Semantic Connections

Semantic connections is a term we introduced (van der Vlist et al., 2010b,a) for meaningful connections and relationships between entities in an ecosystem of interconnected and interoperating smart objects.

Definition 5.3.3 Semantic Connection: A semantic connection is a relationship between two entities in a smart environment and focusses on the semantics—or meaning—of the connections between these entities.

The connection between the remote control and a wirelessly controllable (on/of or dimmable) light bulb (e.g. Philips' Living Whites lights[5]) is a kind of semantic connection. The connection exist between two smart objects that can be physically identified and connected through physical proximity, its communication technology is unknown to its user and remote control and light are conceptually linked by users based on the perceived behaviour. Another example is the Nokia 360° speaker system[6], where music can be streamed wirelessly using an NFC-enabled smart phone. By touching the phone to the speaker, a connection is created that conceptually carries the music from the phone to the speaker. The WiFi connection between a smart phone and a WiFi router is not a semantic connection, as the connection in itself has no clear meaning. A USB cable is also not considered a semantic connection.

Semantic connections make up a structural layer of inter-entity relationships on top of the network architecture. The connections can be the real, physical connections (e.g. wired or wireless connections that exist between devices), or conceptual connections that seem to be there from a user's perspective. The term "semantic" refers to the meaningfulness of the connections. We consider the type of connection, which currently often has the emphasis when interconnecting devices (e.g. WiFi, Bluetooth, USB) not to be the most relevant, but what the connection can do for someone—its functionality (e.g. stream music, share files)—even more. Semantic connections exist in both the physical world and the digital domain. They have informative properties which are perceivable in the physical world. However, these physical qualities might be hidden by default, and only visible on demand by means of a mediating interaction device. The digital counterparts of semantic connections are modelled in an ontology. There may be very direct mappings, e.g. a connection between two real-world entities may be modelled by a `connectedTo` relationship between the representations of these entities in an ontology. Our use of ontologies is described in more detail in Section 5.4.1.1. Semantic connections have several properties, which are explained in the following subsections.

5.3.3.1 Directionality

Semantic connections have a specified direction, or are bidirectional (symmetric). Smart objects that are connected should then be identified as sources and/or sinks[7]. Directionality

---

[5] `http://www.ecat.lighting.philips.com/l/lamps/compact-fluorescent-integrated/energy-saver-specialties/living-whites-esaver/21238/cat/`

[6] `http://nokia.ly/ingpDX`

[7] We use the terminology source and sink analogous to sender and receiver, with the distinction that sender and receiver imply actively sending or receiving information.

may intentionally be specified by user action, or it can emerge from the capabilities of the smart objects e.g. connecting a source to a sink will automatically create a connection going from *source* to *sink*.

5.3.3.2 Transitivity

When connections have directionality and multiple devices (i.e. a minimum of three devices) are involved, devices can also act as bridges, transferring the connections due to transitivity, e.g.: If a music player is connected to speaker A, and speaker A is connected to speaker B, speaker A acts as a bridge between the music player and speaker B.

5.3.3.3 Permanent and temporary connections

Connections can differ in persistence. Connections can be made during an interaction cycle involving several devices to transfer content or data from the one device to another, and the connection then stops existing when the interaction cycle is completed. Connections can also be used to configure more permanent information exchange between entities in a smart space, much like setting up a connection to a wireless network router. These *permanent* connections will persist, and will be automatically reconnected every time the smart objects that are connected, co-exist in the same smart space.

5.3.3.4 Connections connect different entities

Connections can exist between smart objects, people and places. Not only objects and devices have meaning in a system of networked devices—according to (Poole et al., 2008), physical location within the home and device ownership (or usage) are of central importance for understanding and describing home networks by users. Ownership can be seen as a connection between a device and a person. Connections from and to places or locations can be seen as a way of structuring contextual information such as location. With very personal devices (such as smart phones and laptops or tablets) we can, when these devices are used in an interaction, implicitly infer the users identity. With shared devices, we need a way to identify the user. In such cases, making explicit connections from the device at hand to something personal of the user (e.g. a phone or keychain) may be a way to indicate identity.

5.3.4 Interaction events

Interaction events are generated when a user interacts with a smart object that is connected to another smart object using semantic connections, as was described in the previous section. A distinction is made between control and content. Interacting with a device, e.g. pressing a "Play" button or moving a volume slider, is considered control and described using interaction events. Content, e.g. a song or a photo stored on the device, is referred to by where it exists on the device, as well as how it can be rendered using the media capabilities of the device (e.g. auditive, visually or both).

Definition 5.3.4 Interaction event: An interaction event is defined as an event that occurs at a certain time instant and was generated by a specific smart object. It reports either the intent of a user's action directly, or a perceivable change in a smart object's state.

An *intentional* interaction, like pressing a light switch, is an interaction event if this information is shared by the light switch. *Incidental* or expected interactions, like the light turning on if the presence sensor is triggered, is also an interaction event. *System events*, like a `TimeSetEvent`, which are invisible to the user are not considered interaction events.

In the iStuff toolkit (Ballagas et al., 2003) *hierarchical events* were used to abstract low-level events into application-level events (also see Section 2.4.6). We also introduce the notion of an event hierarchy, where low-level events are considered to be very *generic*, as they do not report a user's intention directly (Niezen et al., 2011). These low-level events first need to be transformed into intentional events—events that express user intention.

We build on the different interaction layers of Nielsen (Section 2.4.3), listed earlier in Table 2.1, to categorize interaction events. As an example, consider the case where a rocker switch, modeled as an interaction primitive on a mobile device, is used to control the volume of music in a room. One could start modeling the interaction on the physical level with a `ButtonPressEvent`, but it would be more meaningful to model it on the lexical level as a `ButtonUpEvent`. On the syntax level this event could increment a quantity by one, while an event generated by a volume dial might have a discrete value attached to it. When this is combined with other device information, for example that the device is being used to stream music to the environment, we can infer on the semantic level that it is a `VolumeUpEvent`. When this is combined with other contextual information, for example that the device is currently connected to a speaker system in the same room, we can even infer on the task level that the music volume in the room should be set to a specific value with a `MusicVolumeUpEvent`, to which all connected devices can respond. We acknowledge that in most cases it may not be possible to make inferences on the goal level, which in this example could be the user's intent to set the music volume in the room to a level that is loud enough for everyone in the room to dance to.

5.3.5 Semantic transformers

Semantic transformers were first defined in (Niezen et al., 2011) as virtual entities that transform one type of information into another when a direct mapping is not possible. They transform user actions into interaction events and perform matching and transformation of shared data and content. Semantic transformers enable interoperability between devices by utilizing device capability descriptions and content types to determine how devices may interoperate.

Definition 5.3.5 Semantic transformer: A semantic transformer is defined as a service that transforms information shared between devices from one type to another, while preserving the meaning of the information.

Semantic transformers can be used to map and transform shared content between smart devices, for example a service that transforms a music stream into coloured lighting patterns that can be rendered by a lighting device is one example of a semantic transformer. Semantic transformers can also be used to transform physical actions (such as pressing a button or performing a gesture) into representational events like adjusting the level of lighting in a room, or adjusting the volume of a speaker. Semantic transformers may also be employed to perform simpler transformations such as inverting values. Physical identifiable objects are not considered semantic transformers and should rather be modelled as smart objects.

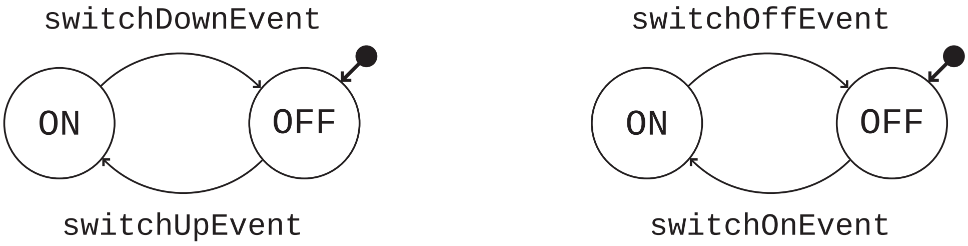


Figure 5.4: FSMs for a simple light with a switch and a light with a labelled switch

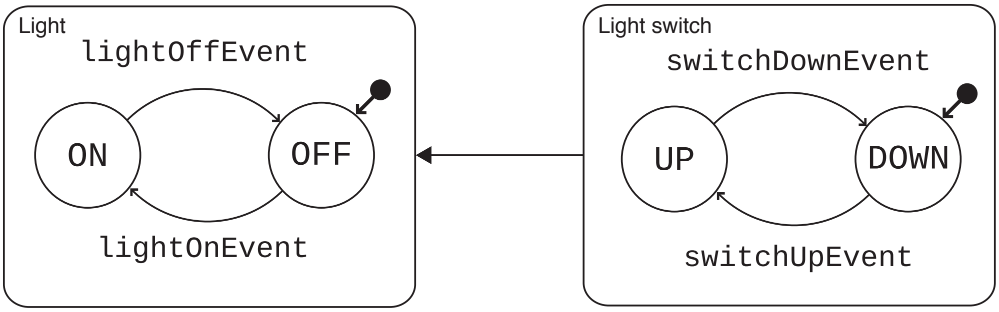


Figure 5.5: Light and light switch as two separate smart objects with a semantic connection

5.3.6 Finite state machine examples

We now use finite state machines (FSMs) to model and explain the different concepts introduced so far in this chapter. FSMs allow us to talk about user interaction in a way that describes how *users* think about user interaction, but that still makes sense to interaction programmers and designers. Turing machines and other sophisticated models of computing do not describe that what users think about (Thimbleby, 2007). Although FSMs do not reflect the parallel and associative thinking of real users, the model is a fair compromise between being intuitive and being formal. The use of FSMs also encourages simplicity (in contrast to more capable, but more complex models such as models based on pushdown automata or even Turing machines). A legend of the used notation is available in Appendix F.

As a first example, consider a simple light with an up/down switch as a single device (seen in Figure 5.4). There are two states (`On/Off`), an initial state (`Off`), indicated by the arrow, and two events (`SwitchDownEvent/SwitchUpEvent`) that cause transitions between the states. If the switch is labeled, we can use more specific (meaningful) wording, for example `switchOffEvent` instead of `switchDownEvent`. We use the notation `[DeviceClass]` (e.g. switch) `[Action]Event` (e.g. down + Event) to define the interaction event - this makes them easier to classify into a hierarchical event structure.

In Figure 5.5 one of the simplest examples of a semantic connection is shown—a light

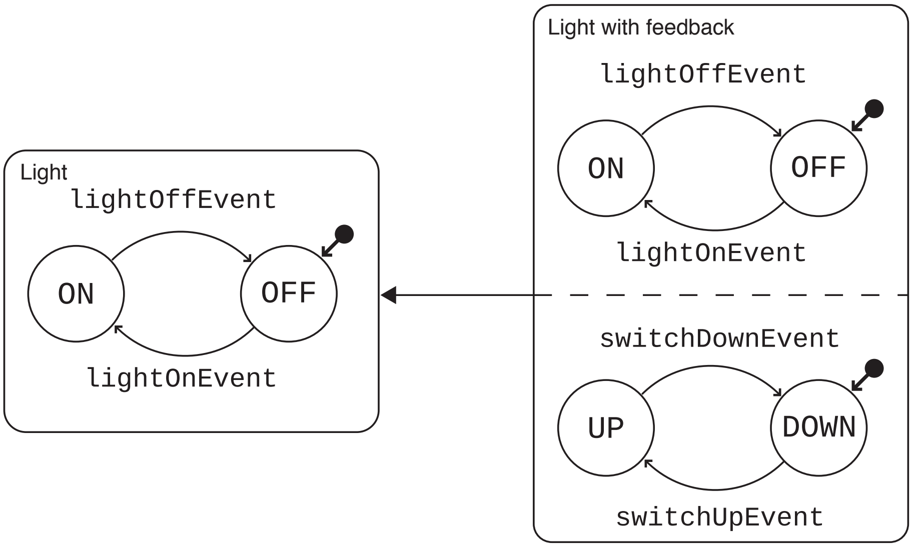


Figure 5.6: Light connected to light switch with augmented feedback

(as a smart object) connected to a simple on/off switch (a second smart object). The light consists of two states (`On`/`Off`) with an initial (default) state of `Off`, and two events (`LightOnEvent`/`LightOffEvent`) indicating the transitions between these states. Boxes with rounded corners are used to signify smart objects, while the semantic connection is indicated using a solid arrow point. As we use arrows for semantic connections, we specify a direction for the connection (we described the concept of directionality in section 6.3). Note that the light has functional feedback, with perceivable light when it is switched on. The switch on the other hand has inherent feedback, with a perceivable `Up` or `Down` state.

We can create mappings between the events to create an interaction path (see section 5.3.2.2), for example we use

SwitchUpEvent → LightOnEvent

to indicate the most meaningful *default* mapping. It should of course be possible to change this mapping, for example by using a semantic transformer (Section 5.3.5) that inverts mappings between devices.

In the case where a smart object is not in the same physical location as the smart object it is connected to, additional augmented feedback may be required. Consider the case where the light switch may be in a different room than the light - we could use an indicator on the switch to give augmented feedback to show whether the light actually switched on. This is shown in Figure 5.6, where the light with augmented feedback is a combination of a switch and an integrated light (also indicator) that changes from the `off` to `on` state with a `switchUpEvent` and vice versa. Note that in this case `switchUpEvent` and `lightOnEvent` are equivalent and there exists a mapping between the two.

A more complex example is shown in Figure 5.7. Here there is a symmetric (bidirectional) connection between smart object A and B, with the result that pressing the switch on smart object A will turn the light in smart object B either on or off, and vice versa, B will control

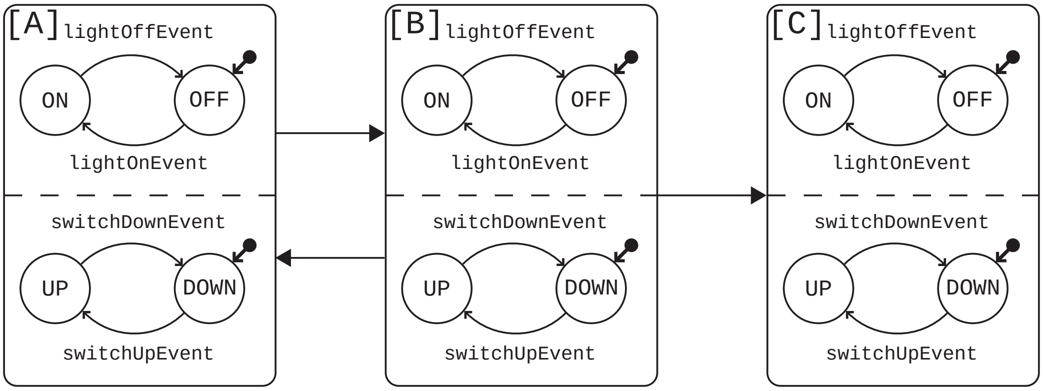


Figure 5.7: FSM showing semantic connection with symmetry

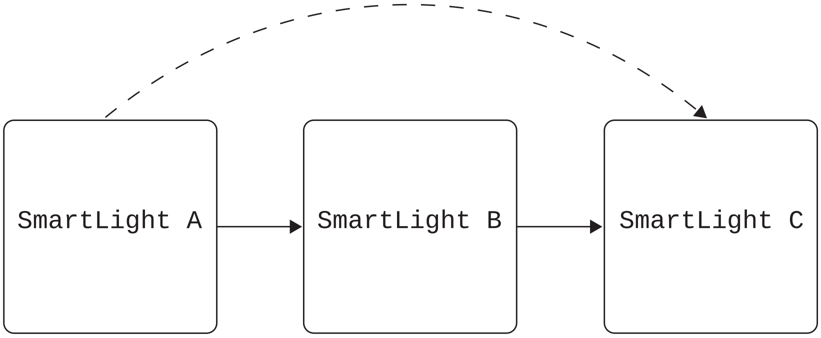


Figure 5.8: FSM showing a semantic connection with transitivity

A. Since B is connected to C, actions on A and B, will also be reflected on C. On the other hand, pressing the switch of C will have no effect on either A or B.

In Figure 5.8 we use the SmartLight abstraction to denote the FSM of a smart light as shown in previous figures. When SmartLight A is connected to SmartLight B, and in turn is connected to C, transitivity allows us to infer a connection (indicated by a dashed arrow to distinguish *inferred* connections) directly between A and C. Pressing the light switch on A will in this case affect both B and C.

We use locked/unlocked icons next to semantic connections to indicate persistence (see Figure 5.9). The locked icons between A, B and D indicate persistent connections between those objects, and a persistent transitive connection is then inferred between A and D. This means that if smart object D moves to another location and back, all three the connections (including the A→D connection) will be re-established. The connection between B and C is temporary, which means that the inferred transitive connection between A and C is also temporary. If smart object C moves to another location, both the B→C and A→C

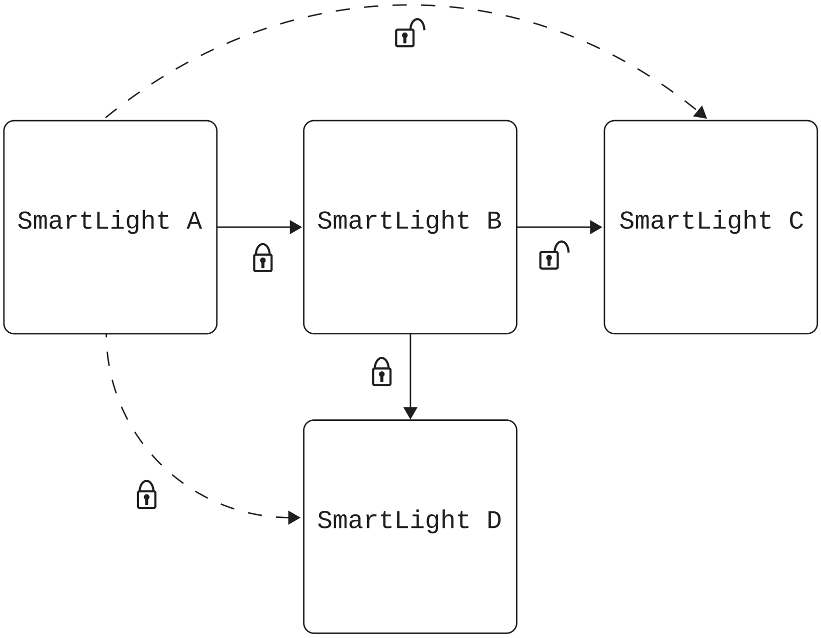


Figure 5.9: FSM showing a semantic connection with transitivity and persistence

connections will be removed and have to be manually recreated.

In Figure 5.10 we show semantic connections between smart objects and specific locations, where the dashed-double circle denotes a location. This places semantic connections between places and objects on the same abstraction level. We use semantic connections between smart objects and places as a way to structure relevant contextual information. In our example in Figure 5.10 we cannot infer that a user actually is able to observe the functional feedback of switching the light on and off, as they are not located in the same space, and might not be able to see the light. The importance of feedback and feedforward and how they should be handled between different locations is described in more detail in sections 5.4.3 and 6.2.9.

When two switches are connected to the same light as is shown in Figure 5.11, the issue of priority arises. We define the most meaningful default to be that the last event that occurred has priority. In Figure 5.12 where the one interaction is incidental, generated by a presence sensor, and the other is intentional (as described in Section 2.4.7), the intentional interaction takes priority.

5.3.7 Feedback and Feedforward

As we have shown in our interaction model (Section 5.3.1) and the examples in Section 5.3.6, the function and use of the different types of feedback and feedforward changes when we interconnect smart objects. If we view our concept of semantic connections in terms of the Interaction Frogger framework (as discussed in Section 2.10), the following interesting insights emerge.

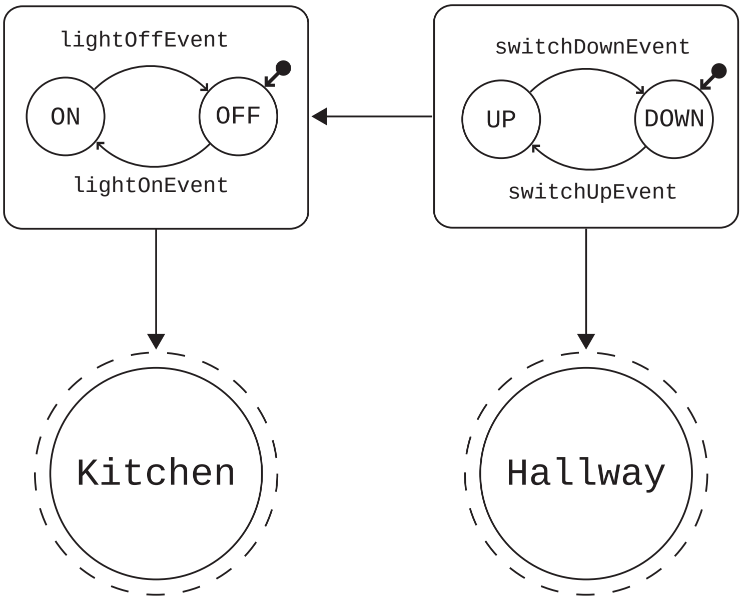


Figure 5.10: FSM showing semantic connections between smart objects and places

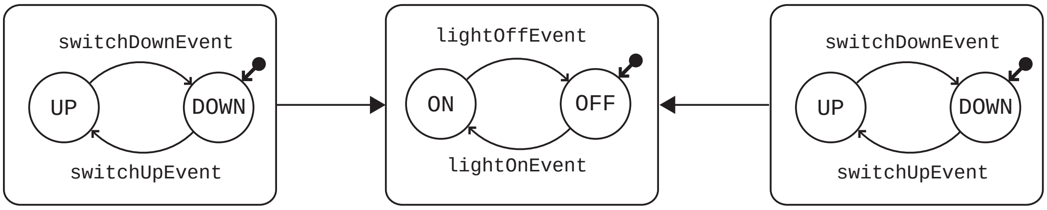


Figure 5.11: FSM showing a situation where priority is an issue

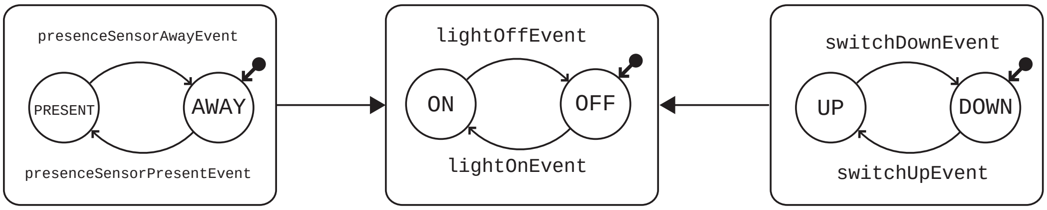


Figure 5.12: FSM showing incidental (presence sensor) and intentional (light switch) interactions

5.3.7.1 Feedback

When we consider multiple interconnected smart objects and the functionalities and services they provide, information like feedback and feedforward gets spatially distributed. A user may operate a device, receiving inherent feedback locally, but receiving augmented feedback about a remote functional event.

As inherent feedback is inherent to the operational controls of the device, these reside only in the physical world and are local to the device. We thus do not model this feedback in the digital domain[8]. Augmented feedback is feedback that is augmented from the digital domain onto the physical world. This type of feedback is subject to change when devices are connected to other devices. In the domain of networked digital artefacts, functional feedback is often of a digital nature. Information, media and services that exist in the digital domain become available in the physical world, through the various devices and their connections. In Figure 5.3, the various types of feedback are indicated.

Although many functionalities of digital devices can be regarded as displaying media, data or services, for some simple functionalities this seems problematic. If we, for example, look at functional lighting, it seems that the presence of light as the functionality of a lighting device is not a concept that is part of the digital domain. However, if we view a lighting device as a networked smart object, the presence of lighting, based on some sensor data, can be regarded the functionality of a digital service.

As mentioned before, semantic connections (potentially) *change* functional and augmented feedback and its location. Additionally, semantic connections themselves show feedback as well, when users interact with them through an interaction device. Inherent feedback becomes feedback that is mediated through an interaction device used to make or break the connection, as one can not manipulate a wireless connection directly. This inherent feedback may however be closely related to the action of making or breaking a physical connection, like a snap or click when the connection is made or broken. Augmented feedback to indicate a connection possibility or an existing connection may be in the form of lights, or in the form of projected or displayed lines (as was shown in our design explorations). Functional feedback is information about the actual function of the connection, like music playback from a speaker that was just connected to a media player. This type of feedback always reaches the user through the devices being connected.

5.3.7.2 Feedforward

Inherent feedforward, conceptually similar to the notion of affordances (Norman, 1998), provides information about the action possibilities with the devices or the individual controls of an interface. Similar to this are also *informatives* (Krippendorff, 2006, p. 117) and partially also *indication* or *marking functions* as defined in the theory of product language (Gros, 1983). Inherent feedforward is always physical and locally on the device. However, when devices or objects are part of a larger system, feedforward also emerges where interaction possibilities between objects exist (e.g. a key that fits a lock, a connector of one device or cable that fits another). The same holds for augmented feedforward, where lights, icons, symbols and labels provide additional information about the action possibilities. These may

---

[8] If possible, the physical properties that may have meaning to an interaction, can be modelled as interaction primitives, e.g. up/down state of a button.

concern the action possibilities locally at the device, as well as action possibilities that concern the interaction with other devices in the environment.

While inherent and augmented information are primarily concerned with "the how", functional feedforward communicates "the what", the general function of the device or the function of a control. This type of information often relies on association, metaphors and the sign function of products, and are described in theories such as product semantics and product language. With multifunctional devices, and even more with smart objects, this becomes increasingly difficult. Introducing the concept of semantic connections tries to address these problems, therefore the functional feedforward is the main challenge when designing semantic connections. Functional feedforward gives information about the function of the semantic connection before the interaction takes place. Properly designing functional feedforward is therefore the crucial part of understanding semantic connections, smart services and smart environments.

Wensveen (Wensveen et al., 2004) further proposes that in interaction, these types of information can link action and function together in time, location, direction, modality, dynamics and expression. Strengthening these couplings between action and function will lead to richer and more intuitive interactions (Wensveen, 2005).

We can also view semantic connections in the Frogger framework in more general terms. Although semantic connections are not a physical device or product, but rather describe the structure or configuration of a system of devices, the Frogger framework can teach us important lessons. When we look at the link between action and functional information in time or location, a strong link would mean they coincide in time and location. For location this would mean that the connection that is made between devices corresponds to the location of the actual devices in physical space. But also that the feedback that is provided is coupled to the action in time and location. Additionally, the direction of the action of connecting/disconnecting devices, being moving devices towards or away from each other, strengthens the coupling in terms of direction. Also, the direction of the action could have a link to the directionality of the semantic connection that is made (e.g. the order in which endpoints of a connections are defined). Couplings in dynamics (of the action) can be used in similar ways and may express the persistence of the connection that is made.

At the end of this chapter (Section 5.4.1) we give examples of how we implemented the different types of feedback and what factors show to be important. In chapter 6 a framework is described in which feedback and feedforward and their behaviour among interconnected smart objects are defined more precisely.

5.4 Evaluation

The theory was put to the test by implementing it in a new use case. The use case scenario revolves around the evening routine before going to bed and sleeping, and is an extension of our previous scenario's. It is cross-domain, as it extends the media domain into the domain of sleep and opens the possibility to exchange other types of information. The domain of sleep was chosen for several reasons:

- Sleep is important for physical and mental well-being, an important application area of our research group.

- The sleep domain is targeted by a number of recent *IoT (Internet of Things)* devices that record and share data and can be accessed through APIs (e.g. Zeo Sleep Manager[9], Jawbone UP[10], Sleeptracker[11] and FitBit[12])
- The sleep domain allows us to reuse some of our existing work on media sharing and lighting, extending it into a new domain.

