Smarter Objects:
Programming Physical Objects with AR Technology

Valentin Heun

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning
on August 9th, 2013, in partial fulfillment of the requirements
for the degree of Master of Science in Media Arts and Sciences.

Abstract

This thesis describes a system that associates a virtual object with each physical object. These objects are connected to support a flexible and easy means of modifying the interface and behavior of a physical object. It also supports the creation of simple interactions with other "smarter" objects to create more complex functionality. As a user interacts with a physical object, an augmented reality interface visualizes the virtual object on top of the physical object and gives the user an intuitive graphical interface for modifying the object's interface and behavior or defining how it relates to other objects. As such, the system provides an innovative interface for programming the behavior of physical objects.

The contribution of the thesis lies in the meaningful separation of physical and virtual object interactions, a concept of direct mapping and the interprogrammability of Smarter Objects. The thesis presents the employment of Smarter Objects from all aspects and discusses a working implementation as well as several usage scenarios demonstrating potential applications. Finally, the thesis reflects on the implications of this approach for industrial design.

Thesis Supervisor:
Prof. Pattie Maes
Alex W. Dreyfoos Professor of Media Technology
Program in Media Arts and Sciences
Smarter Objects:
Programming Physical Objects with AR Technology

by
Valentin Markus Josef Heun

The following people served as readers for this thesis:

Thesis Reader: ____________________________
Hiroshi Ishii
Jerome B. Wiesner Professor of Media Arts and Sciences
Program in Media Arts and Sciences

Thesis Reader: ____________________________
Neri Oxman
SONY Corporation Career Development Professor
Program in Media Arts and Sciences

Thesis Reader: ____________________________
David A. Smith
Chief Innovation Officer and LM Senior Fellow
Lockheed Martin Global Training and Logistics (GTL)
Acknowledgements

Throughout my life, many people helped me form the mind and character that is able to write this thesis today. For their friendship, mentorship and community, I want to acknowledge these people.

Thanks:

To Pattie Maes, my advisor, for seeing both an designer and an engineer in me, supporting me with full freedom and great leadership to explore important concepts and ideas that otherwise would not have been explorable.

To my readers Hiroshi Ishii, Neri Oxmen and David Smith for strong thoughts, vision and discussion that made me dig deeper into my vision and concepts. I am honored to learn from your remarkable perspectives.

To Shunishi Kasahara for all the creative collaboration, talks, support, great friendship and for working together to create the foundation that eventually lead to this thesis.

To James Hobin for his assistance writing the code for the prototype. His expertise in mathematics and object-oriented programming has been crucial for achieving good results.

To Leonard R. Bacich for teaching me about his secrets of happiness and for helping me to trust my talents.

To Prof. Wolfgang Sattler for opening the doors that gave me opportunities to think and shift major directions in my life to finally pursue studies in Media Arts and Science.
To the MIT Media Lab, Bauhaus University and the Pratt Institute communities for helping me discover my talents and the Palomar5 community for letting me grow beyond the sky.

To Anissa Talantikite for never ending support, love and all the happiness that one never could write in such acknowledgements about.

To my family Christof, Jutta, Katharina, Kristina, Franziska, Frank, Jan, Florian, Leo, Kaya, Luke, my uncle Winfried and everyone in my greater family.

To all my friends.
# Table of Contents

1. **Introduction and Motivation**  
2. **Definitions**  
2.1. Object  
2.2. Smarter Object  
2.3. Virtual  
2.4. Physical  
3. **Related Work**  
3.1. Ubiquitous Computing  
3.2. Internet of Things  
3.3. Augmented Reality  
4. **Design**  
4.1. How do we use Physical Objects?  
4.2. Direct Mapping  
4.3. Interaction with Smarter Objects  
4.4. Data-Flow Interface  
4.5. Elements of the Visual Design  
4.6. Benefits of Smarter Objects  
5. **Technological Implementation**  
5.1 Overview  
5.2. Interaction Elements  
5.3. Data Structure  
5.4. Network  
5.5. Server  
5.6. Electronics  
5.7. Graphical User Interface for the Reality Editor  
6. **Prototypes**
6.1. Markers 55
6.1.1. Natural Materials 61
6.2. Radio and Speaker 68
6.3. Overlapping Interaction 73
6.4. Memory Bar 74
6.5. Lights 77
6.6. Mental Object Model 79
6.7. Data-Flow Prototype 81
6.8. Improving the Data-Flow Prototype 82

7. **Use Case Scenarios** 85
7.1. Waking Up in My Home 85
7.2. Sunday Morning Lawn Care 88
7.3. Watching the Game 90
7.4. My Surgery in the Hospital 92

8. **Evaluation and Discussion** 97
8.1. Critique and Evaluation by Professional Designers 97
8.2. Communicating virtual connections 100
8.3. Connected Complexity 103
8.4. Physical Gestures 104
8.5. Markers in Everyday Design 106
8.6. Impact on Industrial Design 108
8.7. What I have Learned 109

9. **Conclusions & Future Work** 111
9.1. Function 111
9.2. Technology 112
9.3. Design 112
9.4. Gestural Interfaces 113
9.5. Interaction 113
9.6. Vision for Future Work 114

10. **References** 116
1. Introduction and Motivation

When I arrived at the Media Lab for my studies in Media Arts and Sciences, all I needed were my clothes, my computer and a phone (Fig. 1) to feel at home. I could manage many aspects of my social life with the computer and see all the memories of past events by browsing through my digital photos. After a while, I realized that some aspects of my usual life were missing. It is quite difficult to access memories in my mind or interact with my environment only through a Graphical User Interface (GUI). To be able to meaningfully interact with the physical world around me, I needed to buy lots of kitchen tools, a bed, and bathroom supplies. After a while I also created lots of pictures and objects that I could place in my office to feel good. I needed physical objects to feel like a complete human.

The difference between humans and all other living creatures is the ability to create objects. Objects can be formed by describing them with spoken words. Objects can be physically present as tools that help us to operate the world. We combine mental objects in our mind to create new things or use such mental objects to form our self-conscience. As social creatures, we use objects to exchange knowledge by creating them as representations of our thoughts and passing them on to others so they can learn about what we think.

Having persistent objects is important for communication and for one’s self-confidence. Stable means that the object cannot completely disappear, lose power or data which would result in loss of its functionality. It also means that the object is easy to access for a fast
thought and communication process. The more a person is able to associate memories with physical objects, the faster and richer he can interact with other people, since all memories and objects one wants to talk about are directly accessible and visible through pointing, with words by talking about them, with gestures and muscle memory that lets us interact with them and physical touch that allows us to experience the object. As we create objects using our body, we form our mind. The body evolved as human's fastest interface.

Since we have increasingly transformed physical objects into digital objects, we have not found solutions to fulfill our need for physical accessibility; namely, for objects that can be operated with our body using a diverse combination of human senses. We can only access digital objects through a complex, unintuitive interface (keyboard, mouse or touchscreen) and the multiple shapes and functions we developed over centuries for all kinds of objects have gotten lost. It was easier to form a well-developed, social self before the digital revolution.

From my perspective today's dominantly used paradigms of digital communication - the use of keyboards, mouse and touch interfaces combined with GUIs, reduces any object or tool for the mind to the same identical form so everything can be accessed through a well-known medium. There are a lot of research examples that show other perspectives on how we can interact in a richer way with digital objects. But the dominant forms of digital communication have changed very little. We have traded the ability to communicate through richly defined objects for an everywhere and with everyone communication. If our physical objects disappear, what can communication ultimately be about? Communication will lack enough quickly accessible physical objects to be rich and effective.
The question addressed by this thesis therefore is:

*How can we design rich and diverse objects that reflect everything we are while at the same time offering the benefits of digital adaptability and flexibility?*

Based on my thoughts, such objects need to be at once flexible enough to reflect our expressions while being physically stable. If an object can become digitally linked with a virtual object, one can form methods for an expressive form of programming that provides a new form of control over physical space. Such a programming language needs to be easily accessible and operable so that everyone can adjust the world around them.