The use case consists of several devices. It includes an Android smart phone (Samsung Nexus S), an internet radio (Logitech Squeezebox Radio), a dimmable lamp (built in-house, Arduino controlled), a Zeo Sleep Manager and an Android tablet (Samsung Galaxy Tab 10.1 WiFi). We purposefully did not define a narrative for the use case, to prevent from only implementing the functionality described in the narrative. Instead, we looked at meaningful combinations we could make with the devices, attempting to allow for *emergent functionalities* to surface by sharing device capabilities and interaction events.

The use case was implemented in the master bedroom of the Context Lab, a lab with a setting that resembles a real home. Implementing the use case in-context allowed us to see its behaviour and implications in a realistic setting, giving insights that are regarded more valuable than obtained when building a set-up on e.g. one's office desk. In this section we describe the implementation of the sleep use case as an evaluation of the completeness and applicability of our semantic connections theory, evaluating whether: (a) the defined concepts in our theory are sufficiently defined to use them to implement the required functionalities, (b) the defined concepts can be used universally (for different use cases) and (c) the defined concepts form a complete set to describe the behaviour of semantic connections. Additionally, the implementation served as an example of how our theory can be used in a relevant and contemporary setting. We describe implementation details of how the concepts of the theory are modelled and conclude with changes and additions to the theory that were deemed necessary.

We started by implementing a very basic configuration, connecting the phone and the internet radio. Based on the capabilities of the devices, possible connections included sharing music player functionality and alarm clock functionality. After implementing the first basic functionalities, we gradually increased complexity by adding another smart object, the dimmable lamp, followed by the implementation of the several types of feedback that were described in Section 5.3.7.

The Connector, as is described in Section 4.4 was used as an interface to the semantic connections in this use-case. The intended capabilities of the connector included more than what was implemented for the SOFIA pilot evaluation (Section 4.7), as for the stability and consistency of the setup, some of the functionalities were simplified. Reusing the Connector in contrast to redesigning a new solution, was also expected to result in useful insights on how the physical design restricts use and what properties need to be changed if these restrictions are unwanted. Taking the Connector and the semantic connections theory as a given, also allowed us to shift our focus to how subtleties in design and interaction with the smart objects impact the overall user experience. Therefore, the design focus was not on the object interfacing with the connections but rather on the behaviour of the smart objects which are connected.

---

[9] http://www.myzeo.com/sleep

[10] http://jawbone.com/up

[11] http://www.sleeptracker.com

[12] http://www.fitbit.com/product/features#sleep

5.4.1 Implementation Examples

In the sleep use case, music can be shared between the smart phone and the internet radio. Alarms can be shared between the phone and the internet radio, the internet radio and the lamp as well as the phone and the lamp. Because the lamp has only `LightOn/LightOff` and `AdjustLevel` capabilities, the most basic functionality of the lamp responding to an `AlarmEvent`, would be to turn on at the time that the event occurs. However, a wake-up service can be connected that *transforms* an `AlarmSetEvent` into a wake-up experience, sending a sequence of `AdjustLevelEvents` to the lamp. This wake-up service then functions as a *semantic transformer*, transforming one type of value into another, in a meaningful way. Semantic transformers are virtual entities and therefore they do not have a physical presence, in contrast to smart objects that must have a physical representation. Therefore, the use of a semantic transformer is automatically inferred based on its capabilities.

The Wakeup KP (Figure 5.13) is a type of semantic transformer, and transforms an `AlarmSetEvent` into a series of `IncreaseLevelEvents` (which is a sub-property of `AdjustLevelEvents`). However, this happens in a way that is designed by the developer of the KP and as such, can be seen as a (digital) service. This wake-up service has similar functionalities as a Wakeup Light (e.g. as sold by Philips[13]) which means it starts increasing its light level over a 30 minute time-period, reaching full intensity (as calibrated) at the set alarm time. The semantic connection between the phone's alarm and the dimmable light is an example of how such a connection can have emerging functionality, which does not exist without the connection (and in this case a digital service).

This opens up many possibilities for users, as they may connect other lights, and potentially even other devices such as a networked thermostat, to either the alarm or the dimmable lamp, creating their own wake-up experience[14]. They may also connect a sleep monitor as an alarm source, helping them to wake up at the right time in their sleep cycle [15].

Our use-case was implemented using the same software infrastructure (with minor modifications) as was used during our pilot evaluation, which is described in Section 4.6. For a full description of the implementation see the work of Niezen (2012).

5.4.1.1 The Semantic Interaction ontology

The devices (in terms their capabilities and interaction primitives), the interaction events and the semantic connections that exist between the smart objects, are modelled in ontologies. We now show the ontology that was used to describe the smart objects. The Semantic Interaction ontology we have developed, as was discussed in Section 4.6.2, is shown in Figure 5.14. A device, defined as a (`SmartObject`), is described in terms of its interaction primitives, as described in Section 5.3.2.2. It is also uniquely identified by some kind of identifier (`Identification`, for example a IP address and port number, RFID tag or barcode (see Section 5.3.2.1). Different ID types can be defined as required. Devices

---

[13] `http://www.philips.co.uk/c/wake-up-light/38751/cat/`

[14] Whether such emerging functionalities are possible obviously depends on the way the smart objects are implemented. E.g. in our implementation, the dimmable lamp is described as a sink, which means it is only capable of accepting input. In the case it was described as a source as well, sharing for instance its on/off state or its current light value, it could act as a bridge and allow for more interesting configurations.

[15] Although it was not implemented, the Zeo Sleep Manager shares its data like sleep states and alarm events, which can be used in our sleep use case.

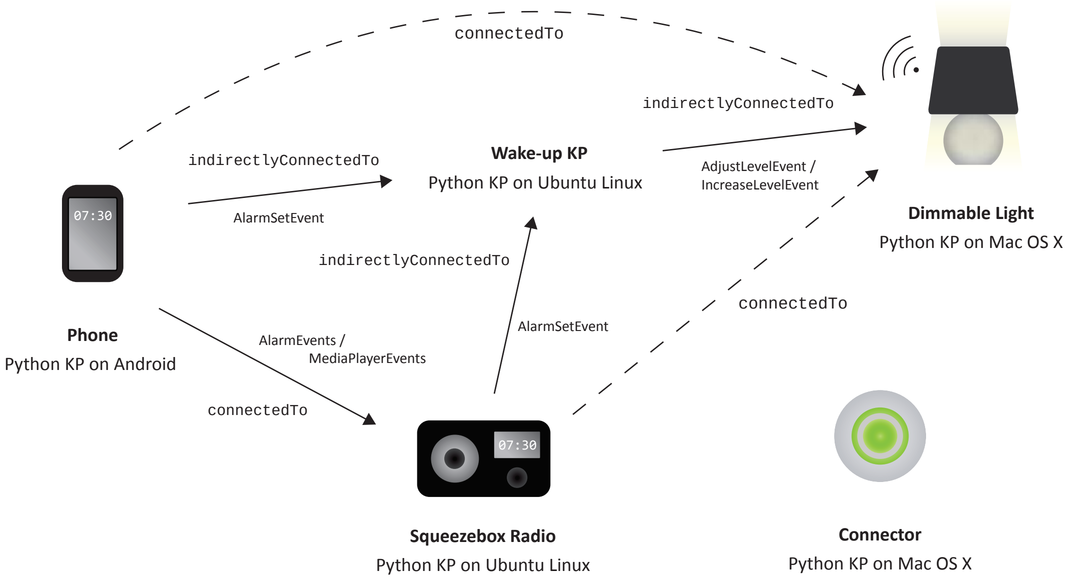


Figure 5.13: An overview of the sleep use case

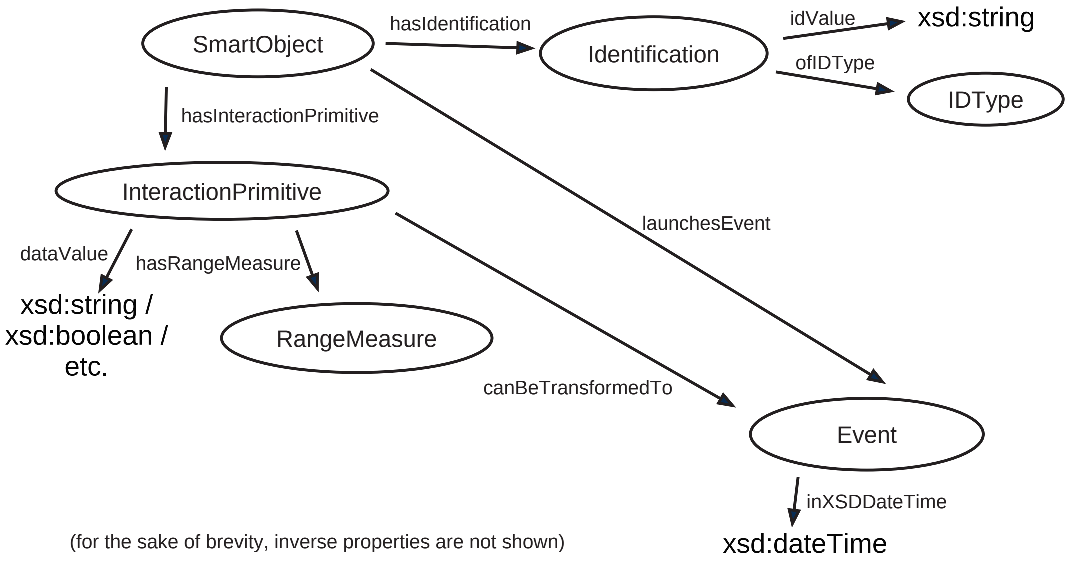


Figure 5.14: Semantic Interaction Ontology

Table 5.2: Range measures for interaction primitives

Range measure	Possible values
Binary	True/False, 0 or 1
SingleDigit	up to 9 discrete values
DoubleDigit	up to 99 discrete values
TripleDigit	up to 999 discrete values
LargeDigit	more than 1000 discrete values

can either launch events themselves (e.g. a media player generating a `PlayEvent` when music starts playing) or an interaction primitive of the device can be transformed into an event.

The range of values that an interaction primitive can take on is specified using a `RangeMeasure`. The current list of range measures is shown in Table 5.2. These range measures are similar to the measure of the domain set used by MacKinlay et al (Mackinlay et al., 1990). Using the range measures, we can then infer which transformations may be used to map the input values to other interaction primitives or events. The ontology could be extended to also describe the different manipulation operators of the interaction primitive, e.g. rotation on the z-axis or movement along the y-axis.

By specifying the transformation using the proper OWL 2 semantics, the reasoner should be able to infer which user inputs can be mapped to which specific high-level events. This shows up as a `canBeTransformedTo` property between an interaction primitive and an event.

Information about the other ontologies (or other parts of the ontology) are described in Section 4.6.2 and (Niezen, 2012)[16].

5.4.1.2 Device states

Interaction events (Section 5.3.4) cause device state changes. When smart objects are interconnected, mismatches in device states may occur, as not all interaction events cause the same state transitions in all smart objects. Most of the developers that worked on the Smart Home Pilot (section 4.6) preferred to describe their smart objects in terms of the device states, and also shared these device states with other smart objects using the SIB. Our reasoning to rather describe the interactions in terms of events as opposed to states was that states can be logically inferred from events. Exchanging the events still leaves some autonomy to the smart object (or its developer) to decide what to do with the event. However, only sharing events is not always enough to create consistent behaviour. Although, it is the responsibility of the source to communicate all state changes (events), in order for the sink to keep in sync, it is still possible for two smart objects to be in different states while connected.

We encountered a problem with state mismatches in our sleep use-case that resulted in unexpected behaviour. In this specific case, the mobile phone was connected to the internet radio, sharing both alarm and music functionality. If the user opens the music player on the mobile phone and presses Play, it causes the music to play back on the internet radio.

---

[16] The ontologies described in Section 4.6.2 and this section were designed by Niezen, based on ideas that were developed jointly. More specific, I made a substantial contribution to the ideas behind the interaction design specific elements they aim to describe.

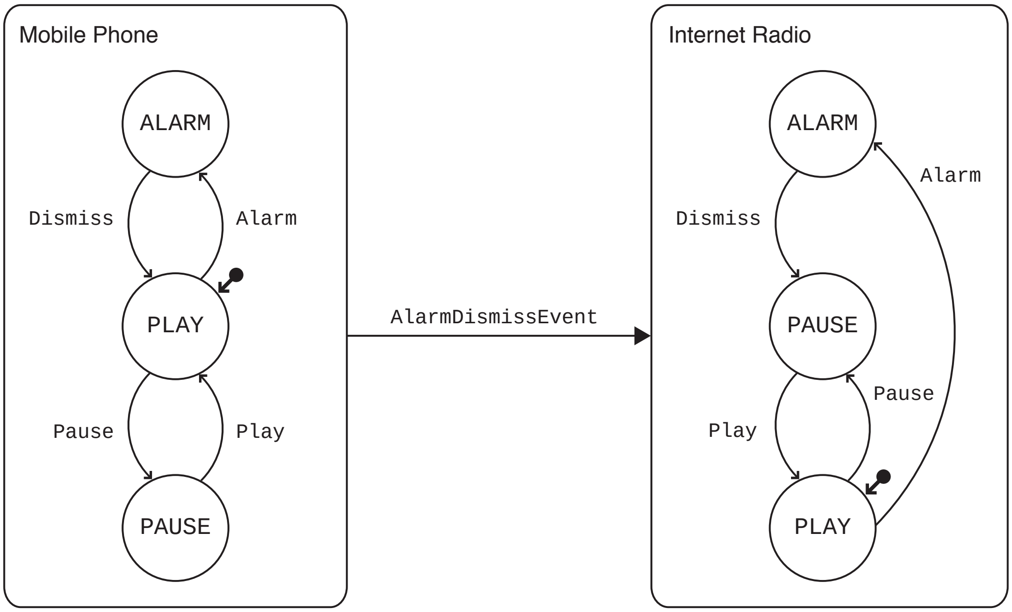


Figure 5.15: Finite state machine showing the state mismatch between phone and internet radio

Besides playing music, an alarm was also set. When the `AlarmAlertEvent` generated by the mobile phone occurs, both the phone and the internet radio go to an alarm state. Dismissing the alarm on the phone causes the internet radio to go into a `Pause` state, as this is the default behaviour when the alarm is dismissed on the internet radio itself. The mobile phone, however, is now in the `Play` state (although no audio is audible since the speakers of the phone are muted because of the connection to the internet radio). In order to play music again, the phone needs to be paused first, followed by pressing play, to get the device states back into sync. Figure 5.15 displays this behaviour using a finite state machine.

To prevent this state mismatch, the mobile phone should not only send the `AlarmDismissEvent` that was generated through the user interaction, but also a `PlayEvent` to indicate that it is now playing music again.

5.4.1.3 Semantic transformers

In the sleep use-case, a semantic transformer was implemented in order to generate lighting values for the dimmable lamp to create the desired wakeup experience. During the implementation, several observations and decisions were made:

- Between smart objects and semantic transformers only `indirectlyConnectedTo` connections can exist, as the semantic transformers are virtual entities that cannot be directly connected to smart objects using the Connector object.
- When a `canIndirectlyConnectTo` relationship is inferred between smart object A and the semantic transformer B, and between B and smart object C, a `canConnectTo` relation between A and C should be inferred (transitive).

- When a connection is made between two smart objects that can be connected through a semantic transformer, the semantic transformer is connected to the smart objects with `indirectlyConnectedTo` relationships, and a `connectedTo` relationship between the smart objects is then automatically inferred.

- A semantic transformer thus acts as a *bridge*.

- A semantic transformer is *not* a smart object.

5.4.2 Interaction with semantic connections

During the implementation phase, several changes were made to the Connector object to support the functionalities of the sleep use-case. In this section, the most important changes are discussed, as well as the limitations of the Connector's design and their implications for designing for interoperability in general.

5.4.2.1 Directionality

Although the Connector was originally designed to support the notion of directionality in the interaction with semantic connections, it was not implemented during our previous iteration. The two rings of the Connector's display signify input and output, or source and sink. Where in the previous iteration they indicated the selection of a first and then a second endpoint for the connection (which was symmetric) this was now changed to identify source and sink. The order in which the objects are physically identified is used to determine the direction of the connection.

Practically this means that the first object a user selects, should be the source object of a connection. When the smart object which is selected is not a source, this is indicated by means of feedback. The same holds for the selection of the second object, which is required to be a sink. When the second object that is selected is not a sink, this is indicated by feedback. For exploring existing connection, the procedure to follow is similar. The order of selecting defines the source and sink to query for an existing connection between the two and feedback is provided accordingly.

To help users understand why connections are not possible, e.g. their capabilities do not match, the first object is not a source, the second object is not a sink, or an error is returned, the connector should provide distinct feedback for each situation[17]. In case the first object is not a source, the outer ring flashed red (indicating it is a sink). When the second object is not a sink, the inner ring flashes red, indicating the object is a source.

Introducing directionality with the current design of the Connector basically doubles the number of actions as between two given smart objects two connections instead of one may exist. This makes the required actions for more complex configurations cumbersome, and clearly shows the limitations of the current approach. A more comprehensive discussion on directions for a redesign are discussed in section 6.3.4.

---

[17] The Connector already differentiates between connection possible, connection exists, connection is not possible and a state where both objects are scanned, but an error occurs (e.g. error in the KP, smart object is not subscribed to the SIB). In such cases, both rings remain orange.

5.4.2.2 Transitivity

With directional connections implemented, the notion of transitivity defines some smart objects as *bridges*. Bridges are different from sources and sinks when they are interacted with, as they behave both as a source and a sink.

5.4.2.3 Permanent and temporary connections

A notion of persistence for a connection was implemented in this iteration. When a source and sink are selected and a connection is possible, temporary connections (in contrast to regular connections; `connectedTo` relationships) were implemented to enable the use of *preview events* (also see Section 5.4.3.1 below). Temporary connections may also be created between the source and the sink when a possible connection is being explored. Therefore, while the system is in this state, the temporary connection can be used to exchange information. When the connection is confirmed, or when the view/explore state is cancelled, the temporary connections either become permanent, or are removed again.

5.4.3 Feedback and Feedforward

As was described in Section 5.3.7, feedback and feedforward need to be carefully designed when smart objects are interconnected. However, as the smart objects themselves are unaware of each other and, at development time, their designers did not know to what other devices users may connect the smart objects to, the total user experience can not easily be designed[18]. In Chapter 6 a framework is introduced that is aimed to help designers of interoperable smart objects design for such unknown future uses. In this section we will describe how feedback and feedforward were used to enhance the user experience and enable devices that are in-fact unaware of one another, appear to show awareness of each other to their users.

We use feedforward to display a device's functional possibilities. We can use feedback to confirm user actions, using augmented feedback where direct functional feedback is not available. When an alarm is set on the phone, augmented feedback should be given on all devices connected to the phone. For example, consider the setup where the alarm is connected to both the lamp and the Squeezebox radio. Immediate feedback only makes sense when the event and its feedback coincide in time and modality (e.g. audio, visual). When the generated event is a `SetEvent`, the event itself will occur sometime in the future, so we generate the functional feedback as augmented feedback instead. For example, for an `AlarmSetEvent` we generate a 1s alert sound on the Squeeze- box radio as augmented feedback, providing functional feedforward of what will happen when the alarm is triggered. We also provide visual augmented feedback by displaying a popup message on the display for a few seconds. On the lamp, feedback is given in the form of a short light pulse to confirm that it has been notified as well.

---

[18] This is why many examples of well working solutions for interconnecting devices often employ the *vendor lock-in* strategy (Section 2.2) which enables manufacturers to have full control over their ecosystem of products and the resulting user experience.

5.4.3.1 Augmented and functional feedforward

For semantic connections, functional feedback and feedforward can only be considered for the combination of source and sink. The source object has functional feedforward (or semantics) that may communicate its function. *Only* when both the source and sink object have been identified, functional feedforward is available for the semantic connection. Important to note is, that functional feedforward is derived from the junction of functionalities of both the source and the sink. This is both problematic and unambiguous, as both source and sink may be multifunctional. If this is the case, users should make explicit what information or data they want to exchange by selecting the desired mode on the source object (e.g. selecting the alarm application on your smart phone to share the alarm time or go to a picture viewer when pictures should be exchanged), restricting the possibilities. If this is not possible, or a multifunctional smart object is connected when it is in idle mode, semantic reasoning will be used to match all meaningful capabilities of the source and sink objects.

Whenever users wish to make a connection, they have certain expectations. We can employ functional feedforward to influence these expectations. Additionally, we can enhance the user's understanding by explicitly adding augmented feedforward (i.e. augmented *functional* feedforward in contrast to augmented *inherent* feedforward). In the sleep use-case we employed augmented feedforward in the process of exploring connection possibilities i.e. before the connection is made. We do this by giving a *functional preview* at the *sink* object, viewing the functionality of the connection that is currently explored. Our reasoning is, that only when both source and sink are identified, we can speak of a semantic *connection* and, by giving the feedforward at the sink, we ensure that the sink object is in-fact capable of producing this feedforward (i.e. has the necessary capabilities). Additionally the location of the feedforward corresponds to the location where the action (identifying the sink object) was performed. To do so, a `PreviewEvent` is generated when a possible connection is being explored, displaying the possible functionalities enabled by the connection.

Example 5.4.3.1: When a user, after having identified the phone as a source object, identifies the internet radio as a sink, the display of the internet radio displays a message: "Alarm can be shared" and "Music can be played". Previews can also be less explicit, like briefly sounding an alarm and playing a short music clip. Note that the preview can be ignored or bypassed by establishing a connection.

Example 5.4.3.2: For exploring a connection between the internet radio and the dimmable lamp, the lamp simulates a wake-up sequence, increasing the light level from zero to its maximum intensity in a given period of time (in our implementation three seconds). This may be enhanced with simulating an alarm at the Squeezebox Radio when the maximum of the intensity is reached. However, due to limitations in synchronisation in our current implementation this was not implemented.

Practically, this means that the designer/developer of a smart object should design how to respond to a `PreviewEvent`. Technically, this is implemented by having the Connector object (more precisely; the connector KP) create a temporary connection to the devices to be connected in order to generate a `PreviewEvent`. This `tempConnectedTo` property is a sub-property of the `connectedTo` property (which denotes a regular semantic connection). This means that the smart objects will handle it as if it is a regular connection, and when the Connector object removes the `tempConnectedTo` relationship, the inferred `connectedTo`

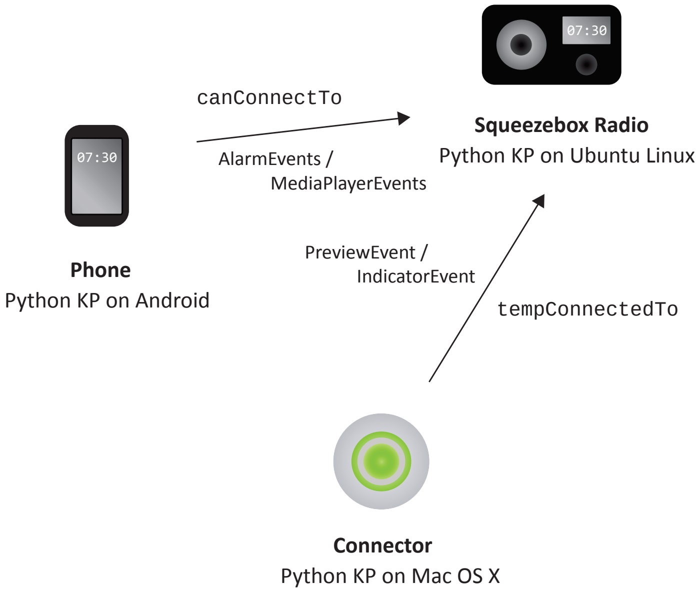


Figure 5.16: Temporary connections for a `PreviewEvent` *when source and sink are directly connected*

relationship will disappear as well. The type of functionality the preview is for, is added to the preview event as a data value.

The system behaves differently depending on the type of relation between the smart objects. When there is an indirect connection, i.e. going through a semantic transformer, the preview event is sent to the *semantic transformer* (Figure 5.16) instead of the sink object (Figure 5.17). Additionally, a temporary connection is made between the semantic transformer and the sink, ensuring that the sink displays the actual feedforward.

When the semantic transformer receives the `PreviewEvent`, it generates yet another event for the preview (as was designed by the developer).

5.4.3.2 Functional feedback

In many cases functional feedback of a semantic connection is trivial (e.g. hearing sound from a speaker that was just connected to a media player, seeing photo's on a TV when connected to a smart phone). However, functional feedback may only be available at another place and/or at another time. If we for instance take the example of synchronising a phone's alarm with the alarm radio, the real functional result may be hearing the alarm radio, play a song at the alarm-time, that was set on the phone.

In such cases, it is accustomed for interaction designers to use augmented feedback as an *indicator* that the alarm-time was successfully set. When a semantic connection exists between a source and a sink, actions at the source should also be confirmed at the sink.

With the source and sink objects located at different places, interaction designers should make sure that feedback is visible for a prolonged time-period or until it is dismissed by the user, ensuring that the confirmation of the action will be notified. For the same reason, the order of connecting two spatially separated objects together is important, ensuring that

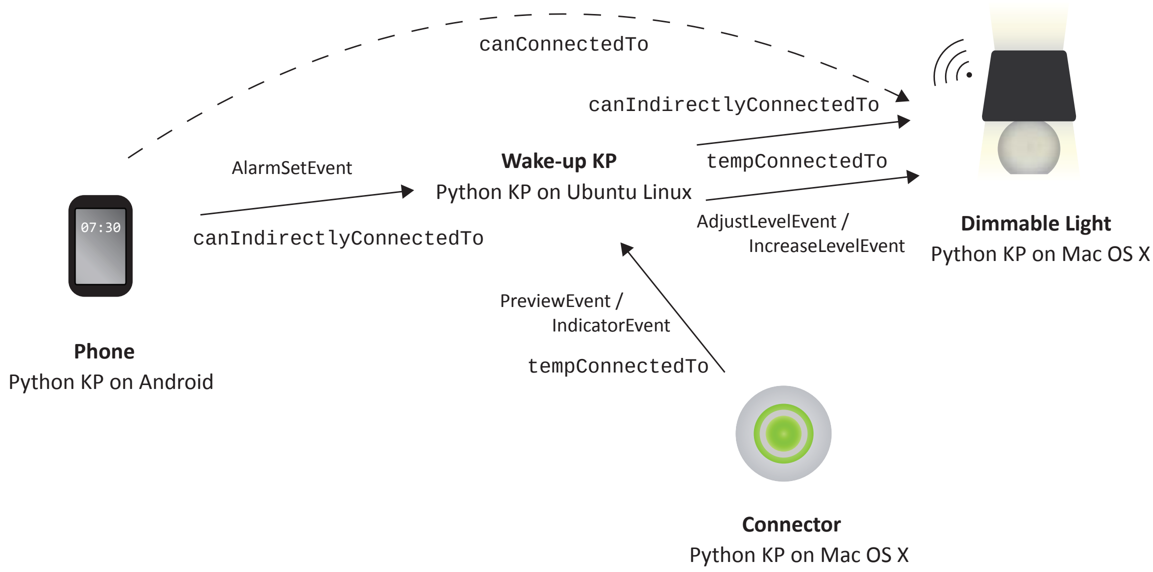


Figure 5.17: Temporary connections for a `PreviewEvent` *when source and sink are connected via a semantic transformer*

establishing the connection happens in proximity of the sink, so that the feedback can be observed.

Example 5.4.3.3: When music is playing on the phone and a connection is made between the phone and the internet radio, functional feedback is immediately given: the internet radio starts playing the same music, and an image of the album cover (or when that is not available the name and artist of the song) is displayed. The music on the source (phone) is muted, as the music playing at the internet radio is considered of a higher quality, and both share the same physical space. We intend to use context information such as place/location to decide what to do.