The innovative solution proposed in this thesis is to create physical objects that have a virtual component to benefit from their physical design while making them adaptable and extensible. I describe what I call *Smarter Objects* -- their creation, benefits and a programming interface to interact with them called the *Reality Editor*. Smarter Objects live in the physical and virtual world at the same time.

Following the definitions and related work chapters, the fourth chapter of this thesis explores a beneficial separation of user interaction with Smarter Objects so that the separation into the physical and virtual is intuitive. It then explores the concept of direct mapping so that a virtual object is linked to a physical object as if they are one. This discussion leads to a discussion of the benefits of Smarter Objects and the consequences for industrial design.

The Technology Implementation chapter 5 explores the foundation of physical interfaces as well as networking structure that Smarter Objects require in order to talk to each other. Since an everyday objects might not be technologically advanced, this thesis explores the possibilities of how such objects can still be integrated and used with in the Smarter Object system.
Smarter Objects builds on Augmented Reality (AR) techniques, and such technology requires visual Markers. In the prototyping chapter 6, I explore how Markers can be used to enhance Smarter Objects’ functionality. This chapter also provides a detailed description of diverse Smarter Object prototypes in more detail and shows the incremental steps that have led to the final system prototype.

Once the full system has been explained, four different stories introduce in chapter 7 possible use cases in which Smarter Objects can help a user to manage daily life. In chapter 8 a discussion with IDEO designers evaluates the thesis and produces a set of related questions which are later explored. Finally in chapter 9, a conclusion is drawn and future work introduced.
2. Definitions

Certain words used in this thesis have very specific meaning. For purposes of clarity, I want to start the thesis with a definition of those words.

2.1. Object

The word object can be used to describe many things, and speaking of a Smarter Object requires a concise definition. As I wrote in the introduction, objects can be seen from a mental perspective of thoughts up to an object consisting of stones or digital objects consisting of zeros and ones. Apples and Blackberrys can be food or they can be electronic devices.

I therefore want to narrow down the definition of what I use the word object for. From the perspective of this thesis, an object is an electronic device that is created to support an individual in daily physical work and life. Often such objects have functions invented before the digital age. These objects are still the main interfaces to our world but it is unclear how to properly integrate them into our current digital world. Such objects can be a dishwasher or car, toaster or light switches.

These objects have a common value. They can be operated with interfaces that have no other representation than physical buttons and sliders. Their meaning is to serve humans with a function that would be difficult or laborious to perform otherwise. If one wants to make a connection between them they need to be connected with ordinary wires. Many activities in our daily life have become easy by using such products. Smarter Objects discusses the possible digital supported interaction with these objects.
Current research on new electronic components blurs the line of what electronic means. Soon we will have electronic components that harvest energy from body temperature, or can become so small and cheap that every mug could measure the temperature of the coffee that is inside the mug. The definition for an object in this thesis will also be shaped and transformed by such research.

2.2. Smarter Object

When normal phones started to connect to the internet, the interface was unable to map the experience of the internet onto these phones. Such phones gave the feeling of being very stupid as they did not have enough functionality to render the webpages. Some companies tried to create a new internet just to serve such devices called WAP. Then with the rise of touch interfaces, faster processors and bigger screens, such services and bottlenecks became obsolete. So called Smart Phones are able to not only render the internet perfectly but also replace most of the functions one needed a computer for. This transformation represented a big benefit for every user.

As electronic components become even more small and more affordable, everyday objects as described start to connect to the internet as well. They were given the name Smart Objects or Smart Things in an analogy with the transformation from a normal phone to a smart phone.

However, a phone represents a tool for communication, whereas a dishwasher washes dishes. These are two fundamentally different goals. If a communication tool becomes more complex, so that the communication can become more complex, it serves its needs, but if a dishwasher becomes more complex it distracts from the original goal to simply wash the dishes. A manufacturer uses the word “smart” to communicate that the dishwasher has a certain complexity or even enough complexity to connect to the internet.
Smarter Objects are objects that fulfill both needs. Objects that have a certain complexity that enables more extensible functionality but also present a simple physical interface for the user.