5.4.3.3 Augmented feedback

If there is no immediate link between action and function (e.g. functional result is delayed, information is given about an internal state change), augmented information can be used to provide this information. For semantic connections, augmented feedback is important information from the underlying digital state of the connection. It should be used by designers in cases where the action and functional result do not coincide in time, but also more generally to indicate that a user's physical actions had its effect in the digital domain. We use an `IndicatorEvent` to provide augmented feedback when a smart object is connected and there is no immediate functional feedback, e.g. a sink "beeping" when the alarm is set on the source; used to confirm actions. The type of feedback required depends on the functionality of the connection. It is important for the feedback to coincide in time and modality with the event generated, as to maintain the causal link that is perceived by the user.

When a connections exists and an action performed on the source that has no immediate functional feedback, augmented feedback is provided to serve as an *indicator*. This feedback is provided by the smart objects that are connected, in the modality that is supported by their interaction capabilities. Designers should aim for maintaining the modality of the augmented information across the smart objects. Additionally, ensuring that the feedback occurring at distributed objects coincide in time may strengthen the perceived causality of the link. Indicator events may also be used to indicate existing connections, e.g. when a user wishes to see what smart objects are currently connected to a source.

Example 5.4.3.4: When the phone is connected to the internet radio and the internet radio is connected to the dimmable lamp, both the internet radio and the lamp gives augmented feedback when an alarm is set. The internet radio displays the alarm-set screen, confirming the alarm time and the dimmable lamp slowly flashes, to indicate that they are both connected and that the action on the source is confirmed.

5.5 Discussion

In this chapter the *semantic connections theory* was introduced, and we used finite state machines to model and explain the different concepts. We showed an implementation of a new use-case in a smart home setting to validate the theory, and reported on interesting observations, and changes and additions made to the theory.

We defined *smart objects* and means to describe them in terms of a unique *physical* and *digital* identity, their *interaction primitives* and devices states. We showed that, even when important interaction events are shared between smart objects, state misalignments will occur. Good practise is to share all important user interaction events and events that cause state changes that are prominently perceivable by users. Using finite state machines to understand such behaviour appeared helpful.

The importance of being able to uniquely identify smart objects in both the physical and digital space, as well as sharing their interaction capabilities and states, was shown in this implementation and also the previous implementations. In this chapter we showed how it was grounded in the theory of interaction models by Nielsen, Card and others.

In our theory we describe semantic connections as meaningful relations that can exist, not only in between smart objects, but also between smart objects, people and places. Although we still believe this is interesting to consider, we have not implemented it in the sleep use case. It may be interesting for future work, including e.g. relationships between persons (digitally described in social networks) and investigate whether these different types of relationships should be transitive as well (e.g. person A is friends with person B, person A owns device a, and B owns device b, should we infer friendship between a and b?) Or, when device A is in the same room as device B and they are touched together, is it safe to infer data can be exchanged over a connection established without passwords?

We employed augmented and functional feedback and feedforward to help users to better predict the functional result of the connections they create. Functional and augmented feedback also showed to be key in maintaining the causal links between user action and function, distributed over interconnected smart objects.

A fundamental difficulty we encountered during the implementation of the feedback and feedforward (and which is a big challenge in interoperability in general), is what we call the *awareness paradox*. To foster emergent functionality, efforts are aimed at enabling smart objects to interoperate without their combined functionality being specifically designed. This makes the smart objects factually unaware of each other, exchanging information though an information broker. For the users however, it is imperative that smart objects show behaviour as to appear to be aware of each another. Since the connections that may be created during use are not known at development time, wise decisions have to be made by the developers on how to describe the interaction events and functionalities that are shared.

By describing feedback and feedforward of the semantic connections as a result of the match in capabilities and functionalities, and having the semantic transformers and sink objects (instead of the source) produce the preview and indicator feedback, we make sure that they are capable of displaying (i.e. in the widest sense of the word, not limited to the visual modality) this feedback. Our reasoning is that, if a sink can be the sink of a functionality, it should also be capable of giving feedforward and feedback for this functionality.

Moreover we showed that semantic connections and using semantic transformers to create services is an appealing idea, leading to additional and, more importantly, more meaningful functionalities of ensembles of existing devices. This may reduce the number of devices needed in our daily lives by reducing functionality redundancy.

After we implemented our theory we feel more confident that we defined a complete set of entities and properties to achieve our goal of creating interoperability between smart objects and to enable users to interact with the objects and the relationships between them at a semantic level. Even though our theory describes how to implement smart objects and provides the necessary core ontology, we believe enough freedom remains for designers and developers to create their own unique objects and interactions.

5.6 Concluding remarks

Our *semantic connections theory* provides a foundation for modelling user interactions with interoperating smart objects in smart environments, and therewith the possibility to improve the interoperability among them. Using this theory and the resulting interaction model, we implemented an example use case and refined the Connector prototype which was used to manage semantic connections. We put our theory to the test practically, to evaluate in terms of completeness, usefulness and applicability. Finally we carefully considered the notions of feedback and feedforward in our use-case and designed and implemented augmented feedback and feedforward to enhance perception of connectivity and the perceived causality between user action and feedback.

Summarising, we made the following observations and changed and/or added to the theory accordingly:

- we refined the definition of smart objects;
- we refined the definition of semantic transformers and the connections between smart objects and semantic transformers;
- we implemented direction in the semantic connections and observed its implications on information flow and user interaction;

- with the directionality and transitivity, smart objects are defined as sources, sinks and bridges, semantic transformers are also bridges;
- we implemented augmented and functional feedback and feedforward, introducing `PreviewEvents` and `IndicatorEvents`; and
- we implemented temporary connections between the connector and smart objects (defining a Connector object as a smart object itself).

With regards to feedback and feedforward, we made important steps towards defining how this important information is changed when devices are considered in a system of networked objects as opposed to the devices in isolation. These findings are explored in more detail when we describe a framework for designing for interoperability in chapter 6. We believe that our theory may help other interaction designers and developers to deal with design opportunities and challenges that emerge when designing for interoperability.

Chapter 6

Semantic Connections Design Framework

6.1 In this chapter

In this chapter we present a design framework to analyse and guide the design for interaction across products in an ecosystem of interconnected *smart objects*. The framework is based on our *Semantic Connections theory*, which was introduced in chapter 5. It translates the concepts of the theory into practical handles that can be used to design for interoperability. First we introduce a set of principles and conditions that are considered important for success. Second, a more practical set of handles is provided to help designers to achieve these conditions. The designs that are described in chapters 3 and 4 are then placed in the framework to show how the framework can be applied. Finally, the framework is evaluated by a small group of designers. They evaluated the framework in terms of understandability, applicability and usefulness. A discussion of the results conclude this chapter.

6.2 Designing for Interoperability: a Framework

Based on our semantic connections theory, the following subsections describe concepts and practical handles that should be taken into account when designing for interoperability. We identify nine guidelines, which are described in Sections 6.2.1 through 6.2.9.

6.2.1 Reversibility of actions and connections

The reversibility of a users actions are important when we want to enable users to explore the possibilities of exchanging information between smart objects by interconnecting them. As we can probably never give enough information to users, to fully understand the consequences of their actions, we should at least give them the possibility to easily undo what they just did. In particular, this guideline applies to the setting up and removing of semantic connections (the actions local to the smart objects are not considered in this framework).

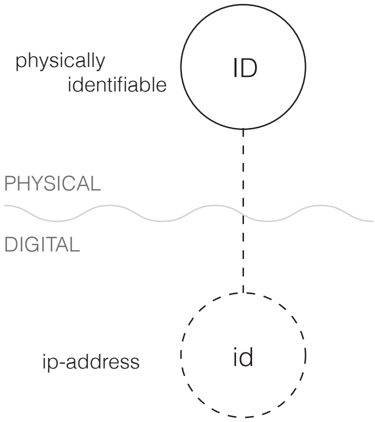


Figure 6.1: Every smart object is physically perceivable and identifiable, and has a digital representation

6.2.2 Physical identification of smart objects

Smart objects are physical, and therefore can be touched or pointed at. When users want to connect two objects together that they can see and touch, it seems unnecessarily complicated to select them by searching for a name in a list in the GUI of one of the devices and negotiating passwords. However, this is the way pairing often is performed currently. To allow for a more natural way to identify smart objects, this physicality of the smart objects should be exploited, by using touch and proximity *between objects* to identify and select them for making connections and exchange information. Every smart object *must* be physically identifiable and is also uniquely identifiable in the digital domain (Figure 6.1)

6.2.3 Source, connector and sink constitute a semantic connection

A connection can only exist between two smart objects. The meaning of the semantic connection is derived partly from the meaning of the *source* and completed by the meaning of the *sink* and therefore not fully defined until *source* and *sink* are both identified (Figure 6.2).

6.2.4 Directionality

Connections are directional or symmetric (bi-directional). Direction is a powerful concept in the hands of users as they may use it to express the desired direction of information flowing between smart objects. Directionality defines one smart object as the *source* and the other as the *sink* (Figure 6.3). To allow for mirroring or synchronisation of information exchanged between two smart objects, symmetry can be defined.

6.2.5 Transitivity

When connections have directionality and multiple devices (i.e. a minimum of three devices) are involved, devices can also act as *bridges*, transferring the connections due to transitivity, e.g.: If a music player A is connected to speaker B, and speaker B is connected to speaker C, speaker B acts as a bridge between the music player and speaker C (see Figure 6.3). A

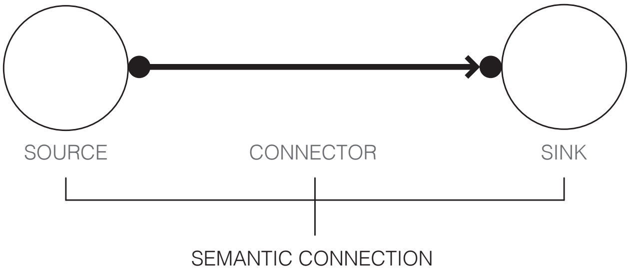


Figure 6.2: Source, semantic connector and sink constitute a semantic connection.

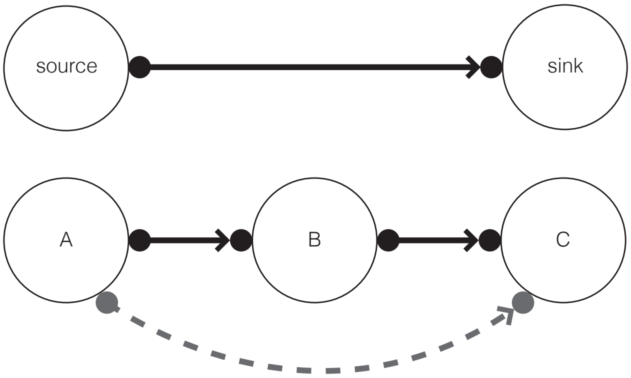


Figure 6.3: Every semantic connection is either directional or symmetric. Semantic connections are also transitive.

transitive connections between A and C is automatically inferred, however, should a users not desire the transitive connection, it can manually be removed in a similar fashion as removing any other connection.

6.2.6 People and places; structuring contextual information

Connections can exist between smart objects, people and places. Not only objects and devices have meaning in a system of networked devices—according to (Poole et al., 2008), physical location within the home and device ownership (or usage) are of central importance for understanding and describing home networks by users. Ownership can be seen as a connection between a device and a person. Connections from and to places or locations can be seen as a way of structuring contextual information such as location. Places in someone's home can be meaningful in terms of their use e.g., a particular chair may be associated with reading books, and these associations may be described and used as relevant context information. The relevance of physical space is not limited to places and locations,

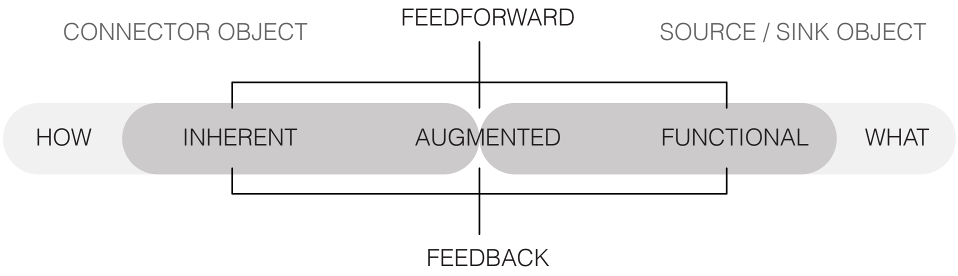


Figure 6.4: Inherent, augmented and functional feedforward and feedback in the context of the semantic connections theory.

as meaningful associations can also be made to areas within one's personal space e.g., in an interaction with a smart object, my left-hand side may be (temporarily) associated with specific content such as photo's.

For very personal devices (such as smart phones and laptops or tablets) we can, when these devices are used in an interaction, implicitly infer the user's identity. With shared devices, we may need a way to identify users. In such cases, making explicit connections from the device at hand to something personal of the user (e.g. a phone or key-chain) may be used as a way to indicate identity.

6.2.7 Permanent and temporary connections

Connections can differ in persistence. Connections can be made during an interaction cycle involving several devices to transfer content or data from the one device to another, and the connection then stops existing when the interaction cycle is completed. Connections can also be used to configure more permanent information exchange between entities in a smart space, much like setting up a connection to a wireless network router. These *permanent* connections will persist, and will be automatically reconnected every time the smart objects that are connected, co-exist in the same smart space. Temporary connections should be visualised or conveyed to a user as long as they are active. They are also used to trigger functional previews during the connection process as was described in Section 5.4.

6.2.8 Feedback and feedforward

Providing proper feedback and feedforward to a user is crucial to help them to make sense of the connections that are established, and to allow for the development of an effective mental model. We differentiate between inherent, augmented and functional information, adopted from (Wensveen et al., 2004). Figure 6.4 shows our interpretation of the different types of feedback and feedforward in relation to the semantic connections theory. A distinction is made between inherent, augmented and functional feedback and feedforward. Inherent feedforward reveals an object's action possibilities and inherent feedback is the return of information when acting on these possibilities. Augmented information can and may be used to provide additional information for this purpose. Together they inform a user about *how*

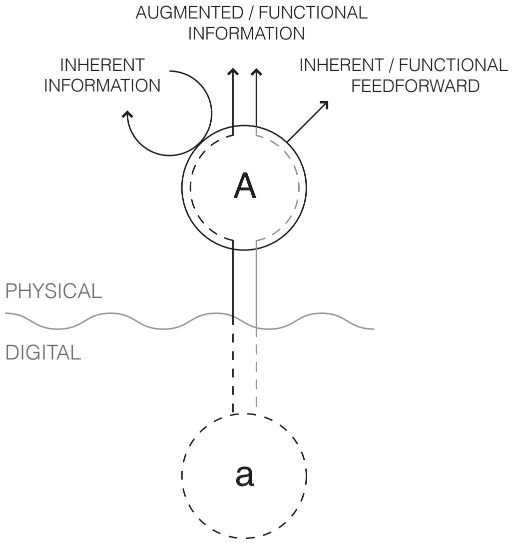


Figure 6.5: Feedback and feedforward provided by a smart object.

actions can be carried out with the object. Figure 6.5 shows the feedback and feedforward provided by a smart object.

For semantic connections, this information is predominantly provided by the connector object, in the process of making (or viewing/breaking) the connection (Figure 6.7). On the other hand, information about an objects function, *what* an object is, is provided by functional feedback and feedforward. This kind of information can be enhanced by providing augmented feedback and feedforward or, if functional feedback is missing or is unclear, may be provided by augmented feedback instead. Therefore functional information (what an object is or can do) is mainly provided by the source and sink objects. For semantic connections, the function of the semantic connection is defined by that of the source and sink objects. Their state/mode, the connection's properties (e.g. direction), and the functional result provides functional feedback. Augmented feedback and feedforward of the connection may be given to enhance the functional information (as is explained in Figure 6.6).

6.2.8.1 Inherent information

Inherent feedback and feedforward is information that is inherently coupled to the action itself, and appeals primarily to a user's perceptual motor skills (Wensveen et al., 2004). Inherent feedforward is information that reaches a user before the action takes place and informs about the action possibilities. Inherent feedback is information returned on a user's action and is a direct consequence of the action itself. Examples are the feeling of displacement or the sound a button makes when being operated. Inherent information is limited to information about how an action can be executed and does not necessarily communicate the purpose of the action. Because inherent feedback and feedforward are physical properties of an object or control, they are not directly applicable to a semantic connection, as a semantic connection is not necessarily physical and may be a mental construction. However,

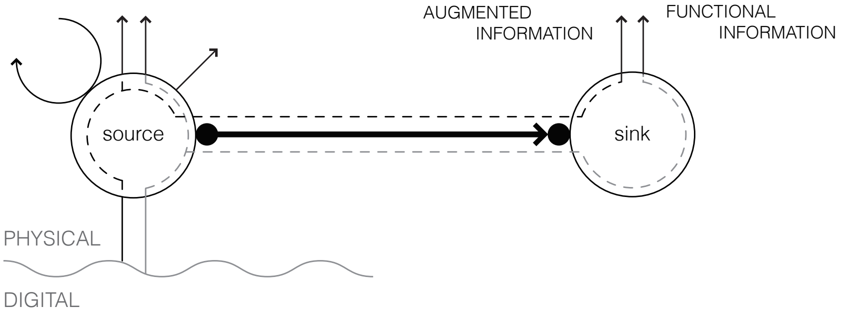


Figure 6.6: Information about what *a connection can do is provided by augmented feedback and feedforward, and functional feedback and feedforward at the source object.*

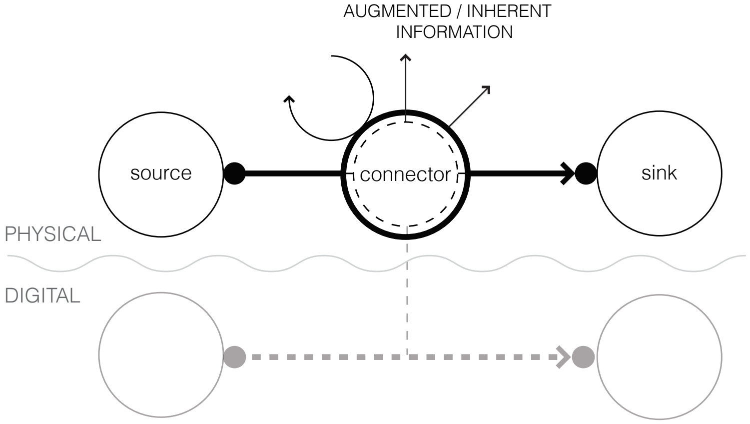


Figure 6.7: Feedback and feedforward provided by a connector object.

when users are given the possibility to explore and manipulate a semantic connection, by embodying it (permanently or temporarily), inherent information will be available, and can be carefully considered when designing.

First of all, inherent information may be found at the place where two objects physically interact. Like the male and female part of a connector, objects can be designed to fit together. This can be recognised visually, but also by simply trying to fit the two shapes together. The feeling of touching objects, sliding into one another with guidance and a clear snap may give inherent feedback on the establishing of a connection. Such physical properties can be considered when designing a *semantic connector*.

identifying the source and sink for a connection - designers can think of physical touch, fitting, and the proximity of or between objects as both feedback and feedforward.

confirming or making the connection - designers can think of physical qualities that invite for actions (feedforward) and give feedback associated with confirming or making a connection.

viewing a connection - e.g. physical qualities, icons, symbols and metaphors that invite for viewing or exploring (e.g. hand loupe, binoculars) for a viewing action can be considered.

breaking a connection - similarly, physical qualities that are related to breaking or unplugging can be considered.

using a connection - physical qualities of the source object may be modelled using *interaction primitives* (Section 5.2 and 5.3.2.2) to preserve the meaning of the interaction in relation to its (remote) function.

6.2.8.2 Augmented information

Augmented information comes from an additional source and thus not directly from the action itself (inherent feedback) or the related function (functional information) (Wensveen et al., 2004). If there is no immediate link between action and function (e.g. functional result is delayed, information is given about an internal state change), augmented information can be used to provide this information. In our semantic connections theory, augmented information is important information from the underlying digital state of the connection. It should be used by designers in cases where the action and functional result do not coincide in time and/or location, but also more generally to indicate that a user's physical actions had its effect in the digital domain. As is shown in the semantic connections interaction model (Figure 5.3), augmented feedback is feedback from the digital domain, augmenting physical space. This allows designers to use this type of information not only when designing a semantic connector, but also as feedforward to inform users what to expect as the result of a connection or to give feedback if there is no immediate functional feedback. Augmented feedback and feedforward can thus be provided by the smart objects themselves in the modality that is supported by its interaction capabilities. Designers should aim for maintaining the modality of the augmented information across the smart objects. Additionally, ensuring that the feedback occurring at distributed objects coincide in time and/or location will strengthen the link. In summary, augmented feedback and feedforward may be used to:

identifying the source for a connection - provide augmented feedforward about smart objects that can be connected, provide augmented feedback on the current mode the smart object is in, and provide augmented feedback to confirm successful identification;

identifying the sink for a connection - provide augmented feedforward about smart objects that can be connected to the source, provide augmented feedback to confirm successful identification and provide a functional preview;

making the connection - provide augmented feedforward to indicate the result of making the connection, provide feedback to confirm a successful connection and/or provide feedback when no immediate functional feedback is available;

viewing a connection - provide feedback on the existence of a connection, other than showing its function;

breaking a connection - provide feedforward to inform a user on the consequences of breaking a connection, confirm breaking the connection;

using a connection - provide augmented feedback when no direct functional result is available.

6.2.8.3 Functional information

Functional information informs the user on the direct functional result of an action. Functional feedforward goes beyond showing the action possibilities of a product and informs the user about the more general function of a product. This type of information often relies on association, metaphors and the sign function of products and are described in theories such as product semantics (Krippendorff, 2006) and product language (Gros, 1983). When a user performs an action and the device responds with information that is directly related to the function of that product (lighting switching on when a light switch is operated), we speak of functional feedback. When a device has more than one functionality, functional feedback should be viewed with respect to the users' intentions and goals when performing the action (Wensveen et al., 2004).

For semantic connections, functional feedback and feedforward can only be considered for the combination source, sink and the connector. The source object has functional feedforward (or semantics) that may communicate its function. Only when both the source and sink object have been identified, functional feedforward is available for the semantic connection. Important to note is, that the functional feedforward is derived from the junction of the functional feedforward (or function) of both the source and the sink (Figure 6.8). This is also problematic, as both source and sink may be multifunctional. If this is the case, users should make explicit what information or data they want to exchange by selecting the desired mode on the source object (e.g. selecting the alarm application on your smart phone to share the alarm time or go to a picture viewer when pictures should be exchanged). If this is not possible, or a multifunctional smart object is connected when it is in idle mode, semantic reasoning will be used to match all meaningful capabilities of the source and sink objects.

In many cases functional feedback of a semantic connection is obvious (e.g. hearing sound from a speaker that was just connected to a media player, seeing photo's on a TV when connected to a smart phone). However, functional feedback may only be available at another place and/or at another time. If we for instance take the example of synchronising your phone's alarm with your alarm radio, the real functional result may be hearing the alarm radio play at the alarm time you set on your phone. Interaction designers should make sure that setting the alarm already gives suitable feedback, so a user can be sure to trust the alarm to go off the next morning.

With the source and sink objects located at different places, interaction designers should make sure that the functional feedback is visible when the connection is made. Therefore, the order of connecting two spatially separated objects together is important, making sure that actually making the connection happens in proximity of the sink, so the functional feedback can be observed. In summary, when thinking about functional feedback and feedforward, the following considerations are important:

identifying the source for a connection - the source provides a part of the functional feedforward of the connection; it limits the possible uses of the connection. When

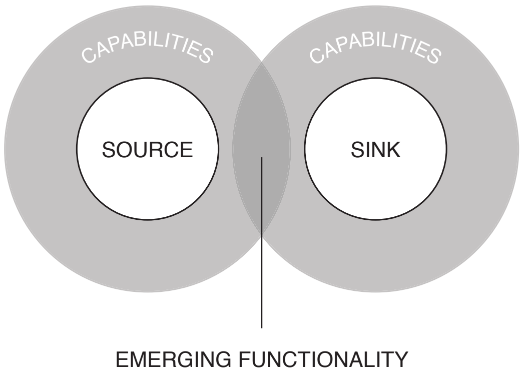


Figure 6.8: The functional feedforward of a semantic connection is derived from the junction of the feedforward of both the source and the sink.

the object has multiple functionalities, its current mode defines the function;

identifying the sink for a connection - given the functional feedforward of the source, the sink provides functional feedforward for the connection. Designers may provide augmented feedforward to communicate the functional result of the connection (functional preview), before the connection is made;

making the connection - when the connection is made, functional feedback is the functional result of the connection. When the functional feedback is delayed, augmented feedback should be provided;

viewing a connection - once connections are established, users will get functional feedback of the connection when acting at the source object;

breaking a connection - provide feedforward to inform a user on the consequences of breaking a connection, breaking the connection is confirmed by the absence of functional feedback;

using a connection - provide functional feedback to confirm the users action.

6.2.9 Modality, Time and Location

Another obvious and for our framework relevant subdivision of feedback (and also feedforward) is its sensory modality. This is particularly relevant when feedback is transferred over semantic connections. Aspects like continuity and discontinuity of modality, time and location are important to consider. During our endeavour of exploring the interconnection of smart objects, we often encountered the importance of objects sharing the same physical space and the implications this has on perception. For example, when an audio source is playing audio at multiple loudspeakers that are co-located in the same physical space, their outputs together create a combined music experience. When music is playing on a mobile phone with speakers of quality inferior to the loudspeakers in the same room, it may be desired to mute the speakers of the phone to increase the overall quality of the music.

However, when the phone is taken to another *location* or earplugs are plugged in the various audio sinks, physically *separating* the audio spaces, the audio signals stop interfering and should be made available at the sinks again. The importance of location also suggest that it may be good practice to give feedback to the user about the smart object's location awareness.

Let us consider another example from our implementation (as described in Section 5.4). An alarm that is set on a smart phone, is synced with an alarm radio and also connected to a dimmable lamp. The functionality emerging from this configuration is that, when the alarm triggers, the phone displays the alarm and vibrates (muting the audible alarm) and the radio starts playing music. Thirty minutes before the alarm time the dimmable lamps starts slowly increasing its light intensity, until reaching its maximum (pre-set) value at alarm time. The alarm on the phone is muted as its audible alarm would interfere with the music from the alarm radio. When the alarm is snoozed or dismissed on the phone, this is reflected at the radio. The lamp however is unaffected by this as a snooze functionality does not make sense for a wake-up light which aim is to gradually wake a person up[1]. Moreover, the wake-up functionality attempts to replicate a sunrise (at least metaphorically) and one cannot snooze the sun.

An interesting issue arises with the vibrating alarm. Despite the principal difference in modality between the tactile vibration alarm and the audible alarm, they can cause annoying interference in the auditory modality, as vibrations produced by the phone are also audible. Now, the exact modelling of the alarm capabilities of the phone, influences the inferences made, as the vibration alarm being modelled as both belonging to the auditory and the tactile domain, will cause the vibration alarm to be disabled as well.

For indicator events i.e. augmented feedback, time, location and modality are also important. When indicators are given to confirm an auditory connection between a audio source and one or more audio sinks, it would be natural to maintain the modality and provide audible indicators. However, because of the spatio-temporal character of the indicators, they may have to be at slightly different times when in the same space to make sure they can be identified distinctively (e.g. when a source is connected to three sinks, an indicator may be sounded at the different sinks at a different time so a natural sequence can be perceived). When smart objects are distributed over different locations, unity in modality and time is less important, and may be even impractical. when the sink and source are at different locations, the confirmation at the sink should be preferably in a non-temporal form (e.g. a visual indicator) and should stay visible until it has been dismissed (or expired).

Summarizing, based on the observations made during implementation and testing we propose that:

- feedback should preferably be provided in the same modality as the functional result,
- when connected smart objects are in the same space, feedback should be provided at the same time,
- when connected smart objects are at different locations, feedback should (preferably) be given in a non-temporal modality,

---

[1] The original wake-up light as is commercially available from Philips behaves similarly. Although snooze functionality is available for the build in audible alarm and alarm radio, the light is unaffected even for dismissing. The light should be manually turned off in the same manner as one would if it is used as a bed light.

- when connected objects are in the same space, and the perceived function (or feedback) is in the same modality, interference should be minimized, and
- when functional feedback does not correspond in time, location or modality, augmented feedback should be employed to maintain the perceived link.

6.3 The framework in action

The framework aims to:

- guide designers when designing *smart objects*
- help designers to design *interfaces* to semantic connections
- help designers and developers to design for interoperability, creating a natural and consistent *user experience* for ubiquitous computing environments

It aims to help designers to approach the design of products from a connection centred point of view, inviting them to think of how their designs fit in the larger ecosystem of smart objects and understand what kind of issues arise when coupling action, perception and meaning. In this section, the designs from Chapter 3 and 4 are reviewed again by placing them in this framework. Doing so, we hope to show the relevance of the framework and identify potential improvements to the proposed designs.

6.3.1 Interaction Tile

The Interaction Tile physically identifies smart objects by means of their proxy objects or directly (for mobile devices). Although using proxy objects is not preferred, there will be a trade-off between remote and local interaction (Section 3.6.4) or the like.

The central tile physically represents a semantic connection, however, the semantic connection can only be defined when two or more proxies are aligned. In the case of three or four proxies, the Interaction Tile represents *multiple* semantic connections, of which each consists of source, connector and sink. Note that the notion of source and sink are not very relevant for the connections made by the Interaction Tile, as smart objects are not explicitly defined as such.

In terms of feedback and feedforward, functional feedforward (design semantics) of the smart objects is available, to convey the *what* (Figure 6.4). Functional feedback is also available, e.g. music that starts/stops playing when the connection is established/broken. Inherent and augmented information signifying *how* a connection can be made is provided by the Interaction Tile. Inherent information is available for where to place the proxies at the Interaction Tile. Augmented information is available to indicate: whether a proxy is recognised, whether a connection is possible and whether a connection already exists. An obvious flaw in the design is the missing feedforward on how to establish or break a connection. When the action of making/breaking is performed, feedback is provided auditory (for making) and tactile (for breaking). Additionally, when the Interaction Tile is placed back in alignment with the proxies, a confirmation of the action is provided and feedback is given on the *digital* state of the connection (this also serves as feedforward, informing users on the result of another make/break action). Note that the feedback and actions are *symmetric*, making it easy to undo an action by performing the same action again.

The Interaction Tile also enables users to explore transitive connections. When users connect device A to B and B to C, they will also find a connection between B and C when it is investigated with the Interaction Tile. The Interaction Tile does not make a distinction between temporary and permanent connections.

6.3.1.1 Insights and possible improvements

Placing the Interaction Tile in the framework reveals several possibilities for improvements. Although inherent feedback is available for where to place the proxy objects and augmented feedback to indicate it was recognised, there is no feedforward on how to make or break a connection. The action to make and break the connection should be designed so it better fits its purpose. The Interaction Tile is also very limited in expressive power, which could be improved by adding more detailed information about the connection (e.g. direction, persistence). Additionally, temporary connections could be established when the proxies are aligned with the Interaction Tile to allow for quick data exchange, without having to conform and make the connection (permanent). Finally, functional previews (augmented *functional* feedforward and augmented feedback) should be provided at the smart objects to more clearly communicate the meaning of the connections.

6.3.2 Interaction Tabs

An interaction tab stands for the smart object it represents, and can in this way directly be used in the interaction. There is clear inherent feedback and feedforward, on how to act (aligning the tiles) and whether the tabs have been aligned (the feeling of two tabs touching). This is enhanced by augmented information, showing whether a connection between the tabs has been established. Therefore, there is no distinct mode for exploring possibilities, a connection whenever possible is directly established. Functional feedback is also available, e.g. music that starts/stops playing when the connection is established/broken.

In the Interaction Tabs design, there is no explicit notion of a *connector*. This can be looked at in terms of the framework, as if the endpoints of the connector are permanently linked to the smart object and therefore do not have to be identified. As a result, semantic connections can not exist without two tabs being physically aligned, and a transitive connection cannot be viewed.

Similar issues occur when smart objects have a capability to physically identify other smart objects (e.g. mobile devices with integrated NFC technology). Conceptually, one endpoint of the connector is permanently associated with the smart object, and only one other smart object needs to be identified to establish a connection.

6.3.2.1 Insights and possible improvements

In view of the framework, Interaction Tabs support temporary connections rather than more permanent ones. This is the direct result of the way smart objects are identified and connections are established. To make a distinction between temporary and permanent connections, a way to physically fix the tiles together could be employed (which would also prevent accidentally disconnecting) or another action to establish a connection. Tabs could also temporarily represent smart objects (or a part thereof, e.g. a control) which could then be used in the interaction in a similar way. Decoupling physical connection state from digital state will also enable viewing existing connections and transitive connections. Finally, similar to the Interaction Tile, functional previews (augmented *functional* feedforward and augmented feedback should be provided at the smart objects themselves to communicate the meaning of the connections.

6.3.3 Nodes

The Nodes design takes a different approach by physically representing the endpoints of the connections, and associating the endpoints with the smart objects they are connected to by means of proximity. The Nodes define direction explicitly by means of defining source and sink at the nodes themselves. A downside of the very explicit way direction is defined is that defining symmetric connections is currently cumbersome, as the reverse connection should be made in the same way. Additionally, the way connections are established are tapered for making rather permanent configurations, and the actions required are not very suitable for making temporary connections.

The Nodes employ inherent feedback and feedforward. Nodes are placed close the the smart object for physical identification, providing direct inherent feedback. The arrow-shaped flags of the nodes signify source/sink, and by pointing the endpoints each other, users connect (also *mentally*) the two devices together. This can also be classified as inherent information. The design however lacks augmented feedback. When a connection is established, the only available feedback is functional feedback and there is no information available about the success of the connection (or digital state of the connection).

6.3.3.1 Insights and possible improvements

Augmented feedback may be considered to create a better link between the digital states and the physical appearance of the nodes. Augmented feedback could be considered to confirm: the link between smart object and node, digital state of the connection, and for the smart objects themselves to indicate/confirm the actions on the source objects at the connected sinks. Functional previews may also be considered. Finally, a way to easily insert reverse connections for symmetric relations may be added to improve usability.

6.3.4 Connector

The Connector physically identifies smart objects by *scanning* and selecting. In this process, inherent and augmented feedback and feedforward is available: When the tag is in the range of the Connector's RFID field, augmented feedback is provided. After holding the Connector over a device tag for a moment, a sequence starts, lighting up the second, third and fourth segment of the inner ring. This can be seen as feedforward to hold the Connector over the tag until it has been selected and all four segments are lit.

A distinction is also made between selecting a source or a sink object, as the inner circle represents a source and the outer ring stands for a sink. When a sink object is selected first, and it is not currently connected to a source, the connector also indicates this by flashing the outer ring red. The digital status of the connection between the identified source and sink objects is also clearly indicated, as was depicted in Figure 4.6c.

Inherent information is available: the design of the lower part of the connector was intended to invite for scanning/touching actions. And the connector employs physical cues about the possibility to pull the upper and lower part apart or compress the connector. A tangible and audible click when the action is successful serves as inherent feedback[2]. Functional feedback is also available, e.g. music that starts/stops playing when the connection is established/broken.

To strengthen the link between source and sink during the use of the connection, actions at the source either cause a functional result at the sink or trigger augmented feedback at the sink to confirm this action. Users perceive this action—reaction and mentally (re)construct the connection[3].

A functional preview of the connection is also available to serve as functional feedforward (as was described in Section 5.4.3.1). Since a temporary connection is created as long as the connector indicates a connection possibility, immediate functional feedback may also be perceived (e.g. when music is playing at a the source and the sink accepts music, it will start playing the same song).

6.3.4.1 Insights and possible improvements

They way the connector is currently used to express directionality (i.e. the order of selecting the smart objects, identifying the first as source and second as sink), leads to seemingly unnecessary actions when a user wants to create a symmetric connection, as in the current implementation, a user has to manually insert the reverse connection. Similarly, when a user selects a smart object that is the sink of an existing connection and can also be a source, there is currently no possibility to express which possible *option* is the one desired. Therefore, a redesign is required, or a kind of selection mechanism should be added. Selection may also be employed to allow users to express what functionality they want to be connected, in a way other than using the mode/state of the source at the time a connection is made.

6.3.5 Spotlight Navigation

Spotlight Navigation physically identifies smart objects by pointing gestures [4] Our intention is to recognize the identity and physical location of each device and to provide augmented feedback once a device is recognised by projecting a frame around or icon close to the smart object. Inherent, augmented and functional feedforward are available for the operation of the Spotlight Navigation device. The device has design semantics (although subtle) of a projector, with a lens that produces a light beam. By pointing the projector around like a flash light, users can explore smart devices (with a dot-shaped pointer) and existing connections between them. Connections are created by drawing lines between the devices. An erasing gesture with the Spotlight Navigation device pointed at an existing connection, breaks the connection. Feedback is provided when a line is drawn (a yellow line, originating at the first object), and will indicate a possible connection once the endpoint approaches the

---

[2] The click of the prototype may have been too subtle, as one of the participants broke the prototype because he did not clearly perceive the click and continued pulling, eventually pulling too hard to break it.

[3] this was observed during the evaluation as is described in section 4.8: users perceived a connection to exist between the coloured lamp and the other non-functional lighting in the room

[4] Our first Spotlight Navigation prototype has no capability of knowing where a specific device is located in the real world. We were assuming that a detection- or setup-step had already been performed, from which Spotlight Navigation learned in which direction it would find the respective devices.

second object by turning green. Releasing the button on the Spotlight Navigation device establishes the connection.

Note that there is no sense of direction in the connections, there is no difference in the result when the order of connection two smart objects is changed. Our current prototype projects transitive connections in the same way as it projects regular ones. There is no notion of direction or a distinction between permanent and temporary connections.

6.3.5.1 Insights and possible improvements

Currently, Spotlight Navigation projects icons of the smart objects, onto or in proximity of the location of the smart object in physical space. This may be replaced by augmented feedback, such as a rectangle around the smart objects. Using projection, additional information like capabilities that can be connected can be projected as well. With projection capabilities, providing (augmented) functional feedforward becomes easier, even for devices without display.

Directionality may be obtained by the order of selecting source and sink objects and can be visually indicated. A additional action could be used to indicate a symmetric connection and transitive connections may be projected differently from regular connections (e.g. different opacity, dotted line).

The persistence of a connection may differ according to the framework. A distinction can be made between permanent and temporary connections, where permanent connections persist when objects exit and re-enter a smart space and temporary connections do not get re-established. However, Spotlight Navigation also offers the possibility to approach temporary connections in a conceptually different way. Items (e.g. calendar items, alarms, images, video clips, etc.) could also be exchanged directly, virtually dragging them from the one device to another, without leaving a connection after the action is finished.

6.4 Evaluation

To evaluate the framework, we invited a group of six designers to; (1) watch a demonstrations of the set-up as was described in chapter 5, asking them to analyse the system using the framework, and; (2) based on this experience, select either a part of the system (e.g. one or more smart objects, a specific type of feedback), an existing (related) product or something they are working on themselves to apply the framework and do a (re)design. The designers were given the freedom to select something themselves based on their professional opinion of how they thought the framework could be applied best.

6.4.1 Participants

Designers with different backgrounds and current occupations were selected. They were selected from the personal and professional network of the researcher. Their age ranged from 27 to 31 years old, and they were all male. Four participants hold a BSc and a MSc degree in Industrial Design (Eindhoven University of Technology) and are currently working as researcher or PhD candidate at the Department of Industrial Design. One participant has a BSc in Media technology and a MSc in Industrial Design and is currently working as an Interaction Designer/User Experience Designer in industry. One has a computer science and user interaction background, and is currently in the final year of the post-graduate programme User System Interaction (USI) at Eindhoven University of Technology. We acknowledge that selecting participants from, or with a relation to our home faculty introduces a certain selection bias. To limit this, the participants that currently work on our university were selected such that they came from different capacity groups within our faculty. Their (professional) design experience ranges from 2,5 years to 5 and self-report familiarity with designing interactive systems ranged from 3 to 5 on a 5-point scale. Two of the designers were familiar with the Frogger framework, on which the framework that was evaluated is based.

6.4.2 Methods and Procedure

The designers were handed Section 6.2 of this chapter. Examples that discuss the sleep use case were removed to keep the framework on a more abstract level, and not explicitly discussing how it can be applied to the use case. Designers were asked to read the document on beforehand, or during the start of the session. They were in the opportunity to ask for additional explanation when parts were unclear, and the key concepts were also explained verbally at the beginning of the session.

The designers were asked to fill out a short pre-evaluation questionnaire and were briefed about the procedure of the evaluation session. After discussing the framework the designers were allowed to ask questions about parts that were unclear, making sure that they understood the framework and the used terminology. The briefing and discussion of the framework took about 15 minutes, depending on their initial understanding of the framework.

With the framework in mind, the designers watched a demonstration of the sleep use-case (Figure 6.9) as was described in Section 5.4. The focus of their observation was to recognise the different types of feedback and feedforward (as described in the framework), understand the connections that were explored and manipulated, and to understand how these connections influence the available feedback and feedforward. They were asked to pay specific attention to the details of the interactions (e.g. pay attention to displays, lights, sounds, etc.). The designers were allowed to take notes during the demonstration and ask for repetition of actions were necessary.

The evaluation itself comprised of two parts: First, the designers were asked to reproduce the *designer* model[5], with the help of the framework, focussing on the smart objects, the connections between them, and information available in the form of inherent, augmented and functional feedback and feedforward. The designer was asked to make drawings of the model and was invited to use the notations used in the framework. The designers were given 20–30 minutes to finish the first part of the evaluation, including the demonstration.

Secondly, the participants were asked to apply framework, to:

- redesign an existing related product;
- redesign a part of the demo;

---

[5] I use the term *designer model* to indicate the mental model a designer develops when watching a demonstration, trying to understand the design from his/her professional experience. Norman (1998) uses the term designer model to describe the model a designer holds of the design, opposed to the user *mental model* and the *system model*. Therefore, the use of designer model here, is somewhat different from its original use, as it does not describe the model a designer holds of his/her own design.

Figure 6.9: Setup as was used in the evaluation.

- add yet another smart object to the demo set-up and explore/design the necessary feedback and feedforward, or;
- apply it to their own work/current project and evaluate what insights the framework could possibly lead to.

After that, feedback was asked from the designer on the use of the framework in terms of its understandability, applicability and usability. Each session was aimed to take no more then 1 to 1,5 hours in total.

6.4.3 Results

All designers were able to understand and utilise the framework. Remarks were made by the designers that the framework was rather abstract at first, but that seeing the demo and doing the analysis helped them to understand it better. The latter is important, because applying it (generatively in particular) requires a good understanding of the concepts involved. Three of the designers felt the need for a single-page graphical schematic overview of the concepts and guidelines, to be able to use during analysis and as a generative tool during the design process.

6.4.3.1 Use-case analysis

The analysis of the sleep use-case according to the framework resulted in a series of drawings (with a minimum of one per participant), which were used by the designers to explain their

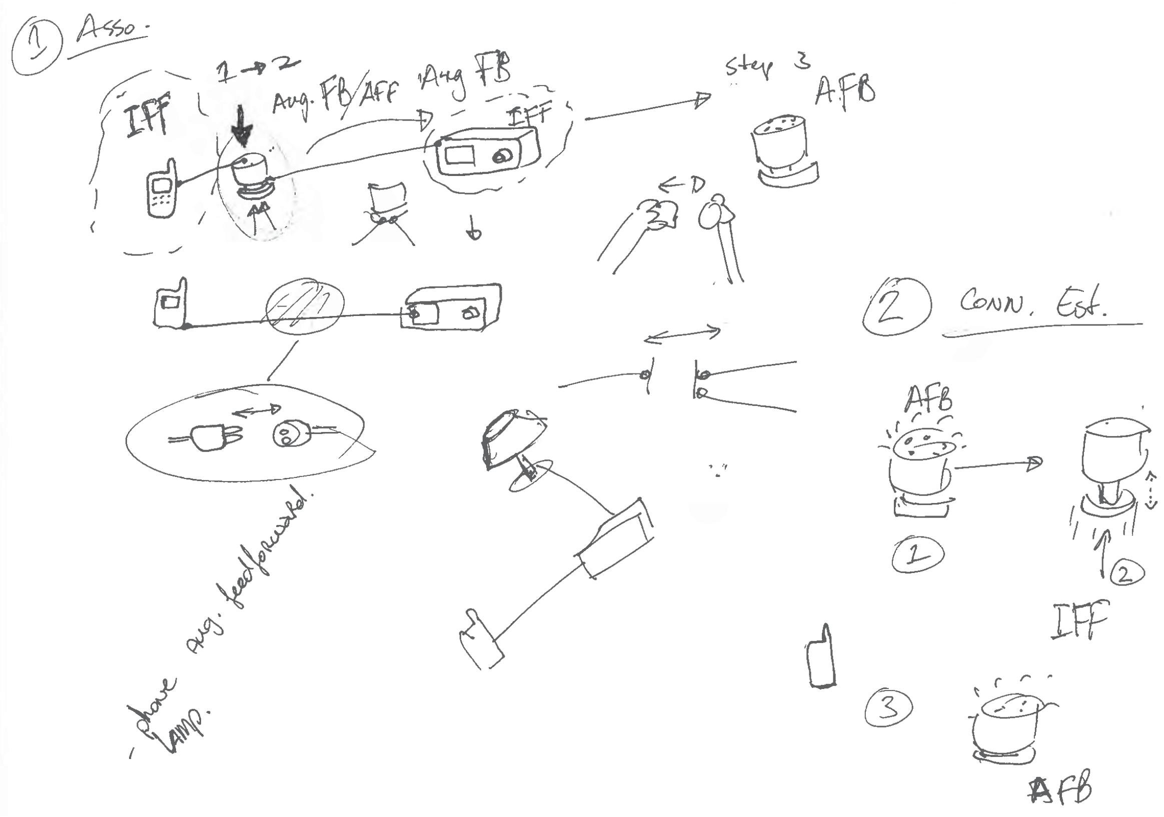


Figure 6.10: Example of one of the collected drawings.

analysis of the demonstration to the researcher. The most important results are described in this section. Figure 6.10 shows an example of one of the collected drawings.

The designers discussed the existence of feedback and feedforward for selecting the smart objects. One designer indicated the absence of clear feedforward on how to act with the Connector. Others indicated that feedforward existed (although not very clear) in the form of a surface that invited to touch other objects, and visible tags on the smart objects. One designer went as far as calling it a form language (i.e. the Connector's bottom surface together with the tags on the smart objects), indicating *connectability*. Opinions varied on whether this feedforward was of the inherent or augmented type, three decided on augmented feedforward, two on inherent feedforward. The point of discussion was whether the information was available in the possibility to touch the objects together or whether recognition of this possibility was based on knowledge (e.g. understanding the concept of scanning tags). The fact that the tags were added to the smart objects was another reason for classifying it as augmented information. The general tendency was that the available feedforward for selecting was not very strong, however, in combination with the feedback of the connector while selecting (lights lighting up in sequence), there was sufficient information.

The information given on the Connector's display rings was recognised as augmented information. Feedback on the recognition of a smart object, and feedforward on how to proceed (i.e. holding it on the smart object until all lights are lit). The two different rings were also considered to indicate that there were two steps involved in the paring action, which was also considered a kind of feedforward. The meaning of the two rings, indicating source (inner) and sink (outer) was not clear to all designers, as only two designers recognised this. Although most designers agreed that the connector mainly provided augmented feedback (and feedforward), two designers recognised the feedback after establishing a connection as functional feedback, as the Connector's function is connecting two devices together.

Overall the designers found the feedback provided by the Connector clear (i.e. provided at the right moments and easy to understand), but its meaning was not always clear at first. In particular, the difference between finding an existing connection and a connection possibility only became clear after experiencing both. The (inherent) feedforward for pushing and pulling the connector to connect and disconnect was recognised by three designers. Inherent feedback (clicking) as a confirmation of connecting or disconnecting two objects, was recognised by all designers, as well as the augmented feedback that (i.e. lights going from green to off or turning red). Clear functional feedforward for the connector was indicated to be missing by two, the other designers did not mention functional feedforward for the Connector in their analysis at all.

A few, more general remarks were made that are worth mentioning: One designer pointed out that there was no difference between scanning smart objects that are connected and smart objects that are not. The suggestion was made that quickly blinking a screen or light, or sounding a beep on the connected smart objects, while scanning would improve the current design. Another remark was made about scanning a sink as the first object. In such a case, feedback is given to indicate that the object cannot be selected, but no feedforward exists on how to act next.

Apart from the Connector, the other smart objects (i.e. smart phone, internet radio and dimmable lamp) also provide feedback and feedforward. The phone, which can be used to set an alarm and play music, provides mainly augmented and functional information. This was generally recognised by the designers, but was not discussed in detail, as this is not any different from a normal (Android) smart phone. Remarks were made about missing feedback on the status of a connection on the smart objects itself (two participants). Feedback on the connection's status is only given after an action is performed (e.g. setting the alarm, playing a song), but no feedback is available before or during an action.

For the internet radio and the lamp, the wake-up light functionality, alarm and payback of music were recognised as functional feedback. The information provided by the smart objects as a functional *preview* showed to be more difficult to denominate. The preview was recognised and classified as augmented feedforward by three designers and one designer indicated the feedforward as functional. Two others described it as being augmented feedback. Providing information about the connection's function, as feedback in the process of making the connection (as the preview occurs after two smart objects are selected, when the connector shows a connection possibility). The same issue surfaced with the functional preview of the internet radio.

Generally, participants showed to have difficulties making clear distinctions between inherent and augmented, and augmented and functional information. Differentiating between feedback and feedforward also seemed difficult, as participants indicated that information often had two roles, giving feedback on an action on the one hand, and giving information about what to do/expect next on the other hand. This is in correspondence with the intentions of the Frogger framework (Wensveen, 2005) where feedback on certain actions can be viewed as feedforward for other actions.

The connections between the smart objects were clear, as well as the concepts of source and sink, directionality and transitivity. Source-sink relationships were clear (due to the feedback) although it was expected by four of the designers that the internet radio could also be a source for the connection with the phone[6]. They understood the link between scanning order and the direction of the resulting connection. Two designers also recognised the importance of context such as location and moving from one location to another.

The designers felt that the feedback strengthened the perception of connectedness, in particular the visual links of synced alarm-times when an alarm was set, the album art appearing on the screens of both the internet radio and smart phone. The fact that this feedback happens at about the same time strengthens this link. However, one designer also pointed out that having the feedback at the same time is not always convenient, as one can only look at one place at a time. Redundancy in feedback was also pointed out (i.e. at the Connector, at the source and at one or more sinks), and that, although sometimes necessary, there exists the risk of cluttering the interaction space.

6.4.3.2 Redesign

The participating designers selected different design tasks for the second part of the evaluation. To illustrate how the designers thought they would apply the framework, their ideas are discussed briefly in this section.

One designer explained how he would use the framework to design a system of inter-operating devices, by using the source and sink concepts as well as the different types of feedback and feedforward in a *sequence diagram*[7]. He then used this diagram to reconsider the feedback that is given when connecting the phone to the lamp.

Another designer used the framework to think about how to use a second screen (e.g., web tablet) in combination with a TV. Figure 2.4 in Section 2.2.4 shows design patterns of how to deal with multiple-screens. The designer explored how to interact with a second screen (sink), while you are not in the same room as the TV (source). In particular, he looked into the situation where the second screen should *mirror* the same content as the TV in the living room, while being away. Design problems he identified included: How to deal with inherent and augmented feedforward for connection possibilities between devices that have not yet been connected. And, how to design inherent and augmented information for devices that can have multiple connections. He expected that the framework could provide guidance to find answers to these challenges.

One designer chose to redesign a part of the demonstrator, in a slightly different setting. A concept sketch of the redesign is shown in Figure 6.11. He combined the dimmable lamp and the radio into one device (similar to commercially available wake-up lights). He then split the interaction of setting the alarm on the phone, and connecting the phone to the wake-up light across two locations (living room and bedroom). The framework was used to explore what feedback/feedforward should be available where and when. Interesting questions arose, such as: how do you give feedforward properly when there are action possibilities with another device that is not in the same room? And, what happens with an alarm that is set on the mobile phone, before it is connected to the wakeup light?

The remaining three designers applied the framework to their own project. The projects were about: the design of a smart parking management system for people with multiple cars living in areas with limited (public) parking space available; designing alternative ways

---

[6] Although this is possible and maybe even logical, it was not implemented in the demonstrator.

[7] A sequence diagram is an interaction diagram as a part of the Unified Modelling Language (UML).

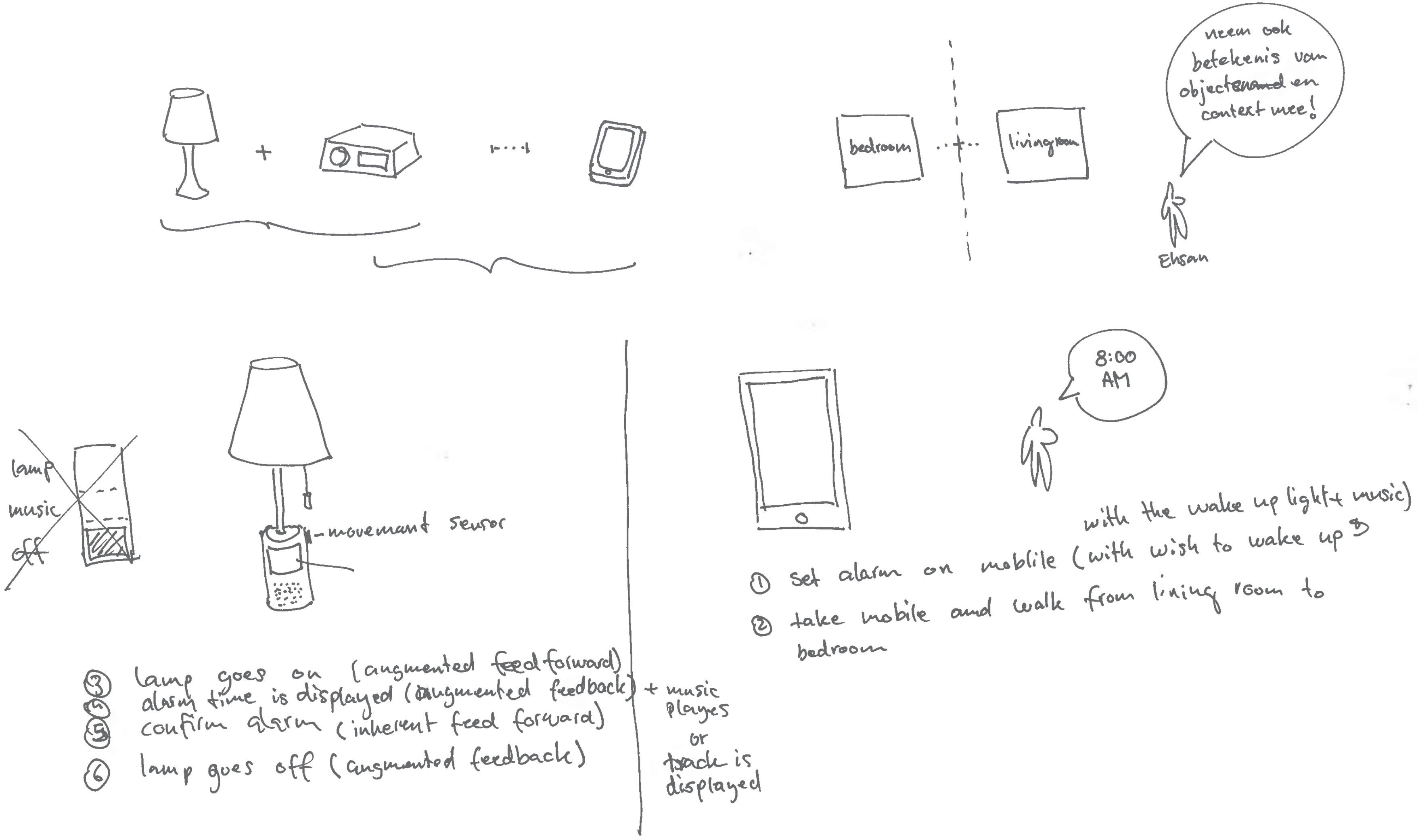


Figure 6.11: Concept sketch of a redesign that was done during one of the sessions.

of interacting with lighting in a smart office setting, and; interactive smart textiles, communicating how the functionality of the textile changes when connecting it to (different) smart objects in the environment.

For the parking management system, the design challenge at hand was to manage parking rights for a first, second and third car owned by one household, based on the availability of parking places in the area. Applying the framework led to interesting insights in the relationships that exist between the entities that are involved in the system, as well as insights in feedback from the system towards its users. Semantic connections exist in between the various smart objects in the system (i.e., devices, sensors, etc.) but also in between entities such as car owners, car users and parking rights. Identification appeared to be a key issue, and the framework gave insights in the various possible ways of identifying a person or car, and its implications for the users.

For example, using the car's identity to determine parking rights would allow different people using the same car to have equal parking rights. Using a person's identity would give this person equal parking rights, even when he or she would be using a rental, borrowed or shared car (e.g. through a car sharing service). The latter identification option could be a conscious choice to foster car sharing. Introducing (a) special identification object(s) that could be used by any member of a household, in any car, would be an option to leave the parking rights management to the members of a household themselves.

Semantic connections also exist between smart objects like the car, a (smart) parking space, a person's smart phone, etc. Feedback and feedforward also play an important role, e.g. feedback on parking space availability, how one is being identified by the system, possible parking costs (for second and third car) and alternative parking places in the area.

The designer working on the interactive lighting project explored the use of semantic connections to control various (smart) lights in an environment. By making combinations between light controllers, sensors, user interfaces and lights (or groups of lights), users can easily configure the lighting in their environment. By defining sensors (presence, light intensity) and user interfaces as *sources* and the lights as *sinks* or *bridges* (combination of source and sink) users can have control over the behaviour of the resulting configuration. Lights can be grouped in a serial fashion, and such a group can then be connected to another group or a UI by just connecting to one of the lights. Semantic transformers can also be used, to invert mappings (on→on to on→off or off→on), or do other kinds of generic transformations that are useful for mapping lights and sensors. Configurations may also be stored in pre-sets or certain settings could be linked to the use of a specific smart object (e.g., lighting settings for watching TV), a specific task or activity such as having a meeting or to a person's identity. Further more, the framework was expected to give insights in how to design feedback and feedforward for such a interactive lighting system.

The third designer who applied the framework to his project explored the connections between smart textiles, garments including sensors and possibly actuators, and other smart objects in the environment. Information collected by the sensors in the garment could potentially be shared with other smart objects, and the garment itself could be used to make the connections. He also proposed to add some of the functionalities of the Connector to the smart objects themselves, allowing the mobile ones to be able to connect directly. To improve the overview of existing connections between smart objects, these smart objects would show whether they are connected to one, or more other smart objects. Another idea that arose was to have a personal, possibly wearable Connector that would make and show person-specific connections.

The designer felt that he was more triggered by the examples in the demonstration than the framework itself. He did however feel that the framework asked for a way of thinking that helped to think about the issues that arise when designing interoperable products.

Two designers (i.e., the designer working on the parking management system and the one working on the interactive lighting project) indicated that they will further use the framework for their projects.

6.4.3.3 General remarks

Some more general remarks were made that are worth mentioning. The designers noticed that the interaction and form language across the smart objects and the connector are not the same. But they acknowledged the difficulty of this issue when you are not redesigning all smart objects or designing them as one system. They also pointed out that there was no clear (visual) overview of the connected devices when they are not used. Also, the source devices show no indication on whether they are connected. The suggestion was made (by two designers) to possibly use the Connector to inquire all connections to and from a particular smart object.

A remark was made that when a source and sink relation is not so clear (e.g., phone, radio) the direction (order of scanning) of the connecting action is not as appropriate as for connections with a clear source—sink relationship, and it thus seems to loose its meaning.

The concept of transitivity lead to some confusion with one designer. In this case the phone was connected to the radio, and the radio to the lamp. A transitive connection now exists between the phone and the lamp. When there is also a (redundant) connection between the phone and lamp, and this connection is removed, the transitive connection still exists. This lead to confusion as the connection between phone and lamp persists, while it was disconnected moments ago.

A surprising observation was that three designers thought that the appearance of the two rings of lights on top of the Connector resembled a crosshair, signifying the selection of a target. One of the light rings flashing red when selecting a sink as the first object, and both light rings lighting up red to indicate that a connection is not possible, reminded the designers of a red cross. Although these interpretations are very understandable, the features that cause them were never explicitly designed to do so.

6.5 Discussion

Generally, the designers responded positively to the framework. The framework was considered somewhat abstract at first, nevertheless, the designers we able to use the framework to analyse the demonstrator set-up and understand the design decisions that led to it. During this part of the evaluation, it could be easily observed that by looking at the demonstration with the framework in mind, the designers rather naturally understood the difficulties at hand and were triggered to think of solutions. The way the designers understood and (successfully) applied the framework in the analysis and the generation of new ideas, gives confidence that the framework is useful for designers and a valuable contribution to the field.

The designers were focussed on recognising the different types of feedback and feedforward, rather then assessing whether the feedback was designed adequately. This may suggest that the framework lends itself better for this purpose. A positive side to this observation is that the framework gives handles and guides to the designers to look at the design challenge from a certain perspective, still leaving room for their own creative solution to solve it.

The results show that the designers understood the vocabulary introduced by the framework, and used this vocabulary in their drawings and diagrams (see figures 6.10 and 6.11 for examples). In particular, *identification*, *source*, *sink*, *inherent*, *functional* and *augmented* (for feedback and feedforward) were often used. Interestingly, the graphical notation that was introduced (i.e., in figures 6.5 through 6.7) did, apart from the arrows, not show up consistently in the designers' drawings. Only one of the designers used a similar notation (i.e. circles for source and sink annotated with the smart objects' names, and arrows to signify connections, feedback and feedforward), the other designers drew the smart objects themselves. This suggests that the framework does not successfully introduce a notation like for instance the Frogger framework, it was based on. This is supported by the remark made by three designers that they would prefer a one-page graphical overview of the framework, to use during analysis and as a generative tool during the design process.

The mentioning of the semantic connections concepts, different types of feedback and feedforward, reached some kind of saturation in the results (repeating results for the different participants). This should be considered a good sign, as it makes the likelihood of more participants leading to very different results, less high.

For the generative part of the evaluation, the result show a wide variety in application and insights gained. Where three of the designers did not go beyond the point of finding a possible useful application for the framework and gaining some basic insights, others were able to apply it in more detail and come to very valuable insights they did not think of before. Two of the participants indicated that they would try to further use the framework in their own (research) project to come to new insights. This is an indication of a real added value of the framework.

6.6 Concluding remarks

This chapter presented a design framework to analyse and guide the design of interoperable smart objects. The framework translates the concepts of our *Semantic Connections theory* into a set of practical handles that can be used to design for interoperability. The framework aims to:

- guide designers when designing *smart objects*
- help designers to design *interfaces* to semantic connections
- help designers and developers to design for interoperability, creating a natural and consistent *user experience* for ubiquitous computing environments

The designs that are described in chapters 3 and 4 were placed in the framework to show how the framework can be applied. Finally, the framework was evaluated by a group of six designers that used the framework to analyse a demonstration. After using it for analysis, they applied it generatively to either redesign a part of the demonstration or applied it to their own project. Their application showed a good understanding of the framework, and, in some of the projects led to surprising results. The framework showed to be applicable to the various projects the designers were currently working on. It enabled the designers to approach their chosen design task from a connection centred point of view, and provided useful insights in what kind of issues arise when designing for interoperability.

Chapter 7

General Conclusions & Future Directions

This chapter summarizes the results from the design iterations, the lessons learned and the implications that our findings have on the design of interoperable smart objects. Furthermore, an outlook is given to the future, what challenges remain, and how we envision our theory and therewith increased interoperability to change the way people interact with the smart products around them.

7.1 General Discussion & Reflection

This thesis described two design explorations, the development of a theory for semantic connections and a framework for the design of interoperable smart objects. The focus of the design efforts have been on the concept of semantic connections, the interfaces to semantic connections and the behaviour of the smart objects that are connected by the semantic connections.

Chapter 3 described the first design exploration. During this exploration we explored the notion of semantic connections, what they could be like, and how they may be implemented practically. The first demonstrator that was developed served this purpose well, and taught us important lessons:

- a semantic connection exists between two real world entities in a smart environment and refers to a `connectedTo` relationship in the digital domain, which describes this relation.
- semantic connections can be used to enable users to allow for interaction on a semantic level, not having to deal with the low level details of networking technologies.

Building on our demonstrator and the notion of semantic connections, two Industrial Design master students then developed two new interfaces, under our supervision. The interfaces were used to investigate the influence of the differences in the designs on the *mental models* users may develop when interacting with a small set of smart objects, focussing on the connections between them. These user studies led to the following insights:

- participants of the experiment conducted by Kwak et al. (2011) were found to project a hierarchical way of thinking on both interface designs (Interaction Tile and Interaction Tabs), identifying a clear source and one or multiple sinks.
- Peeters et al. (2012) found that most of the participants reported to understand a centralised network structure when using the Interaction Tile, where the Interaction Tile was considered the central, mediating entity.
- for the Nodes design, participants reported a peer to peer network structure with a clear indication of direction.

Surprisingly, the two user experiments (Nodes vs. Interaction Tile) and (Interaction Tile vs. Interaction Tabs) resulted in different trends in the mental models for the Interaction Tile. While Kwak reported mainly peer-to-peer and daisy-chained models for the Interaction Tile, Peeters found that the majority of the mental models showed a centralised structure, with the Interaction Tile as a central entity. As discussed previously, this difference may be attributed to the different age groups for the experiments or it may have been caused by the differences in the methods and procedure (e.g. Peeters using a combination of non-functional prototypes and video prototypes).

Interesting insights were also gained when comparing the physically observable states of the interface designs with the digital connections they represent. Generally it can be concluded that:

- if physical state = digital state, care should be taken that the mapping between physical and digital is synchronised, not only physical actions should be reflected in the digital domain, but changes (e.g. errors, manipulations on semantic connections performed with other interfaces, automatically inferred transitive connections) in the digital domain should also clearly be reflected in the physical representation.
- a multitude of different network structures (e.g. centralised, decentralised, hierarchical) on mental models seem to be compatible with the digital `connectedTo` relationships that exist. Care should be taken that this compatibility remains when configurations become more complex.
- direction is a key property of a connection and is inferred by users, even when it is not explicitly used in the interaction or present in the actual network structure.

Chapter 4 introduces a second design iteration, introducing a refined definition of semantic connections. A new demonstrator was built, building on the SOFIA IOP and covering a more complex use case. This use case is build around our ideas of semantic connections. New interface solutions to interact with semantic connections are introduced. In this use case connections exist between various smart objects, but also indirectly between people and locations since the music experience is shared between friends at remote locations.

Building the demonstrator served different purposes. For the SOFIA project it served as a pilot, demonstrating the developed technology and for the project partners as a feasibility study. The implementation showed that semantic connections, together with the SOFIA IOP provide a platform and therewith the possibility to improve the interoperability among devices. For us, the pilot was an opportunity to test the idea of semantic connections and the two UI solutions with real users, in a close to real-world setting. Both showed their potential in moving the interaction with devices from a device-oriented paradigm towards a more task-oriented paradigm (combining the capabilities of multiple devices to perform a task) with increased interoperability.

The results of the user study showed a positive attitude of the participants towards semantic connections and the UI solutions. However, some remarks were made about the limited functionality of the demonstrator. Users saw more potential than what was implemented. With respect to the mental models users developed, the following observations were made:

- differences between mental models surfaced for the different groups; participants using the Connector generally perceived it to be part of the system while participants using Spotlight Navigation described the device itself to be outside of the system.
- smart objects that cause *incidental* interactions (e.g. presence sensor) appear less often in the mental models than smart objects with which users have *intentional* interactions.
- participants not only included connections in their mental models that they created themselves, they also included (causal) connections they perceived and which they were not able to manipulate.
- in their mental models, participants reported the concepts of transitive connections.
- even though there was no explicit direction in the interactions or the connections in this implementation, participants perceived some connections to have a direction, carrying information from its source to a destination.

Chapter 5 introduces a theory to describe semantic connections and the various related concepts that are necessary to enable them. This theory may be used to analyse, i.e. understand, explain and predict what happens with information and interaction events when devices are interconnected and form an ecology of smart objects.

Using this theory and the resulting interaction model, we implemented an example use case (i.e. the sleep/wakeup experience use case, as was introduced in Chapters 1 and 5) and refined the Connector prototype which was used to manage semantic connections. We showed that the theory can be used to implement interoperability between a different set of devices, extending the multimedia domain into the domain of well-being. Thus illustrating that our contribution may solve current interoperability problems, in both single- and cross-domain scenarios, in an elegant way. We carefully considered the notions of feedback and feedforward in our use case. We designed and implemented augmented feedback and feedforward to enhance perception of connectivity and the perceived causality between user action and feedback.

The use of feedback and feedforward was described in more detail in Chapter 6, where we defined a framework for design. The framework aims at supporting designers of smart objects to approach the design task from a connection centred point of view, inviting them to think of how their designs fit in a larger ecosystem of smart objects and understand what kind of issues arise when coupling action, perception and meaning.

The framework was evaluated by a group of six designers that used the framework to analyse the sleep use-case demonstration. After the analysis part, which helped them to understand the framework better, they either applied it generatively to redesign a part of the demonstration or applied it to their own project. Their application showed a good understanding of the framework, and, in some of the projects it led to surprising results. The designers' responses and the results give confidence that the framework is a valuable contribution to the field.

The work that was described in this thesis is part of a larger effort. The research was conducted in the context of a European research project called SOFIA and the work on semantic connections was done in close cooperation with other disciplines such as the discipline of Computer Science and engineering. As was described earlier, work was also done on modelling what happens in the physical world digitally, while preserving its meaning. Physical entities, user actions and their relations are modelled using Semantic Web technologies. In my opinion, this is where the real value lies in this work, in the combination of the work done in the physical world (e.g. design explorations, user studies) and the digital domain (e.g. semantic modelling, semantic reasoning and software architecture).

This work was only possible through the close cooperation between two disciplines, design and computer science. This experience turned out to be a win-win situation that not only favoured the work, but also fostered personal growth by learning about the other discipline. Personally I believe that such cooperative efforts are necessary to come to viable solutions to these newly posed design challenges.

The cooperation also allowed us to make effective prototypes, which led to insights in the interaction design work, the semantic connections theory, and the work on the underlying software models and architecture. Building these functional prototypes was necessary to understand the often complex implications of design decisions. Implementing and testing these prototypes gave valuable insights and resulted in sometimes surprising emergent functionalities and behaviours.

Context has proven to be key when evaluating our prototypes. During the design phase, design decisions may seem logical (and sometimes even trivial), but seeing the results in its context of use was necessary to make the right ones.

7.1.1 Reflecting on the design approach

When we again look at the Lifestyle Home (LH) project (as was introduced in Section 1.2) and the work done in the SOFIA project, we can reflect on the design approaches used in the different projects. As mentioned earlier, LH employed a very user centred approach, whereas the SOFIA project had a strong technological focus.

The designers who worked on LH, employed a method called "Multiple Encounters", an approach (developed within Philips design (Rameckers and Un, 2005)) that involves participants throughout the entire design process by evaluating design ideas, concepts and experience prototypes at multiple times in the design process. They also developed personas, which are archetypical user profiles, aiming to represent a specific targeted user group, in a single fictional person's description (Pruitt and Grudin, 2003). The outcomes remain conceptual, and the prototypes only function on a level sufficient to demonstrate, and let people experience the design concepts (hence the name experience prototypes). They serve this purpose well, and the designs are appealing and believable.

The design process of LH can also be characterised as top-down. The user experiences are designed as a whole, suiting the needs of the targeted persona. Having such a holistic view while designing, is necessary to come up with design solutions that together constitute a coherent user experience. The result, is a set of products developed to create one experience together. Such an approach can also be seen in industry today. Products developed by Apple[1] (Section 2.2) are a good example of a series of products that together compose a closed ecosystem that has been designed top-down, while having full control over all products that comprise the system (see also Section 2.2.1).

In contrast, the SOFIA project aimed at developing a technological framework that allows for multi-vendor devices to work together and share data and capabilities. For designers of such products this means that they are only designing a part of a larger, open ended system. Therefore, having a holistic view on the targeted user experience is difficult, if not impossible. For such design challenges a bottom-up approach is needed, enabling designers to "prepare" their designs for integration in a yet unknown situation at a later stage. For this purpose, we developed the alternative design approach, as described in Chapters 5 and 6 .

In this alternative design approach, the emphasis is on the connections that connect devices together, rather than on the devices themselves. By moving the emphasis towards the connections, designers are invited to think about the role their design may have in a larger whole. It invites designers (and developers) to think about what capabilities of, or information contained within the device would be suitable to share, and how the device would function in a larger ecosystem. Our proposed approach challenges designers to think of how the design will respond to interaction with users, interaction between their designs and other (connected) devices, and how to respond to users interacting with those connected devices. We expect that such a design approach suits the needs of designers of interoperable systems better than conventional design methods would, as we have shown in the SOFIA project.

7.2 Future Directions & Challenges

During the course of the three year PhD project, many innovations in the area of interoperability have emerged. The big technology companies have introduced innovations in content sharing among their products, sometimes leading to innovative UI solutions (as was described in Section 2.2). Third parties have developed apps to run on the most popular platforms of smart phones and tablets enabling limited forms of interoperability and remote control. These developments show that the work done in this project is relevant, and that the products and technologies that we target, are currently being developed. However, interoperability still takes place predominantly within one manufacturer's ecosystem and within one domain. Within one domain, vendor lock-in approaches have proven to be successful. Therefore, real *emerging functionalities* are hard to find, and sometimes also hard to envision.

This work attempts to look beyond what is possible within one domain, and provide designers and developers with handles to deal with the challenges this postulates. We also speculate that, once interoperability within one domain has been achieved, added value will be looked for, across domains. When crossing domains, no single manufacturer will be capable of developing and managing the full ecosystem of products and services. This need for cross vendor and cross domain interoperability may eventually lead to the transformations that we envision (and that were envisioned in the SOFIA project).

---

[1] http://www.apple.com

Although this work aims to shift the focus from designing one single product, to designing interoperating products, there are still many challenges ahead. Besides the many technological challenges, there are a few particular challenges we like to point out. While the semantic connections theory describes connections between smart objects, there are many more meaningful connections thinkable. An attempt was made to also describe connections between objects and people, and objects and places. Relationships between people are also increasingly described digitally, by the pervasiveness of social networks. This information is considered very relevant for sharing information among objects that are owned and used by different people, and using such information is an interesting challenge.

Additionally, the challenge remains to elevate user interaction to a higher, more meaningful level (instead of having users deal with low level technicalities). The challenge here is to automate the right actions, still leaving sufficient control to the users. Finally the challenge remains to use increased interoperability for a good cause. Hopefully, it will lead to less redundancy in functionalities and devices around us, by enabling new functionalities to emerge from the interconnection of smart objects. It may open up possibilities to design interfaces that are more task oriented than device oriented, replacing some of the many generic users interfaces. It may also have potential to improve quality of life. Information interoperability may be employed such that people have access to information to help them with staying fit and healthy. The challenge here is to not only provide people with data, but offering them meaningful information and action possibilities they can act on immediately. An example is our sleep use case aimed at improving well-being.

There is also a chance to decrease semantic pollution and to create more aesthetic interactions. Although aesthetics in itself was not a primary aim in this thesis, beauty may be found in the simplicity and elegance we aimed for in our work. In general, it should lead to more control for users, and more importantly, more control over that which is meaningful.

7.3 Conclusions

In Chapter 1 we asked the following research question:

How can the design of smart objects help users to construct meaningful mental models of:

1. *the smart objects and the smart environment these objects are part of*

   and:

2. *the connections (and their meanings) that exist between the smart objects?*

To conclude this thesis we reflect on whether and how this question was answered and discuss the hypotheses that were also introduced in Section 1.3.4.

In Chapter 3 and 4 we explored different designs that were aimed at enabling users to explore and manipulate connections between smart objects. These designs were employed in use cases with varying complexity and were evaluated in terms of the mental models that users developed when they interacted with the designs. The evaluations showed that, among other things (e.g. age group, medium and a user's previous experience), the physical design of the smart objects did influence the way in which users conceptualise the connections between the smart objects. We also found that the mental models allowed users to understand and sometimes predict the functional results of the connections.

We showed that smart objects can be designed in such a way as to invite for connecting smart objects together. This confirms our first hypothesis. To some extent we showed that inherent and augmented feedforward can exist between smart objects, communicating that and how they may be connected.

Confirming our second hypothesis, we showed that the qualities of the connecting actions, such as direction, order and location, can be used by users to express their intentions in terms of the information they want to exchange between the smart objects. Location (and also proximity) was employed to identify a source and sink for a connection. Users seemed to naturally associate the order of identification with direction of the connections. Additionally, actions with qualities associated with physically connecting and disconnecting were used.

We partially confirmed our third hypothesis. The results of the evaluations showed that users were able to connect the correct smart objects together to achieve their goals (or accomplish the tasks they were given) in terms of exchanging information. However, users' goals may be more specific in some thinkable situations. Our *Semantic Connections Theory* describes how semantic connections and the smart objects they connect can be described more precisely, to deal with more specific user needs.

We also showed that enabling users to actively explore and manage the connections between smart objects, helped them to create useful mental models. We showed that, to some extent, these mental models led to an understanding of the emerging functionalities, new action possibilities and information flows within the smart space. Finally, the Semantic Connections Theory and framework show how we achieved our goals, and may help designers and developers to successfully design for interoperability.

Bibliography

Aarts, E. and Marzano, S. (2003). *The New Everyday: Views on Ambient Intelligence*. 010 Publishers, Rotterdam, The Netherlands.

Allemang, D. and Hendler, J. A. (2008). *Semantic web for the working ontologist*. Morgan Kaufmann.

Archer, B. (1995). The Nature of Research. *Co-design*, 2:6–13.

Arnall, T. (2006). A graphic language for touch-based interactions. In *Proceedings of Mobile Interaction with the Real World MIRW*.

ARTEMIS JU (2008). Technical annex. Grant agreement for: SOFIA (Smart Objects For Intelligent Applications).

Ayatsuka, Y. and Rekimoto, J. (2005). tranSticks: physically manipulatable virtual connections. In *CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 251–260, Portland, Oregon, USA. ACM.

Ballagas, R., Borchers, J., Rohs, M., and Sheridan, J. G. (2006). The smart phone: A ubiquitous input device. *IEEE Pervasive Computing*, 5(1):70–77.

Ballagas, R., Ringel, M., Stone, M., and Borchers, J. (2003). iStuff: A Physical User Interface Toolkit for Ubiquitous Computing Environments. In *Proceedings of the ACM CHI 2003 Conference on Human Factors in Computing Systems*, pages 537–544, Ft. Lauderdale, Florida, USA.

Bartolini, S., Milosevic, B., D'Elia, A., Farella, E., Benini, L., and Cinotti, T. S. (2011). Reconfigurable natural interaction in smart environments: approach and prototype implementation. *Personal and Ubiquitous Computing*.

Bass, L., Faneuf, R., Little, R., Mayer, N., Pellegrino, B., Reed, S., Seacord, R., and Sheppard, S. (1992). A metamodel for the runtime architecture of an interactive system. *SIGCHI Bulletin*, 24(1):32–37.

Beardsley, P., van Baar, J., Raskar, R., and Forlines, C. (2005). Interaction using a handheld projector. *IEEE Computer Graphics and Applications*, 25(1):39–43.

Beauduin, M. and Mackay, W. (2003). *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications*, chapter Prototyping Tools & Techniques, pages 1006–1031. L. Erlbaum Associates Inc.

Bongers, B. and Veer, G. C. (2007). Towards a Multimodal Interaction Space: categorisation and applications. *Personal and Ubiquitous Computing*, 11(8):609–619.

Bratman, M. E. (1987). *Intention, Plans, and Practical Reason*. Harvard University Press, Cambridge, MA.

Brooke, J. (1996). Sus-a quick and dirty usability scale. *Usability evaluation in industry*, pages 189–194.

Bürdek, B. E. (2005). *Design: History, Theory and Practice of Product Design*. Birkhäuser, Basel, Switzerland.

Buxton, W. (1983). Lexical and pragmatic considerations of input structures. *ACM SIGGRAPH Computer Graphics*, 17(1):31–37.

Card, S. K., Mackinlay, J. D., and Robertson, G. G. (1991). A morphological analysis of the design space of input devices. *ACM Transactions on Information Systems (TOIS)*, 9(2):99.

Chandler, D. (2002). *Semiotics: The Basics*. Routledge, London.

Chandrasekaran, B., Josephson, J., and Benjamins, V. (1999). What are ontologies, and why do we need them? *Intelligent Systems and their Applications, IEEE*, 14(1):20–26.

Chen, H., Perich, F., Finin, T., and Joshi, A. (2004). SOUPA: standard ontology for ubiquitous and pervasive applications. In *Mobile and Ubiquitous Systems: Networking and Services, MOBIQUITOUS 2004*, pages 258–267.

Coutrix, C. and Nigay, L. (2006). Mixed reality: a model of mixed interaction. In *Proceedings of the working conference on Advanced Visual Interfaces*, pages 43–50.

Crilly, N., Maier, A., and Clarkson, P. J. (2008). Representing artefacts as media: Modelling the relationship between designer intent and consumer experience. *International Journal of Design; Vol 2, No 3 (2008)*.

de Ruiter, M. M. (1988). *Advances in Computer Graphics III*. Springer-Verlag.

Dix, A., Finlay, J., Abowd, G. D., and Beale, R. (2004). *Human-Computer Interaction*. Pearson Education Limited, 3rd editio edition.

Dix, A., Ghazali, M., Gill, S., Hare, J., and Ramduny-Ellis, D. (2008). Physigrams: modelling devices for natural interaction. *Formal Aspects of Computing*, 21(6):613–641.

Djajadiningrat, J. P. (1998). *Cubby: What You See Is Where You Act*. Unpublished doctoral dissertation, TU Delft, Delft, The Netherlands.

Dubois, E. and Gray, P. (2008). A design-oriented information-flow refinement of the ASUR interaction model. *Engineering Interactive Systems*, 4940/2008:465–482.

Eco, U., Young, J., and Walton, P. (1977). *A Theory of Semiotics*. Critical Social Studies. The Mackmillan Press Ltd., London, England.

Fitzmaurice, G. W. (1993). Situated information spaces and spatially aware palmtop computers. *Commun. ACM*, 36(7):39—-49.

Foley, J. D., van Dam, A., Feiner, S. K., and Hughes, J. F. (1996). *Computer graphics: principles and practice*. Addison-Wesley Publishing Company, Inc., 2nd editio edition.

Foley, J. D., Wallace, V. L., and Chan, P. (1984). The human factors of computer graphics interaction techniques. *IEEE Computer Graphics and Applications*, 4(11):13–48.

Frens, J. (2006). A rich user interface for a digital camera. *Personal and Ubiquitous Computing*, 10:177–180. 10.1007/s00779-005-0013-z.

Frens, J. and Overbeeke, C. (2009). Setting the stage for the design of highly interactive systems. In *Proceedings of International Association of Societies of Design Research 2009 - IASDR'09*, pages 1 – 10, Seoul, Korea. Korean Society of Design Science.

Geere, D. (2011). The window into the world of the internet of everything. `http://www.wired.co.uk/news/archive/2011-09/14/qualcomm-internet-of-everything`.

Gros, J. (1983). Grundlagen einer theorie der productsprache. vol 1. HfG Offenbach, Offenbach/Main.

Guldberg, J. (2010). Singular or multiple meanings? a critique of the index/anzeichen approach to design semiotics/semantics. In Chen, L.-L., Dajadiningrat, T., Feijs, L., Kyffin, S., Steffen, D., and Young, B., editors, *Proceedings of Design and Semantics of Form and Movement (DeSForM) 2010*, pages 48–56, Lucerne, Switzerland. Koninklijke Philips Electronics N.V.

Hebeler, J., Fisher, M., Blace, R., and Perez-Lopez, A. (2009). *Semantic web programming*. Wiley Publishing.

Heflin, J. (2004). OWL Web Ontology Language Use Cases and Requirements. `http://www.w3.org/TR/webont-req/`.

Heider, T. and Kirste, T. (2002). Supporting goal-based interaction with dynamic intelligent environments. In *Proceedings of the 15th European Conference on Artificial Intelligence*, Lyon, France.

Hepp, M. (2007). Ontologies: State of the Art, Business Potential, and Grand Challenges. In Hepp, M., De Leenheer, P., De Moor, A., and Sure, Y., editors, *Ontology Management: Semantic Web, Semantic Web Services, and Business Applications*, pages 3–22. Springer.

Hinckley, K., Ramos, G., Guimbretiere, F., Baudisch, P., and Smith, M. (2004). Stitching: pen gestures that span multiple displays. In *Proceedings of the working conference on Advanced visual interfaces*, pages 23–31. ACM.

Hinds, P. (1998). User control and its many facets: a study of perceived control in human-computer interaction. Technical report, Hewlett Packard Laboratories.

Holmquist, L. E., Redström, J., and Ljungstrand, P. (1999). Token-based acces to digital information. In *HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, pages 234–245, London, UK. Springer-Verlag.

Honkola, J., Laine, H., Brown, R., and Tyrkko, O. (2010). Smart-M3 Information Sharing Platform. In *Computers and Communications (ISCC), 2010 IEEE Symposium on*, pages 1041–1046, Riccione, Italy.

Hummels, C. and Frens, J. (2008). Designing for the unknown: A design process for the future generation of highly interactive systems and products. In Clarke, A., Evatt, M., Hogharth, P., Lloveras, J., and Pons, L., editors, *New perspectives in Design Education Volume 1. Proceedings of the 10th International Conference on Engineering and Product Design Education, Barcelona, Spain, 4-5 September*.

Hummels, C. and Frens, J. (2011). *Designing Disruptive Systems, Products and Services: RTD Process*. InTech.

Ishii, H. and Ullmer, B. (1997). Tangible bits: towards seamless interfaces between people, bits and atoms. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 234–241, Atlanta, Georgia, United States. ACM.

Jepsen, T. C. (2009). Just what is an ontology, anyway? *IT Professional*, 11(5):22–27.

Jeronimo, M. and Weast, J. (2009). *UPnP Design by Example*. Intel Press.

Johansson, S. (2009). Sniff: designing characterful interaction in a tangible toy. In *Proceedings of the 8th International Conference on Interaction Design and Children*, pages 186–189.

Koninklijke Philips Electronics N.V. (2006). Lifestyle home lifestyle home - shape your own connected experiences. online.

Krippendorff, K. (2006). *the semantic turn: a new foundation for design*. CRC Press, Boca Raton.

Krippendorff, K. and Butter, R. (1989). *Design Issues*, 5(2).

Kuniavsky, M. (2010). *Smart Things: Ubiquitous Computing User Experience Design*. Morgan Kaufmann.

Kwak, M., Niezen, G., van der Vlist, B., Hu, J., and Feijs, L. (2011). Tangible interfaces to digital connections, centralized versus decentralized. In Pan, Z., Cheok, A., Müller, W., and Yang, X., editors, *Transactions on Edutainment V*, volume 6530 of *Lecture Notes in Computer Science*, pages 132–146. Springer Berlin / Heidelberg.

Kyffin, S., Feijs, L., and Djajadiningrat, T. (2005). Exploring expression of form, action and interaction. In Sloane, A., editor, *Home-Oriented Informatics and Telematics*, volume 178 of *IFIP International Federation for Information Processing*, pages 171–192. Springer Boston.

Lakoff, G. and Johnson, M. (1980). *Metaphors We Live By*. University Of Chicago Press.

Lee, H., Jeong, H., Lee, J., Yeom, K., Shin, H., and Park, J. (2008). Select-and-point: a novel interface for multi-device connection and control based on simple hand gestures. In *CHI '08: extended abstracts on Human factors in computing systems*, pages 3357–3362, Florence, Italy. ACM.

Mackay, W. E. and Fayard, A.-L. (1997). Hci, natural science and design: a framework for triangulation across disciplines. In *Proceedings of the 2nd conference on Designing interactive systems: processes, practices, methods, and techniques*, DIS '97, pages 223–234, New York, NY, USA. ACM.

Mackinlay, J., Card, S., and Robertson, G. (1990). A Semantic Analysis of the Design Space of Input Devices. *Human-Computer Interaction*, 5(2):145–190.

Martinussen, E. S. and Arnall, T. (2009). Designing with rfid. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, pages 343–350.

Martinussen, E. S., Knutsen, J., and Arnall, T. (2007). Bowl: token-based media for children. In *Proceedings of the 2007 conference on Designing for User eXperiences*, page 17.

Merabti, M., Fergus, P., Abuelma'atti, O., Yu, H., and Judice, C. (2008). Managing distributed networked appliances in home networks. *Proceedings of the IEEE*, 96(1):166–185.

Merrill, D., Kalanithi, J., and Maes, P. (2007). Siftables: towards sensor network user interfaces. In *Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 75–78, Baton Rouge, Louisiana. ACM.

Miller, R. B. (1968). Response time in man-computer conversational transactions. In *Proceedings of the Fall Joint Computer Conference*, page 267. ACM Press.

Molyneaux, D., Gellersen, H., Kortuem, G., and Schiele, B. (2007). Cooperative augmentation of smart objects with projector-camera systems. In *Proceedings of the 9th international conference on Ubiquitous computing*, pages 501–518. Springer-Verlag.

Newman, M. W., Sedivy, J. Z., Neuwirth, C. M., Edwards, W. K., Hong, J. I., Izadi, S., Marcelo, K., and Smith, T. F. (2002). Designing for serendipity: supporting end-user configuration of ubiquitous computing environments. In *Proceedings of the 4th conference on Designing interactive systems: processes, practices, methods, and techniques*, DIS '02, pages 147–156, New York, NY, USA. ACM.

Ngo, H. Q., Shehzad, A., Liaquat, S., Riaz, M., and Lee, S. (2004). Developing Context-Aware Ubiquitous Computing Systems with a Unified Middleware Framework. In *Embedded and Ubiquitous Computing*, pages 239–247.

Nielsen, J. (1986). A virtual protocol model for computer-human interaction. *International Journal of Man-Machine Studies*, 24(3):301–312.

Niezen, G. (2012). *Ontologies for interaction: Enabling serendipitous interoperability in smart environments*. forthcoming, Eindhoven University of Technology, Department of Industrial Design, Eindhoven, the Netherlands.

Niezen, G., van der Vlist, B., Hu, J., and Feijs, L. (2010). From events to goals: supporting semantic interaction in smart environments. In *Proceedings of the IEEE Symposium on Computers and Communications (ISCC)*, pages 1029–1034, Riccione, Italy.

Niezen, G., van der Vlist, B., Hu, J., and Feijs, L. (2011). Using semantic transformers to enable interoperability between media devices in a ubiquitous computing environment. In *International Workshop on Self-managing Solutions for Smart Environments (S3E 2011) at the 6th International Conference on Grid and Pervasive Computing (GPC 2011)*, Oulu, Finland.

Nordby, K. (2010a). Conceptual designing and technology: Short-range RFID as design material. *International Journal of Design*, 4(1):29–44.

Nordby, K. (2010b). Multi-field relations in designing for short-range RFID. *Personal and Ubiquitous Computing*, 15(2):175–185.

Norman, D. A. (1998). *The Design of Everyday Things*. MIT Press Ltd, illustrated edition edition.

Oliver, I. and Honkola, J. (2008). Personal semantic web through a space based computing environment. In *Middleware for the Semantic Web, Second IEEE International Conference on Semantic Computing*, Santa Clara, CA, USA.

Oulasvirta, A., Kurvinen, E., and Kankainen, T. (2003). Understanding contexts by being there: case studies in bodystorming. *Personal and Ubiquitous Computing*, 7:125–134.

Peeters, J., van der Vlist, B., Niezen, G., Hu, J., and Feijs, L. (2012). A study on a tangible interaction approach to managing wireless connections in a smart home environment. In *7th International Workshop on the Design & Semantics of Form & Movement (DeSForM) 2012*, 18-12 April 2012, Wellington, New Zealand. Koninklijke Philips Electronics N.V.

Poole, E. S., Chetty, M., Grinter, R. E., and Edwards, W. K. (2008). More than meets the eye: transforming the user experience of home network management. In *Proceedings of the 7th ACM conference on Designing interactive systems*, DIS '08, pages 455–464, New York, NY, USA. ACM.

Pruitt, J. and Grudin, J. (2003). Personas: practice and theory. In *Proceedings of the 2003 conference on Designing for user experiences*, DUX '03, pages 1–15, New York, NY, USA. ACM.

Rameckers, L. and Un, S. (2005). People insights at the fuzzy front end of innovation. Philips Design.

Ranganathan, A., Al-Muhtadi, J., and Campbell, R. (2004). Reasoning about uncertain contexts in pervasive computing environments. *Pervasive Computing, IEEE*, 3(2):62 –70.

Rapp, S. (2010). Spotlight Navigation: a pioneering user interface for mobile projection. In *Proc. Ubiprojection*, Helsinki.

Rekimoto, J. (1997). Pick-and-drop: a direct manipulation technique for multiple computer environments. In *Proceedings of the 10th annual ACM symposium on User interface software and technology*, pages 31–39, Banff, Alberta, Canada. ACM.

Rekimoto, J., Ayatsuka, Y., Kohno, M., and Oba, H. (2003). Proximal interactions: A direct manipulation technique for wireless networking. In *Proceedings of Human-Computer Interaction; INTERACT'03*, pages 511—518, Amsterdam, the Netherlands. IOS Press.

Rock, I. and Palmer, S. (1990). *The Legacy of Gestalt Psychology*. University of California, Berkeley.

Rukzio, E. and Holleis, P. (2010). Projector Phone Interactions: Design Space and Survey. In *Workshop on Coupled Display Visual Interfaces at AVI 2010*, Rome, Italy.

Rukzio, E., Leichtenstern, K., Callaghan, V., Holleis, P., Schmidt, A., and Chin, J. (2006). An experimental comparison of physical mobile interaction techniques: Touching, pointing and scanning. In *8th International Conference on Ubiquitous Computing (UbiComp'06)*, pages 87–104. Springer.

Satyanarayanan, M. (2001). Pervasive computing: vision and challenges. *Personal Communications, IEEE*, 8(4):10 –17.

Scherp, A., Staathoff, C., Franz, T., and Staab, S. (2011). Designing core ontologies. *Applied Otology*, 6:177–221.

Shneiderman, B. (1997). Direct manipulation for comprehensible, predictable and controllable user interfaces. In *Proceedings of the 2nd International Conference on Intelligent User Interfaces*, pages 33–39, Orlando, Florida, United States.

Siio, I., Masui, T., and Fukuchi, K. (1999). Real-world interaction using the FieldMouse. In *Proceedings of the 12th annual ACM symposium on User interface software and technology - UIST '99*, volume 1, pages 113–119. ACM Press.

Steffen, D. (2000). *Design als Produktsprache. Der 'Offenbacher Ansatz' in Theorie und Praxis*. Verlag Form, Frankfurt am Main.

Thimbleby, H. (2007). *Press on: Principles of interaction programming*. MIT Press.

Torge, S., Rapp, S., and Kompe, R. (2002). The planning component of an intelligent human-machine interface in changing environments. In *Multi-Modal Dialogue in Mobile Environments (IDS-02), ISCA Tutorial and Research Workshop (ITRW)*.

Ullmer, B. and Ishii, H. (2000). Emerging frameworks for tangible user interfaces. *IBM Systems Journal*, 39(3):915.

Ullmer, B., Ishii, H., and Glas, D. (1998). mediaBlocks: physical containers, transports, and controls for online media. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 379–386, New York, NY, USA. ACM.

Ullmer, B., Ishii, H., and Jacob, R. J. K. (2005). Token+constraint systems for tangible interaction with digital information. *ACM Transactions on Computer-Human Interaction*, 12(1):81–118.

van der Heijden, H. and Sorensen, L. S. (2003). Measuring attitudes towards mobile information services: An emperical validation of the hed/ut scale. In *European Conference on Information Systems (ECIS)*, Napels, Italy.

van der Veer, G. C. and del Carmen Puerta Melguizo, M. (2003). *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications*, chapter Mental Models, pages 52–80. L. Erlbaum Associates Inc., Mahwah, New Jersey, USA.

van der Vlist, B., Niezen, G., Hu, J., and Feijs, L. (2010a). Design Semantics of Connections in a Smart Home Environment. In Chen L.L., Djajadiningrat T., Feijs L.M.G., Kyffin S., Steffen D., and Young B., editors, *Proceedings of Design and Semantics of Form and Movement (DeSForM)*, pages 48–56, Lucerne, Switzerland. Koninklijke Philips Electronics N.V.

van der Vlist, B., Niezen, G., Hu, J., and Feijs, L. (2010b). Semantic Connections: Exploring and Manipulating Connections in Smart Spaces. In *Proceedings of the IEEE Symposium on Computers and Communications (ISCC)*, pages 1–4, Riccione, Italy.

van der Vlist, B., Niezen, G., Hu, J., and Feijs, L. (2011). Interaction Primitives: Describing Interaction Capabilities of Smart Objects in Ubiquitous Computing Environments. In *IEEE AFRICON 2011*, pages 1–6, Livingstone, Zambia.

van Gent, S., Megens, C., Peeters, M., Hummels, C., Lu, Y., and Brombacher, A. (2011). Experiential design landscapes as a design tool for market research of disruptive intelligent systems. In *Proceedings of the 1st Cambridge Academic Design Management Conference, 7-8 september 2011, Cambridge*. Cambridge: University of Cambridge.

van Kranenburg, R. (2008). The internet of things: A critique of ambient technology and the all-seeing network of rfid. *Institute of Network Cultures*.

Villalonga, C., Strohbach, M., Snoeck, N., Sutterer, M., Belaunde, M., Kovacs, E., Zhdanova, A., Goix, L., and Droegehorn, O. (2009). Mobile ontology: Towards a standardized semantic model for the mobile domain. In *Service-Oriented Computing-ICSOC 2007 Workshops*, pages 248–257. Springer.

W3C (2004). OWL web ontology language use cases and requirements. `http://www.w3.org/TR/webont-req/#usecase-ubiquitous`.

Want, R., Fishkin, K. P., Gujar, A., and Harrison, B. L. (1999). Bridging physical and virtual worlds with electronic tags. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 370–377, New York, NY, USA. ACM.

Weiser, M. (1991). The computer for the 21st century. *Scientific American*.

Wensveen, S. A. G. (2005). *A tangibility approach to affective interaction*. Unpublished doctoral dissertation, TU Delft, Delft, The Netherlands.

Wensveen, S. A. G., Djajadiningrat, J. P., and Overbeeke, C. J. (2004). Interaction frogger: a design framework to couple action and function through feedback and feedforward. In *Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques*, pages 177–184.

Woo, J. and Lim, Y. (2009). Contact-and-connect: designing new pairing interface for short distance wireless devices. In *CHI '09: Proceedings of the 27th international conference* *extended abstracts on Human factors in computing systems*, pages 3655–3660, New York, NY, USA. ACM.

Zingale, S. (2010). Wayfinding using colour: A semiotic research hypothesis. In Chen, L.-L., Djajadiningrat, T., Feijs, L., Kyffin, S., Steffen, D., and Young, B., editors, *Proceedings of Design and Semantics of Form and Movement (DeSForM) 2010)*, pages 22–32, Lucerne, Switzerland. Koninklijke Philips Electronics N.V.

Appendix A

Interaction Tile

The interaction tile consists of the following components:

- ATMEGA168 micro controller with Arduino[1] boot-loader ;
- Arduino mini-USB serial adapter[2]
- ACS 13.56MHz RFID reader[3] with RF SOLUTIONS external antenna[4];
- multi-colour LED's;
- accelerometer;
- vibration motor;
- piezoelectric speaker;
- magnetic switches.

Software running on the Arduino micro controller communicates to a PC using the serial-over-USB protocol. The wiring diagram of the Interaction Tile is available in Figure A.1. The RFID reader is connected to the PC through USB separately, and uses the PC/SC communication protocol for card readers.

Software written in Python runs on the PC and handles the events generated by the Interaction Tile. It inserts events and data into a triple store (SIB) and queries it when information is needed. A reasoner (Pellet[5]) is used to reason about the inserted low-level events in order to infer higher-level results. When a user establishes a connection, two `NFCEnterEvent` events (generated by the RFID reader) by two different devices not currently connected, will result in a new `connectedTo` relationship between the two devices. Because `connectedTo` is a symmetric relationship, the reasoner will automatically infer that a connection from device A to device B means that device B is also connected to device A. Since `connectedTo` is also an irreflexive property, it is not possible for a device to be connected to itself. A `generatedBy` relationship is also created between the event and the smart device that generated it, along with a timestamp and other event metadata.

---

[1] http://www.arduino.cc

[2] http://arduino.cc/en/Main/MiniUSB

[3] http://www.acs.com.hk/index.php?pid=product&id=ACR122U

[4] http://nl.farnell.com/jsp/search/productdetail.jsp?sku=1304031&CMP=i-bf9f-00001000

[5] http://clarkparsia.com/pellet/

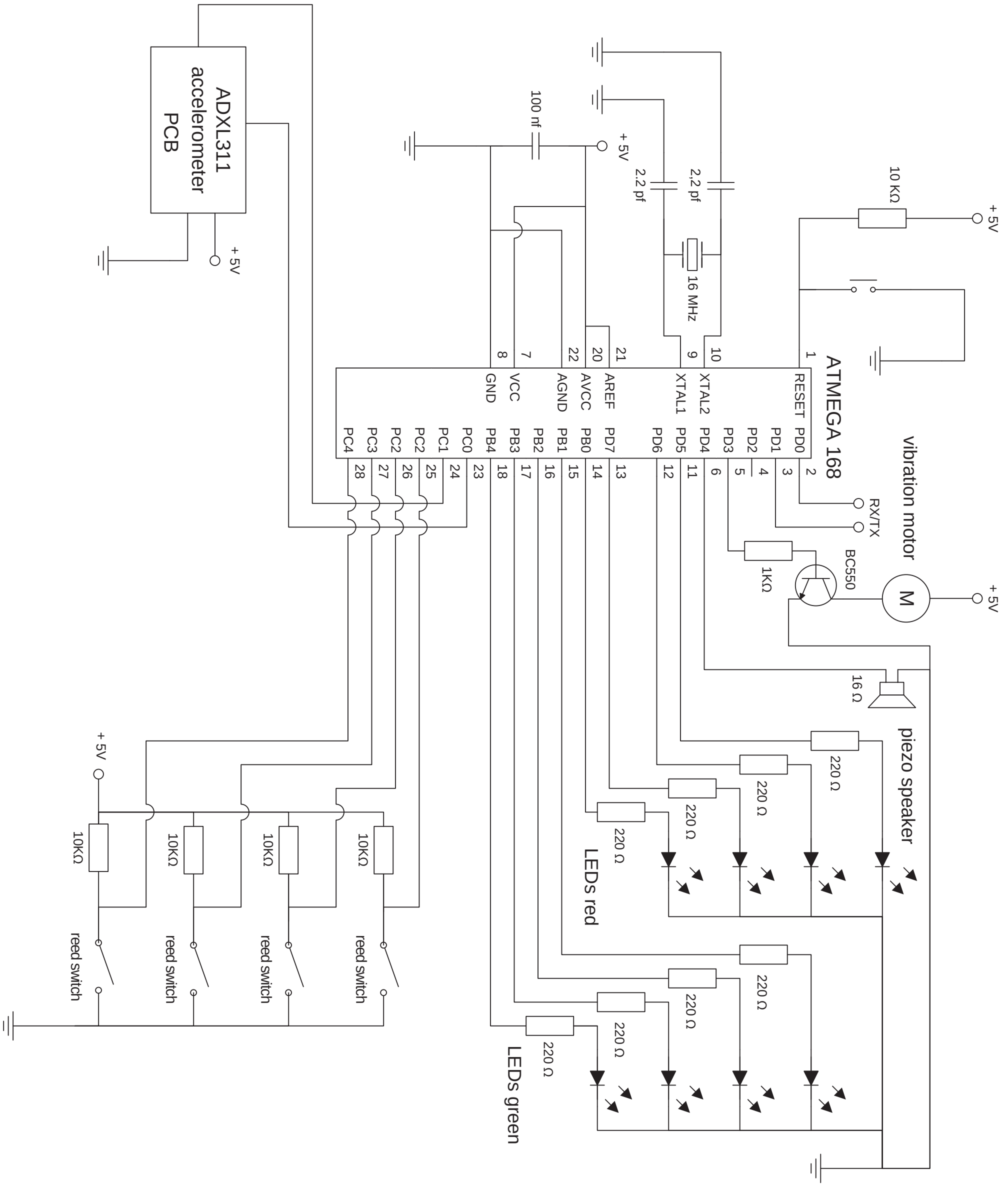


Figure A.1: Circuit drawing of the Interaction Tile hardware

Code examples of inserting triples into and querying triples from the SIB can be found in Listings A.1 and A.2 (code by Gerrit Niezen):

```
def addEvent(eventType, tagID, position):
    """Adds new event of *eventType* with metadata (**inXSDDateTime**,
        **hasRFIDTag** and **hasPosition**) to smart space"""
    t = RDFTransactionList()
    u1 = ns + "event" + str(uuid.uuid4())
    t.setType(u1, ns + eventType)
    #t.add_literal(u1, ns + "generatedBy", ns + self.deviceID)
    dt = datetime.now()
    xsddt = '"'+dt.strftime('\%Y-\%m-\%dT\%H:\%M:\%S\%z') + '"^^<http
        ://www.w3.org/2001/XMLSchema#dateTime>'
    t.add_literal(u1, ns + "inXSDDateTime", xsddt)
    t.add_literal(u1, ns + "hasRFIDTag", "\"" + tagID + "\"^^<http://
        www.w3.org/2001/XMLSchema#string>");
    t.add_literal(u1, ns + "hasPosition", "\"" + str(position) + "\"^^<
        http://www.w3.org/2001/XMLSchema#integer>")
    print eventType, ": Object ", tagID, " next to position ", position
    ts.insert(t.get())
    return u1
```

Listing A.1: Example Python code of inserting information

```
def showConnections():
    """
    Determines which devices are currently next to the interaction tile
         by querying the smart space,
    and an internal array storing the positions relative to the tile.

    Sends a response to the interaction tile (using ``sendToTile()``
    on whether a connection exists or whether a connection is possible
         .)
    """
    global rfid

    #Count number of devices
    devices = 4 - rfid.count("")
    print "Number of devices: ", devices

    for item in rfid:
        if item != "":

            if devices < 2:
                #If there is only one (or zero) devices, show that no
                     connections are possible
                sendToTile('n',rfid.index(item))

            #If the device isn't connected to anything, it won't be
                found and set in the search below,
            #so at the end it should be set to "Possible"
            alreadySet = False

            device = getNameFromTag(item)

            qs = node.CreateQueryTransaction(smartSpace)
            result = qs.rdf_query([((device, ns + "connectedTo", None )
                ,"uri")])
            for connDevice in result:
                print device.split("#")[1], " is connected to ",
                     connDevice[0][2].split("#")[1]

                #Is connDevice next to tile?
                for connItem in rfid:
                    connName = getNameFromTag(connItem)

                    if(connName == connDevice[0][2]):
                        #Yes, it's next to the tile, show as connected
                        sendToTile('c',rfid.index(item))
                        sendToTile('c',rfid.index(connItem))
                        alreadySet = True

            if (alreadySet == False) and (devices > 1):
                sendToTile('p',rfid.index(item))

            node.CloseQueryTransaction(qs)
```

Listing A.2: Example Python code of querying information

Code example of the Interaction Tile software is available in Listing A.3:

```
/** if more then one card exits at the same time, listen to the
 * accelerometer to detect an interaction event and send the
 * result of the interaction event over the serial line.
 * if only one card exits send the position of the exiting card.
 */
 void cardEvent()
 {
   boolean interactionEvent = false; //do we have an interaction event?

   int exitPos = -1;
   int prevExit;
   int reedEvent[4];

   for (int i = 0; i < 4; i++) {reedEvent[i] = digitalRead(reed[i]);}
       //store new reedvalues in an array
   for (int i = 0; i < 4; i++) {
     if (reedEvent[i] != reedState[i]) {  //compare arrays
       if (reedEvent[i] == LOW) {  //if card exits
         count ++;                           // add 1 to counter
         time = millis();                    // store current time
         if ((count > 1) && ((time - prevTime) < interval)) {
         // if more then one cards left and time difference
         // between two subsequent card exits is smaller
         // then the predefined interval
         // listen to interaction event
           for (int i = 0; i < 4; i++) {digitalWrite(ledR[i], HIGH);}
           while(!end) { // wait for an interaction to finish
             checkAccel();
             delay(10);
           }
           end = false;
           interactionEvent = true;
           count = 0;            // reset counter
           exitPos = -1;
         } else if ((time - prevTime) < interval) {
           time = prevTime; //equalize time and prevTime
         } else {
           exitPos = i;
         }
         prevTime = time; //prevTime should store the "current" time
       }

       if (reedEvent[i] == HIGH) { //if a card enters the field, write
           the position plus 10
         timeEnter = millis();
         count1 ++;
         if ((count1 > 1) && ((timeEnter - prevTimeEnter) < interval))
            {
           count1 = 0;
           //timeEnter = prevTimeEnter; //prevTime should store the "
               current" time
         } else {
           prevTimeEnter = timeEnter; //prevTime should store the "
               current" time
           Serial.write(i+10);
           Serial.write('E');  // followed by serial event marker
```

```
        }
      }
    }
  }
  if (exitPos >= 0) {
    Serial.write(exitPos);
    Serial.write('E'); // "E" is the serial event marker
  }
  // if we have an interaction event, send the result over the serial line
  if (interactionEvent) {
    // not sure about this
    //Serial.write(prevExit+20);
    //Serial.write('E');
    if (toggle == true) {
      Serial.write('C');  // connected!
      Serial.write('E');  // "E" is the serial event marker
    } else {
      Serial.write('D');  // disconnected!
      Serial.write('E');  // "E" is the serial event marker
    }
  }
}
```

Listing A.3: Code example of the Interaction Tile software

Appendix B

Connector

The Connector prototype is made out of four separate 3D printed pieces (as is shown in Figure B.1). The lower part and the top part of the Connector can be moved inward and outward serving as a two-way spring-loaded switch. The prototype packages all the necessary components into one integrated device which is wirelessly connected to a computer using a Bluetooth connection.

Connector contains the following main components:

- Arduino Stamp 02
- Innovations ID-12 125kHz RFID reader
- SparkFun BluetoothMateGold
- 8 3mm bi-colour LEDs
- Two switches
- 3.7V LiPo battery (850 mAh)
- 5V DC to DC step-up VPack PCB

A wiring diagram of the hardware is available in Figure B.3.

The software running on the Arduino micro-controller communicates via serial over Bluetooth to a PC that runs the KP software (written in Python). It also uses the SoftwareSerial library[1] for serial communication with the ID-12 RFID reader, using normal digital I/O pins. The code for communicating with the ID-12 RFID reader was based on work by Martin Rädlinger[2] and[3].

The software on the Connector reads the RFID tags and sends the tags-IDs to the Connector KP. The KP then queries the SIB to lookup the tag-ID and find the matching smart object. When two smart objects are selected, the KP determines the connection possibilities and communicates the result back to the Connector, which handles the appropriate feedback. When a user connects or disconnects two devices, this event is communicated to the Connector KP, which in turn inserts or removes a `connectedTo` relationship between the two smart objects.

---

[1] http://arduino.cc/en/Reference/SoftwareSerial

[2] http://blog.formatlos.de/2008/12/08/arduino-id-12/

[3] http://www.arduino.cc/playground/Code/ID12

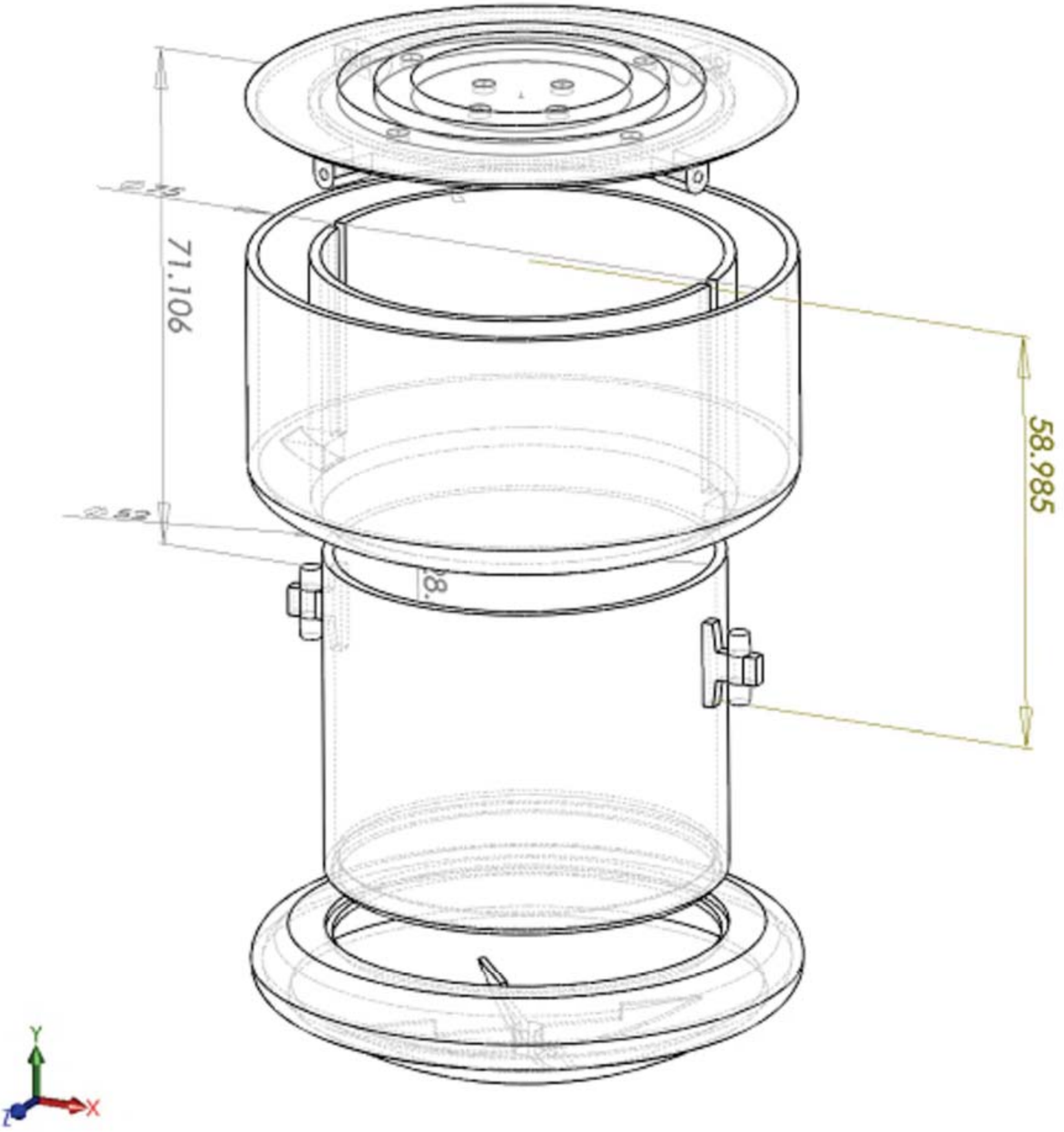


Figure B.1: Exploded view of the Connector

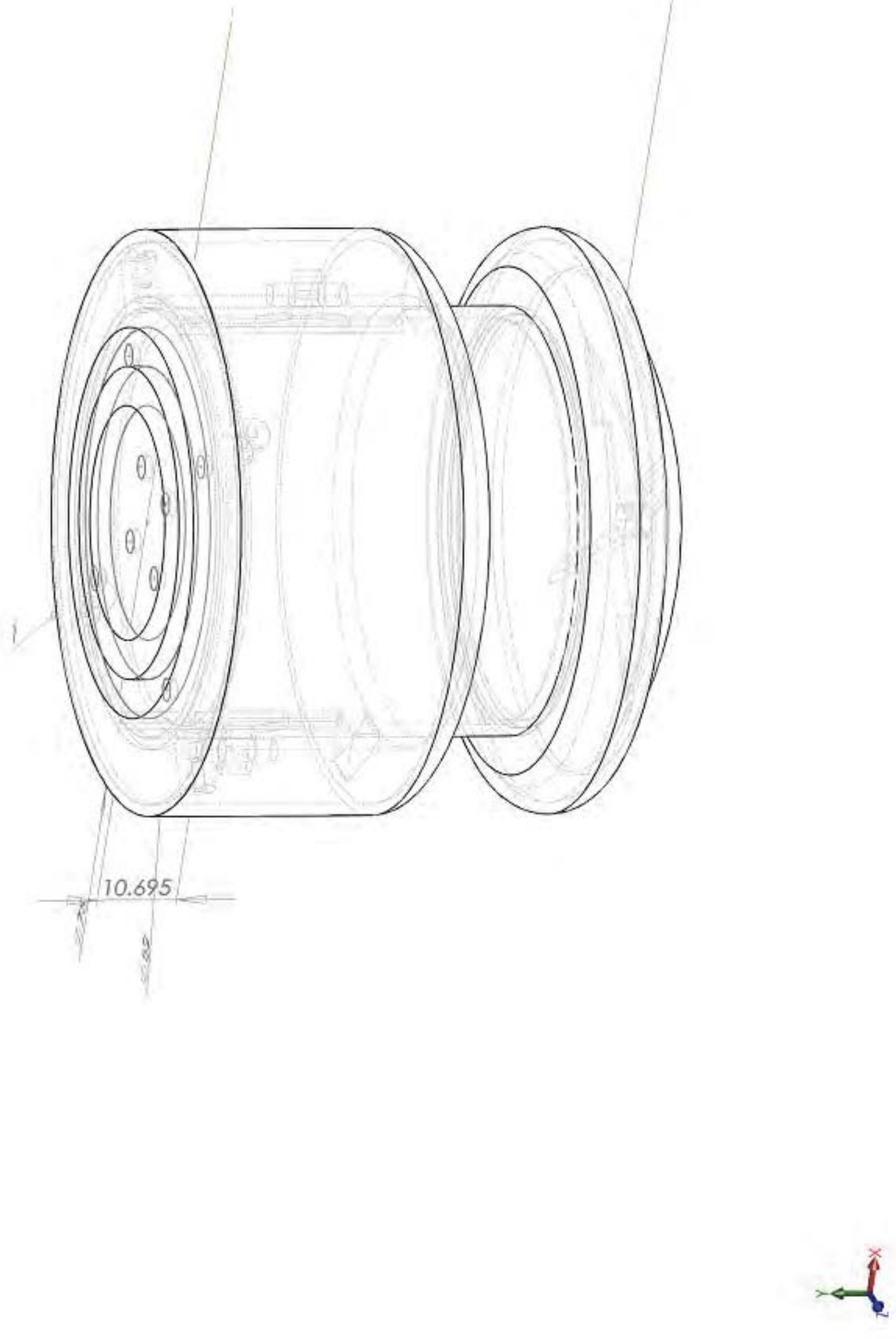


Figure B.2: A wire-rendering of the Connector's 3D CAD model

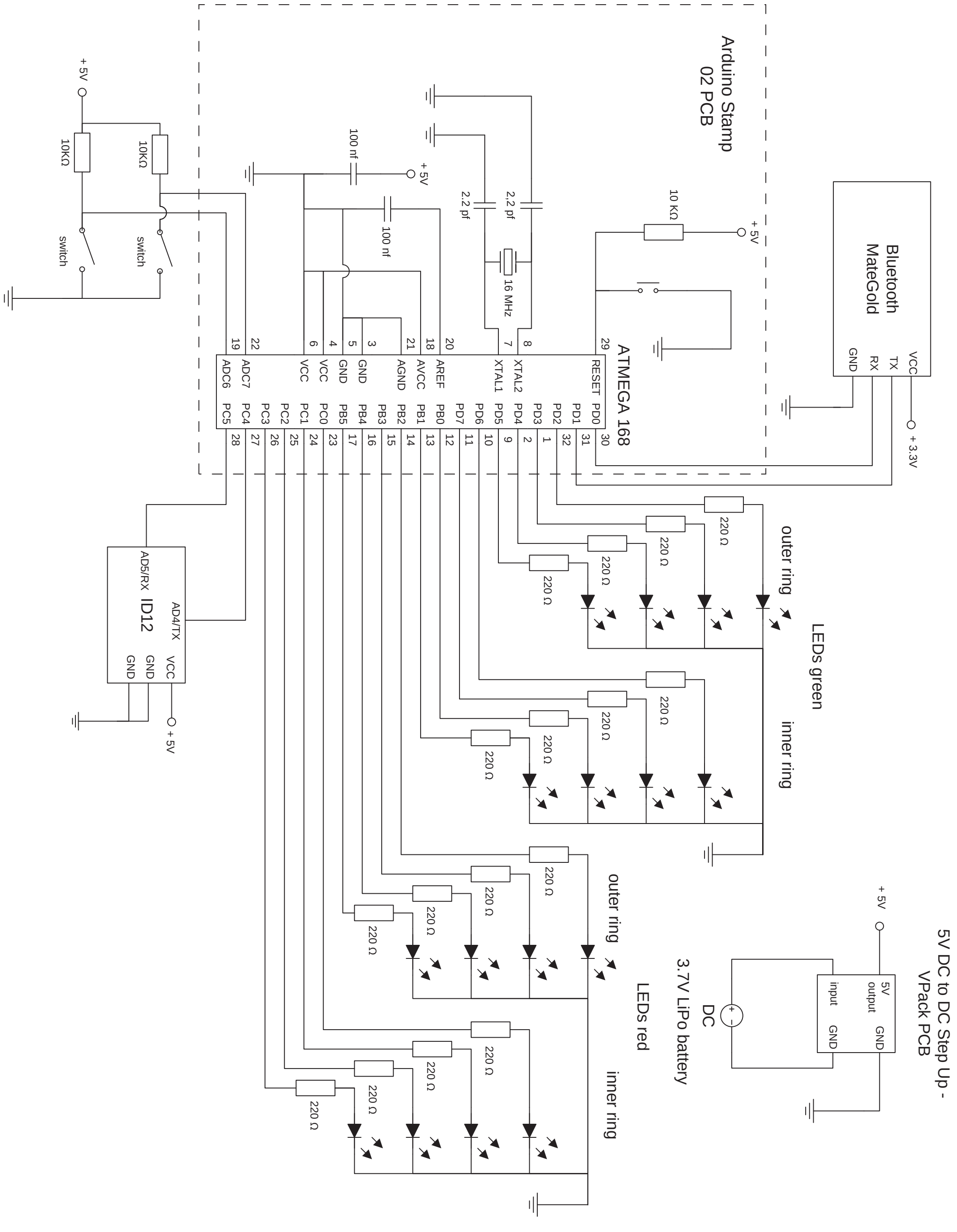


Figure B.3: Circuit drawing of the Connector hardware

Code example of the Connector software can be found in Listing B.1:

```
if (rfidPresent && firstDevice && !secondDeviceConfirmed)
  {
    deviceIDConfirmed = true;
    firstDevice = false;
    Serial.write("C");
    Serial.write("1");
  }
  else if (rfidPresent && secondDevice && !secondDeviceConfirmed)
  {
    deviceIDConfirmed = true;
    secondDeviceConfirmed = true;
    firstDevice = true;          // reset firstDevice
    secondDevice = false;        // reset secondDevice
    Serial.write("C");
    Serial.write("2");

    // wait while reasoning
    //we use analogue I/O ports since we're out of digital ones
    reasoning = true;
    SW_DOWN_MAP = analogRead(SW_DOWN);
    SW_DOWN_VAL = map(SW_DOWN_MAP, 0, 1023, 0, 1);
    SW_UP_MAP = analogRead(SW_UP);
    SW_UP_VAL = map(SW_UP_MAP, 0, 1023, 0, 1);

    while (reasoning)
    {
      serialComm();
      SW_DOWN_MAP = analogRead(SW_DOWN);
      SW_DOWN_VAL = map(SW_DOWN_MAP, 0, 1023, 0, 1);
      SW_UP_MAP = analogRead(SW_UP);
      SW_UP_VAL = map(SW_UP_MAP, 0, 1023, 0, 1);
      if ( SW_DOWN_VAL == 1 || SW_UP_VAL == 1)
      {
        reasoning = false;
        reset();
      }
      delay(100);
    }
  }
  else
  {
    //turn off LEDs
    if (ledRingState == 1) {} //except when LEDs are blinking
    else
    {
      LEDsInner(0);
      LEDsOuter(0);
    }
  }
```

Listing B.1: Code example form main loop of the Connector software

Code example of the ConnectorKP software can be found in Listing B.2:

```
def canConnect(device1, device2):
 global notFound

 if(notFound):
  return 'N'

 connectionPoss = 'N'

 qs = node.CreateQueryTransaction(smartSpace)
 result = qs.rdf_query([((device1, ns + "canConnectTo", None ), "uri")])
 print result
 for canConnDevice in result:
  if(device2 == canConnDevice[0][2] and device1 != device2):
   #Devices can connect
   connectionPoss = 'P'
   break

 node.CloseQueryTransaction(qs)

 return connectionPoss
```

Listing B.2: Code example for checking whether two devices are connected

Appendix C

Mental model abstraction

The mental models are elicited with two types of questions, "what is" questions and "how to" questions. Below, the participants' responses to these questions are listed. These questions were used to elicit the users mental models, and participants were asked to draw their mental models to support theirs answers to these questions. Figure C.1 shows the drawing of one of the Mark characters. Figure D.6 shows the abstraction that was made based on the drawing and the oral explanation.

Q: Can you explain to me what the system is?

System to share music. Easily connect devices together. Lighting effects linked to music.

Q: What are the components?

Connector, **motion sensor**, **Living Colours** *lamp (LC)*, **music player**, **lights** *behind the couch and a* **"central unit"**

Q: How is everything connected?

Mark: **Connection is between the music player and the LC** *because the connector "forgets" the connection again. If you want to disconnect the devices you have to scan the devices again. Lights (behind the couch) also turns off, but you don't connect those. But it does* **respond to the connection between the music player and the LC**. *Perhaps the connector is smarter than we think. Probably there is a* **central system** *somewhere.* **Everything is connected to this system.**

Sofia's LC is also connected to the central system, and can also render the music. Will her LC start flashing on the music automatically? I cannot imagine, because you have to make the connections explicitly (with the Connector).

Q: How does the system work?

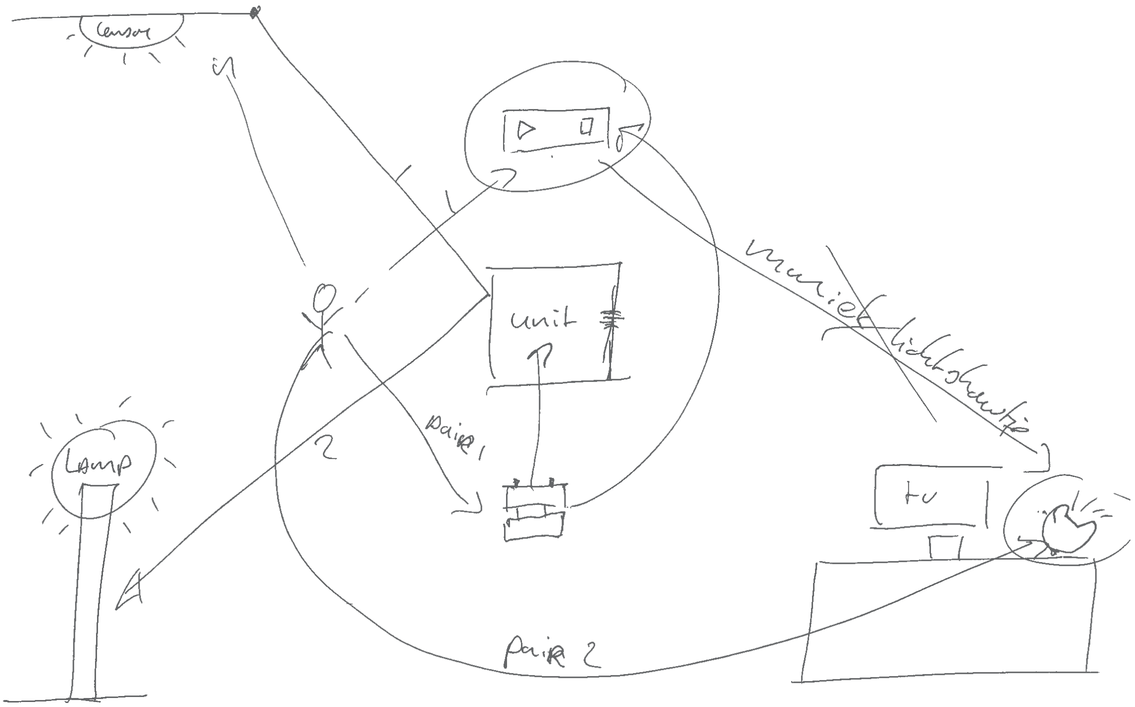


Figure C.1: Mental model drawing of Mark character 6

Mark: *When you enter the room, the various* **lights turn on; presence is detected by a sensor**. *With the Connector you select the mobile phone, device one, and then device two, the LC. Then you establish the connection by compressing the Connector. It doesn't matter whether the music is playing or not. You connect the sound from the phone to the LC. The connection is between the phone and LC because the connector "forgets" the connection again. If you want to disconnect the devices you have scan the devices again.*

There must be a "central unit". *We thought of that when we saw the lighting turn on automatically. So there must be something that controls everything centrally, the Connector only indicates which connections you want to make, the central unit then takes care of the rest. Or perhaps Sofia has been connecting everything upstairs? :)*

Dries: *A sensor registers that we entered and the lights (behind the couch) turn on. Using the connector, I pair with the mobile phone and after that with the LC. The Connector then instructs the "central unit" to connect these two devices together, and to turn of the lights behind the couch to start the lighting show. To disconnect the connections it's the same story, but then the other way around. Then, the lighting show stops and the lighting turns back on.*

I also think that the LC of Sofia will start shining brighter or less bright when the music starts playing here.

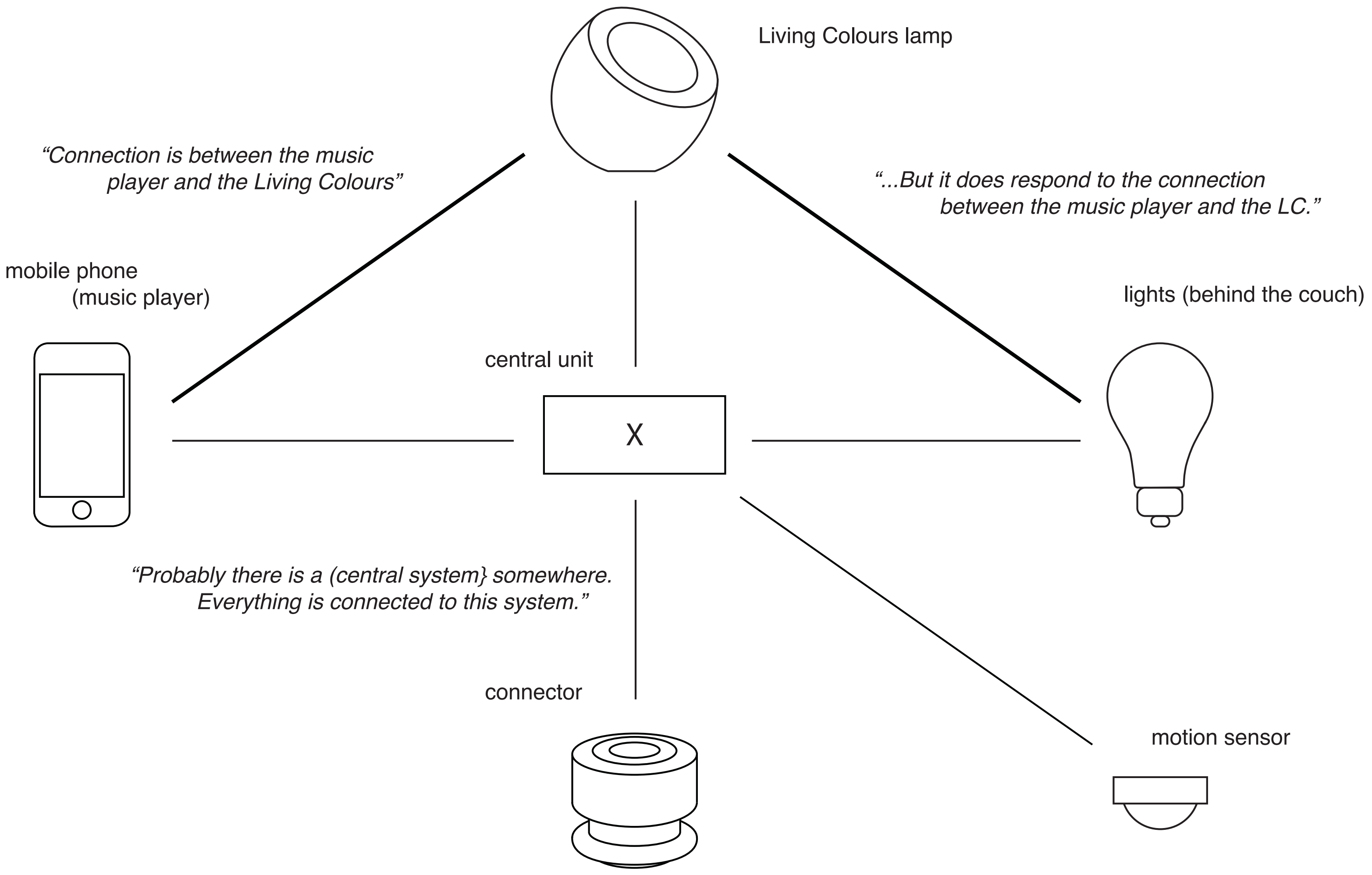


Figure C.2: Mental model abstraction of Mark character 6 and Dries character 6

Appendix D

Mental model abstractions Mark and Dries

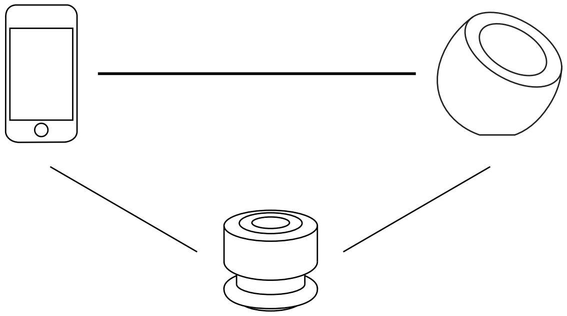


Figure D.1: Mental model abstraction of Mark character 1 and Dries character 1

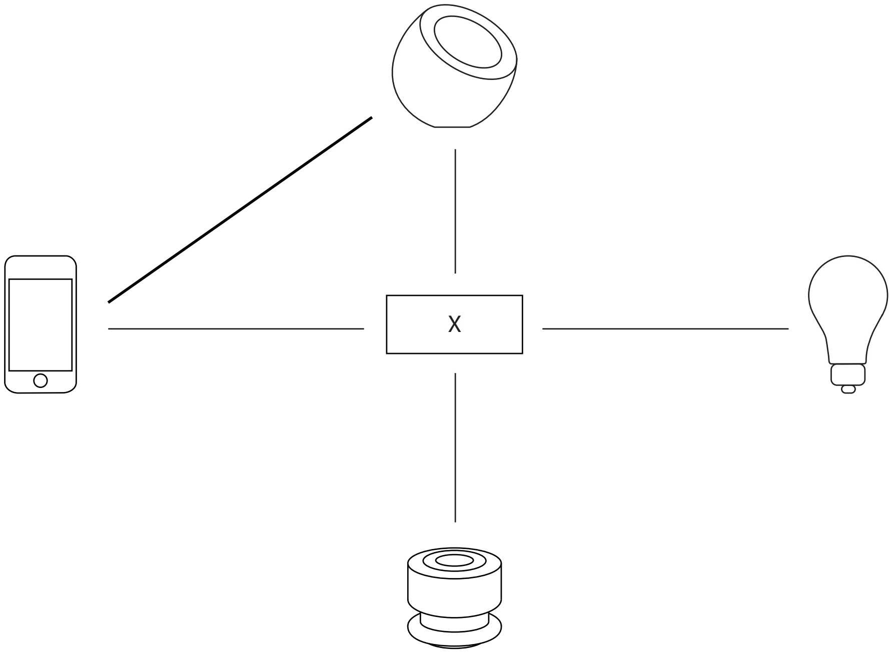


Figure D.2: Mental model abstraction of Mark/Dries character 2 (shared mental model)

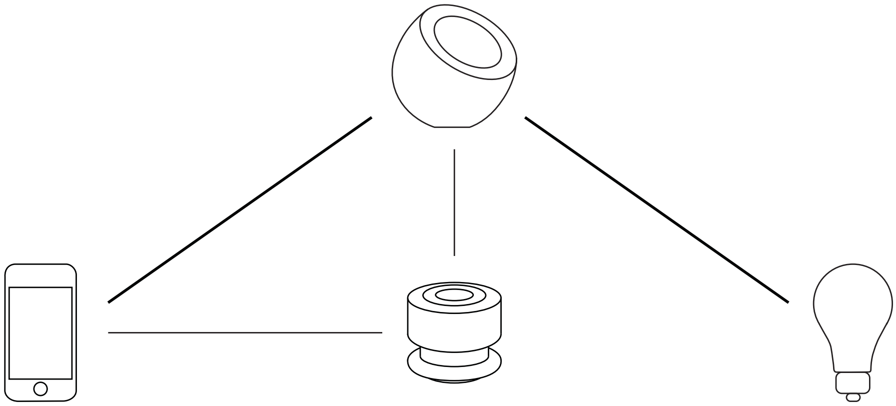


Figure D.3: Mental model abstraction of Mark/Dries character 3 (shared), Mark character 5, Mark character 7 and Dries character 7

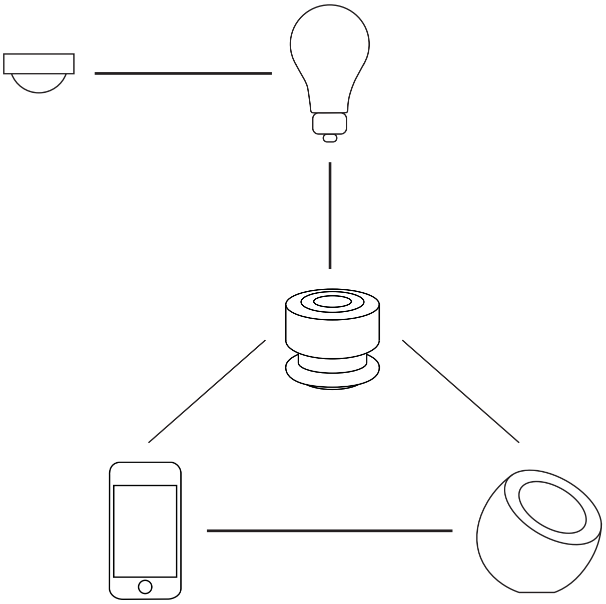


Figure D.4: Mental model abstraction of Mark/Dries character 4 (shared)

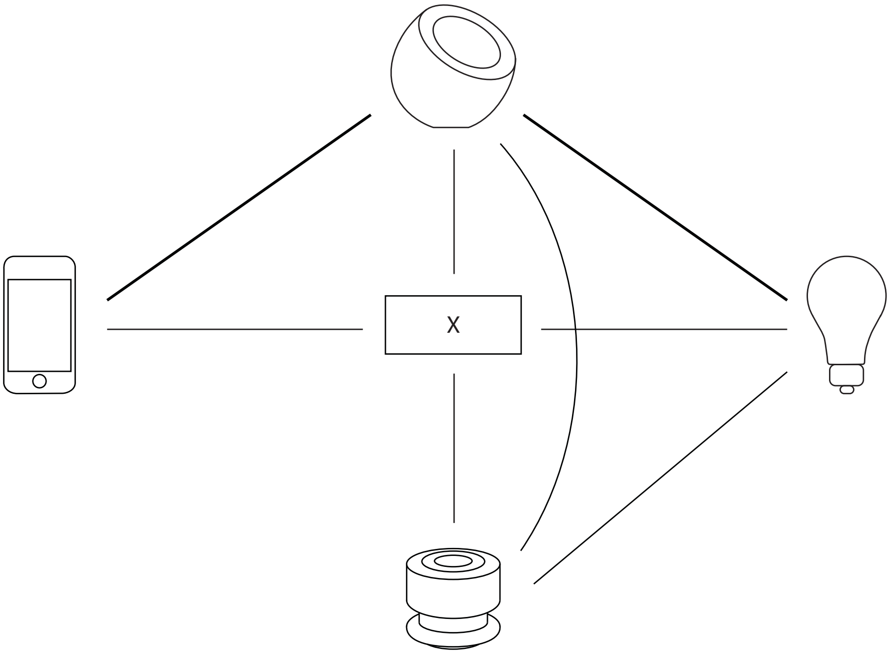


Figure D.5: Mental model abstraction of Mark character 5

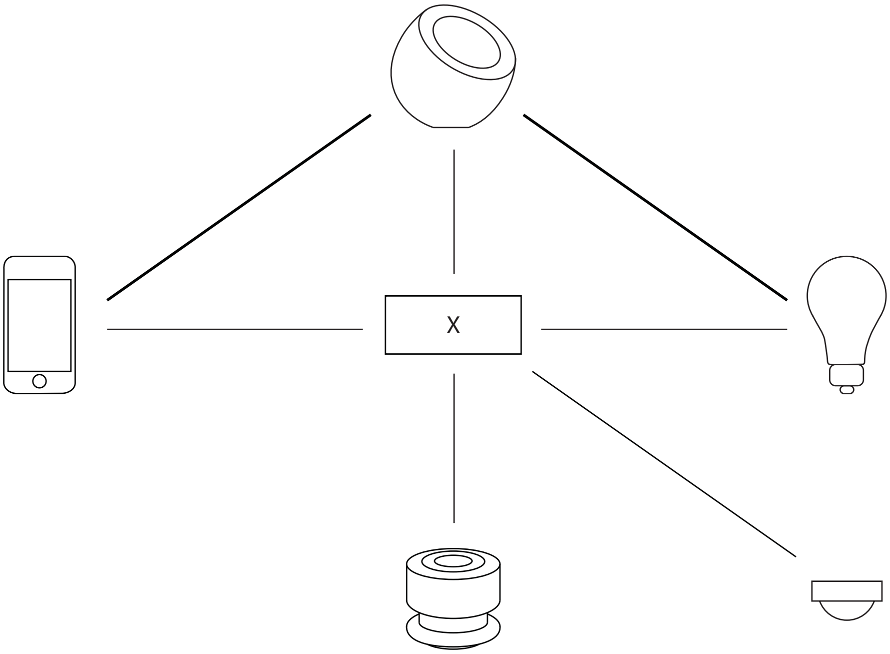


Figure D.6: Mental model abstraction of Mark character 6 and Dries character 6

Appendix E

Mental model abstractions Sofia

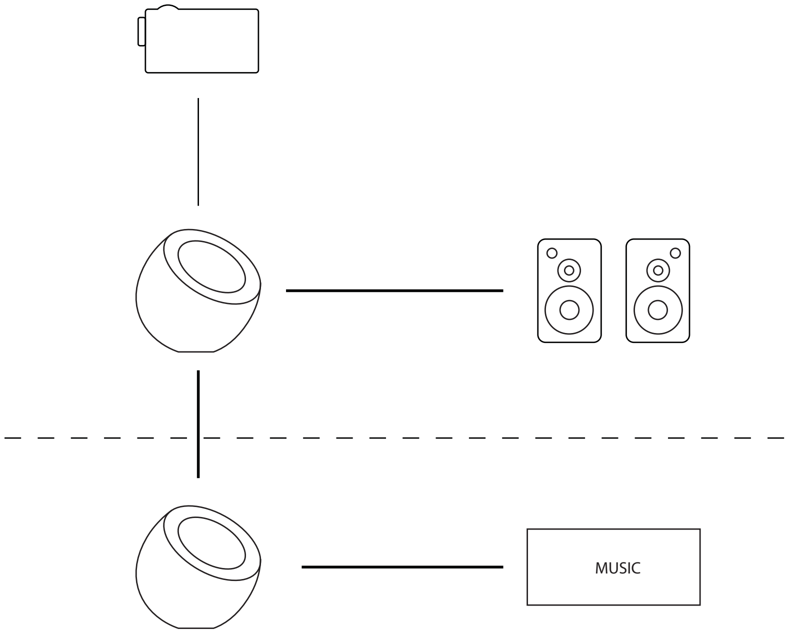


Figure E.1: Mental model abstraction of Sofia character 1

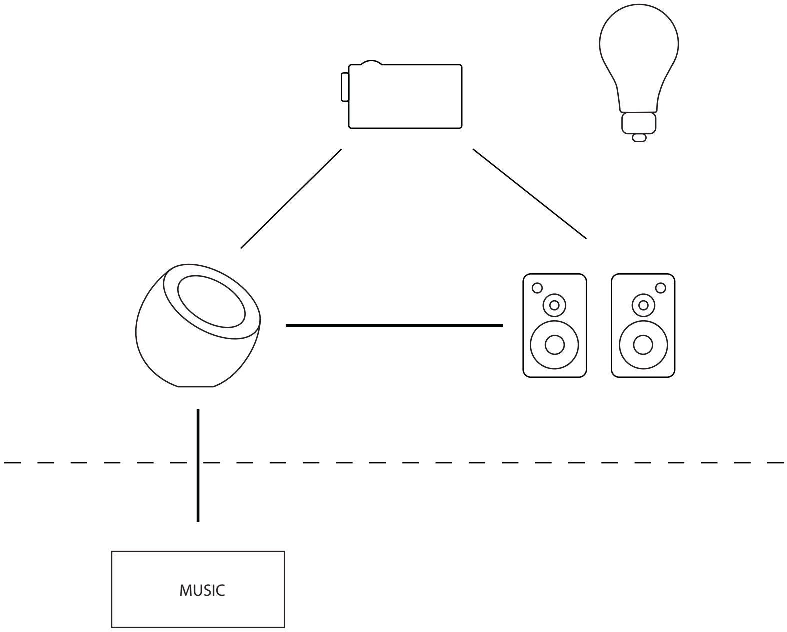


Figure E.2: Mental model abstraction of Sofia character 2

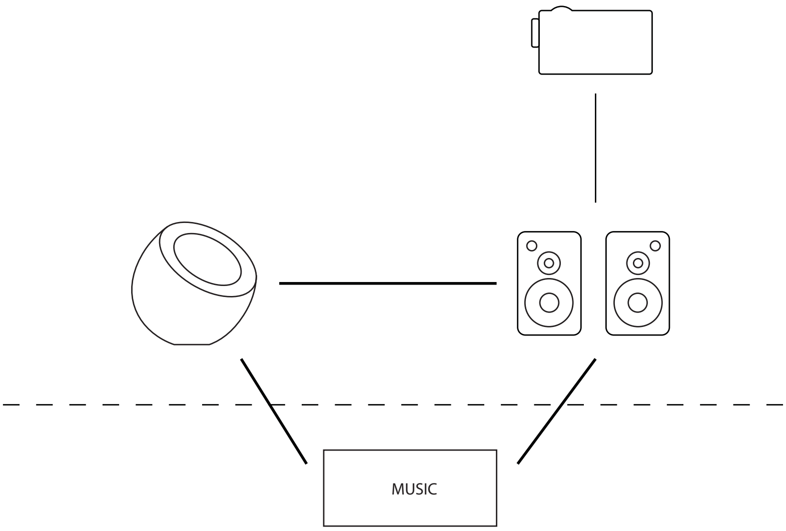


Figure E.3: Mental model abstraction of Sofia character 3

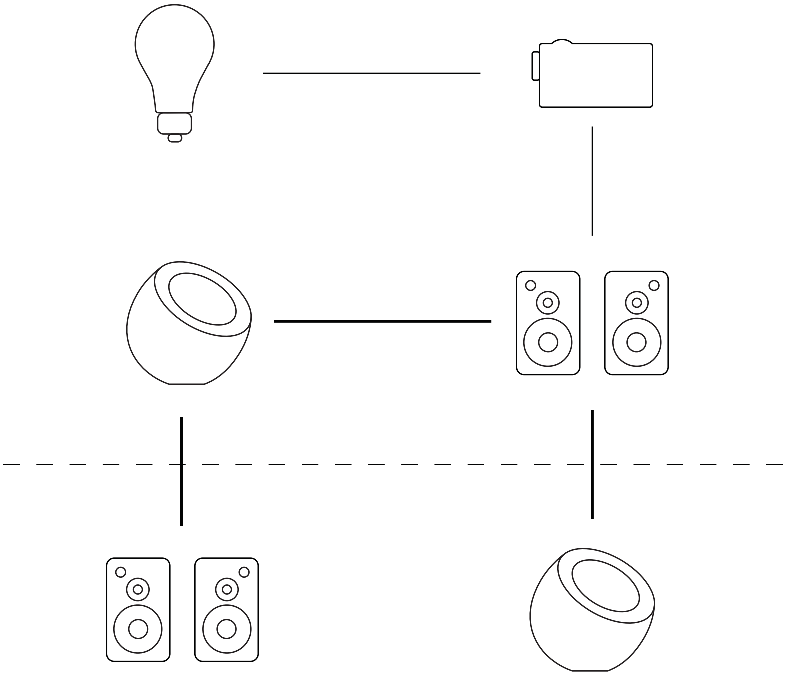


Figure E.4: Mental model abstraction of Sofia character 4

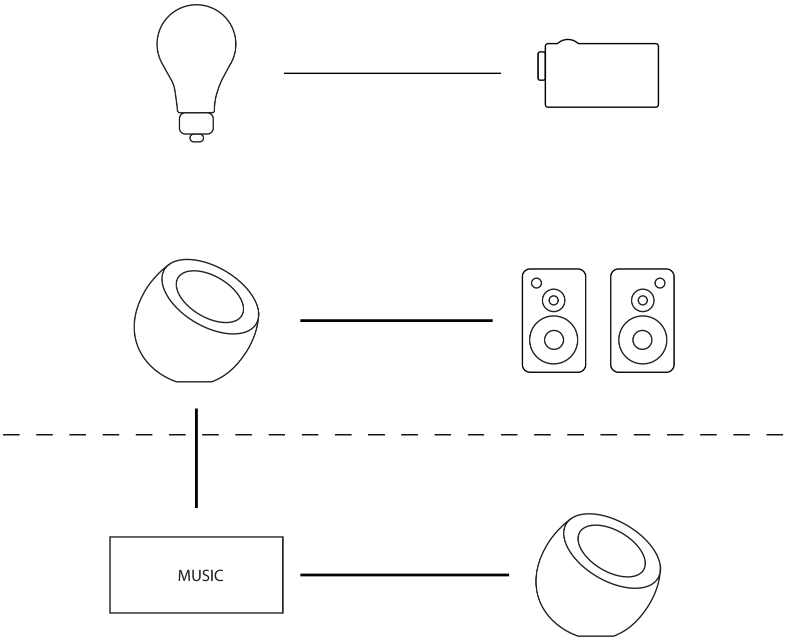


Figure E.5: Mental model abstraction of Sofia character 5 and 6

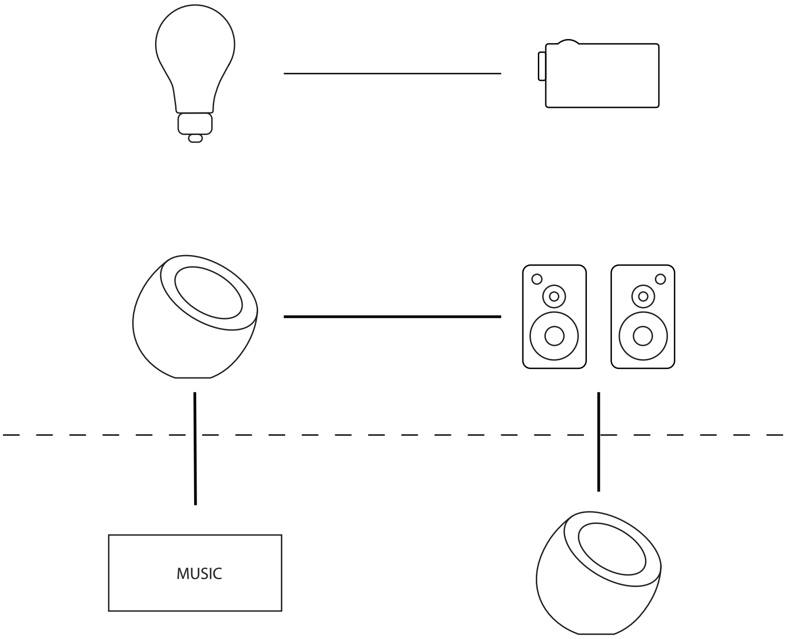


Figure E.6: Mental model abstraction of Sofia character 7

Appendix F

Semantic Connections Theory notation legend

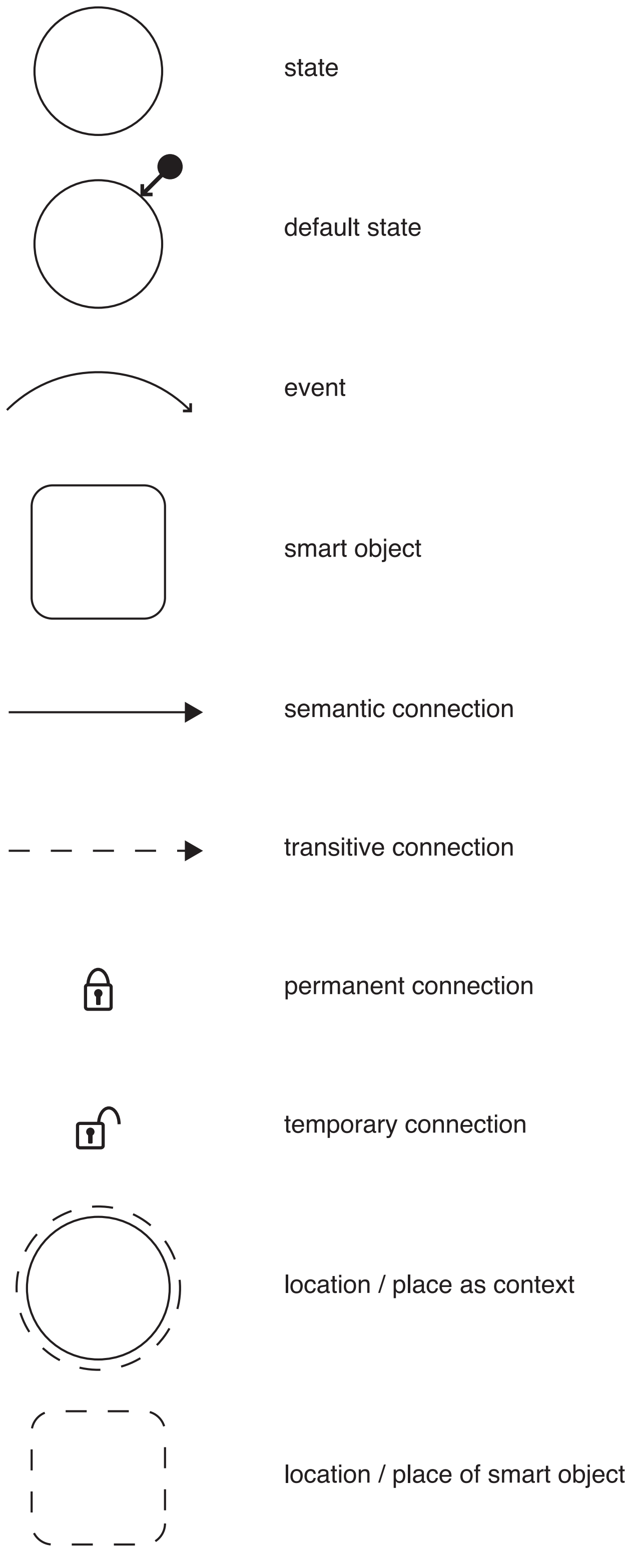


Figure F.1: Legend of the notation used in the Semantic Connections Theory. Based on (Thimbleby, 2007)

Appendix G

List of publications

A number of conference and journal papers were published during the course of the PhD project and are listed below:

Journal articles:

1. Niezen, G., **Van der Vlist, B.**, Hu, J., & Feijs, L. (under review). Semantic Connections Theory: Enabling Interaction Designers and Developers to Create Interoperable Smart Objects. *ACM Transactions on Interactive Intelligent Systems (TiiS).* 32 pages.

2. **Van der Vlist, B.**, Niezen, G., Rapp, S., Hu, J., & Feijs, L. (2013). Configuring and controlling ubiquitous computing infrastructure with semantic connections: a tangible and an AR approach. *Personal and Ubiquitous Computing, 17*(4) Springer-Verlag London. pages 783–799.

3. Hu, J., **Van der Vlist, B.**, Niezen, G., Willemsen, W., Willems, D. & Feijs, L. (2012) Designing the Internet of Things for Learning Environmentally Responsible Behaviour. *Interactive Learning Environments*, Routledge. pages 1–16.

4. **Van der Vlist, B.**, Niezen, G., Hu, J., & Feijs, L. (2011). Design semantics of connections in a smart home environment. *Creation and Design, 13*(2), pages 18–24.

5. **Van der Vlist, B.**, Bartneck, C. & Mueler, S. (2011). moBeat: Using Interactive Music to Guide and Motivate Users During Aerobic Exercising. *Applied Psychophysiology and Biofeedback, 36*(2), pages 135-145.

Conference proceedings:

1. Peeters, J. **Van der Vlist, B.**, Niezen, G., Hu, J., & Feijs, L. (2012). A Study on a Tangible Interaction Approach to Managing Wireless Connections in a Smart Home Environment. In L.- L. Chen, T. Djajadiningrat, L. Feijs, S. Fraser, S. Kyffin, & D. Steffen (Eds.) *Design & Semantics of Form & Movement (DeSForM) 2012.* pages 187–196.

Wellington, New Zealand: Koninklijke Philips Electronics N.V.

2. **Van der Vlist, B.**, Niezen, G., Rapp, S., Hu, J., & Feijs, L. (2012). Controlling Smart Home Environments with Semantic Connections: a Tangible and an AR Approach. In L.-L. Chen, T. Djajadiningrat, L. Feijs, S. Fraser, S. Kyffin, & D. Steffen (Eds.) *Design& Semantics of Form & Movement (DeSForM) 2012.* pages 160–169. Wellington, New Zealand: Koninklijke Philips Electronics N.V.

3. **Van der Vlist, B.**, Niezen, G., Hu, J., & Feijs, L. (2012). Semantic Connections: a New Interaction Paradigm for Smart Environments. In L.- L. Chen, T. Djajadiningrat, L. Feijs, S. Fraser, S. Kyffin, & D. Steffen (Eds.) *Design & Semantics of Form & Movement (DeSForM) 2012.* pages 16–26. Wellington, New Zealand: Koninklijke Philips Electronics N.V.

4. Niezen, G., **Van der Vlist, B.**, Bhardwaj, S. & Ozcelebi, T. (2012). Performance Evaluation of a Semantic Smart Space Deployment. *4rd International Workshop on Sensor Networks and Ambient Intelligence (SeNAmI 2012).* March 19–23, 2012, Lugano, Switzerland.

5. Niezen, G., **Van der Vlist, B.**, Hu, J. & Feijs, L. (2012). Using semantic transformers to enable interoperability between media devices in a ubiquitous computing environment. In Rautiainen, M. et al. (Eds.), *Grid and Pervasive Computing Workshops, Lecture Notes in Computer Science, Vol. 7096.* pages 44–53. Springer Berlin / Heidelberg.

6. **Van der Vlist, B.**, Niezen, G., Hu, J., & Feijs, L. (2011) *Interaction Primitives: Describing Interaction Capabilities of Smart Objects in Ubiquitous Computing Environments.* IEEE Africon 2011, September 13–15, Livingstone, Zambia.

7. Kwak, M., Niezen, G., **Van der Vlist, B.**, Hu, J., & Feijs, L. (2011). *Tangible interfaces to digital connections, centralized versus de- centralized.* In Z. Pan, A. Cheok, W. Mueller, and X. Yang (Eds.), Transactions on edutainment V (Vol. 6530, p. 132–146). Springer Berlin / Heidelberg.

8. Hu, J., **Van der Vlist, B.**, Niezen, G., Willemsen, W., Willems, D. & Feijs, L. (2011) *Learning from Internet of Things for Improving Environmentally Responsible Behavior.* In M. Chang et al. (Eds.), Edutainment 2011, LNCS Vol. 6872, pp. 292-299, Springer-Verlag, Berlin / Heidelberg.

9. Willemsen, W., Hu, J., Niezen, G.& **Van der Vlist, B.** (2011). *Using Game Elements to Motivate Environmentally Responsible Behaviour.* Proceedings of IADIS Game and Entertainment Technologies 2011 (GET 2011), Rome.

10. **Van der Vlist, B.**, Niezen, G., Hu, J., & Feijs, L. (2010). *Design semantics of connections in a smart home environment.* In L.- L. Chen, T. Djajadiningrat, L. Feijs, S. Kyffin, D. Steffen, & B. Young (Eds.), *Proceedings of design and semantics of form and movement (DeSForM) 2010* (p. 48–56). Lucerne, Switzer-land: Koninklijke Philips Electronics N.V.

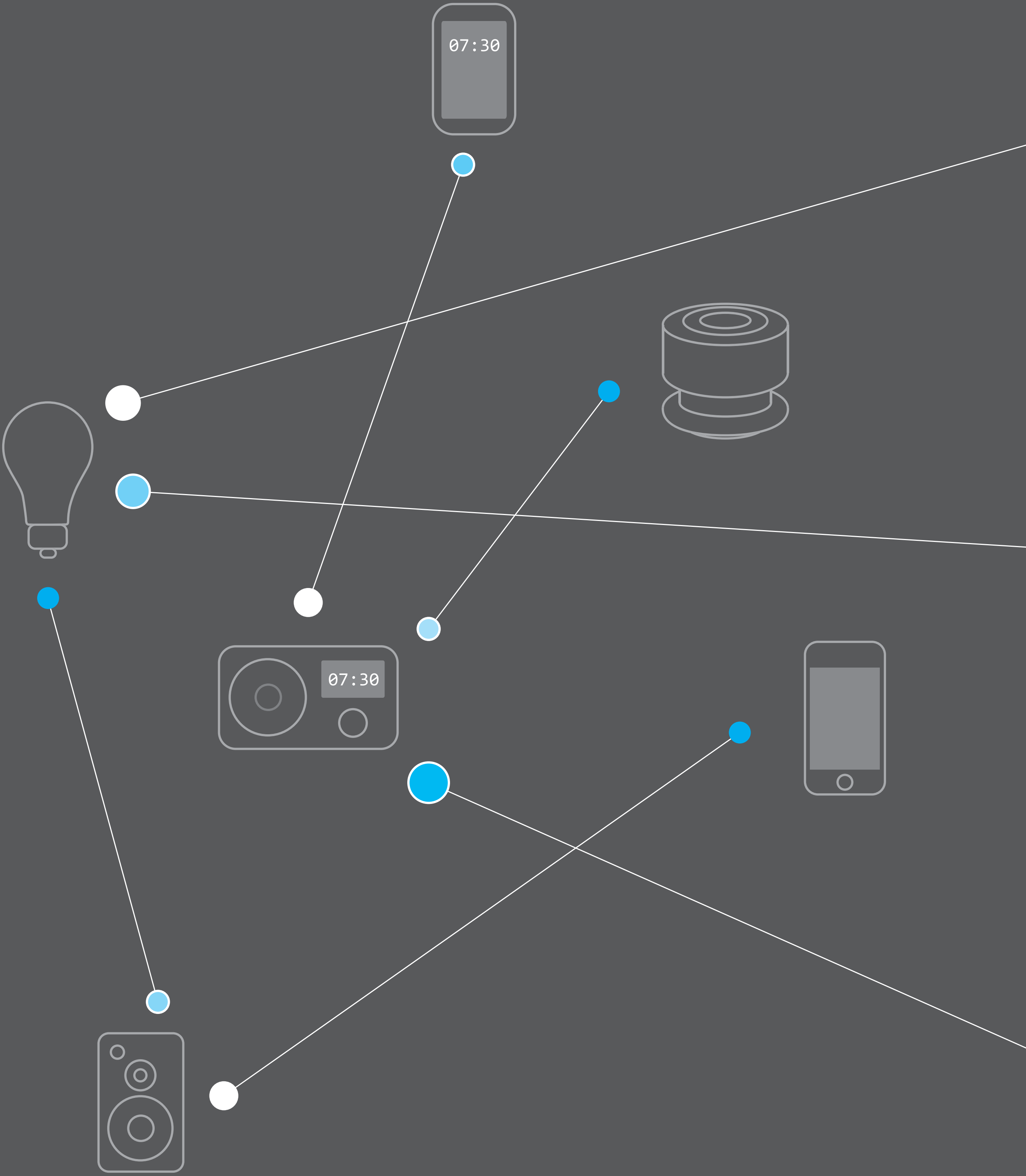
11. **Van der Vlist, B.**, Niezen, G., Hu, J., & Feijs, L. (2010). *Semantic connections: Exploring and manipulating connections in smart spaces.* In Computers and Communications (ISCC), 2010 IEEE Symposium on (pp. 1–4). Riccione, Italy: IEEE.

12. Niezen, G., **Van der Vlist, B.**, Hu, J. & Feijs, L.M.G. (2010). *From events to goals: supporting semantic interaction in smart environments.* Proceedings of the Computers and Communications (ISCC), 2010 IEEE Symposium on, (pp. 1029–1034). IEEE.

Appendix H

Curriculum Vitae

Bram van der Vlist (1983) studied Industrial Design at the Eindhoven University of Technology and received both his B.Sc. (2006) and M.Sc. (2009). He gained practical experience in industry during internships for his bachelor (TNO science & industry) and master graduation projects (Philips Research, department of Care and Health Applications) where he worked on wearable electronics and sports related applications. In April 2009 he joined the Designed Intelligence group at the Eindhoven University of Technology, where he was full-time appointed to the EU- funded SOFIA project. His focus within the three-year project was on the design of, and interaction with interoperable smart products. The results of this work are presented in this thesis. As of the end of 2012, he is working at TomTom International where he joined the User Experience team working on TomTom's navigation products.


07:30
07:30