Since the word “smart” is already familiar from the history of smart phones, I called this thesis Smarter Objects. The words “Smarter Objects” are chosen not to describe an object that is in any way more than another object, but an object that represents the meaning of what the word “smart” implies for an ordinary user in the first place. An electronic object that is physically more simple to use but provides a maximum of freedom and flexibility to become whatever complexity a user wishes.

2.3. Virtual

The word virtual is used to describe the components of a Smarter Object that are digital. These components are summarized as virtual object. This virtual object holds all states and functions that define the functionality of a Smarter Object and perform the connections with other Smarter Objects using virtual inputs and outputs. A 3D rendering of this virtual object is displayed with a graphical user interface to interact with the Smarter Object.

2.4. Physical

The word physical is used to describe objects that use electronics for changing there states and can be touched or manipulated with the hands.
3. Related Work

There are three major fields that focus on the interaction between the digital world and physical space.

The first relevant field is ubiquitous computing. Originally envisioned by Mark Weiser in 1991 with his work, *The Computer for the 21st Century*, he describes a vision in which computers are woven into the world so seamlessly that they cannot be distinguished from any other aspects of reality that form everyday objects.

A second field is the Internet of Things, first described as a term in an article *That ‘Internet of Things’ ‘Thing’* by Kevin Ashton. The concept developed less as a vision and more as a consequence of the internet and a saturation of smartphones that can connect to the internet. Electronics connected with the world wide network and are therefore operable with a smartphone from any location.

The third field is that of Augmented Reality (AR). It focuses on virtual interfaces that can be combined with the physical space. AR evolved in parallel with the processing power of computer and mobile devices and their ability to display 3D-Graphics. The paper, *History of Mobile Augmented Reality, Communications* explores the evolution of AR as mobile technology transformed its potential. In initial experiments, augmented reality algorithms needed to be calculated on an external machine, whereas on today’s phones, all processing can be done on the same device. *A Survey of Augmented Reality Technologies, Applications and Limitation* shows the traditional use cases of augmented reality used to superimpose information onto real objects and environments. The following sections describe the related work in each of these fields separately as there is not a lot of overlap between them.
3.1. Ubiquitous Computing

The paper *Situated Information Spaces and Spatially Aware Palmtop Computers*\(^\text{13}\) explores paradigms for interacting with everyday electronic objects through the use of mobile devices, but it is clear that a deep exploration into these electronic objects is missing. *Dual Reality: An Emerging Medium*\(^\text{14}\) explores the concept of fusion between the physical and the virtual space by filling the world with sensors of all kinds, so that a virtual space can create a better understanding of its physical counterpart and therefore enrich the physical space with electronic objects. As an exploration of space itself used as a projection surface, *I/O Bulb*\(^\text{15}\) researches the comprehension of digital connections in space, by augmenting virtual interfaces on every surface or by mapping digital information directly into physical space.

These related works reflect the foundation of Ubiquitous Computing in vision, interaction and projection. More specific work has been explored in the paper, *GENIE: Photo-Based Interface for Many Heterogeneous LED Lamps*\(^\text{16}\), which describes an augmented interface for changing the colors of lights. The camera of a smart device must be focused on a light that sends out an infrared beacon. Another work, *Galatea: Personalized Interaction with Augmented Objects*\(^\text{17}\) shows a digital augmented object that combines the benefits of software agents with digital sensors to proactively attract a user’s attention, once the user is in reach of the object’s it displays information that is interesting for the user. Augmenting books has been the focus of Galatea. *Tangible Programming Bricks: An approach to Making Programming Accessible to Everyone*\(^\text{18}\) explores the possibilities of using physical objects as modular building blocks that enable a user to create small programs using physical objects as the programming elements. The work explores whether tangible programming blocks are easier to use compared to graphical programming elements. These explorations reach into the physical space and show that there is a benefit to using the physical space itself as an interface. However, the paper *Whats next, Ubicomp? Celebrating an intellectual disappearing act*\(^\text{19}\) argues, that the right platform for programming and creating such interfaces has not been found.
The field of Ubiquitous Computing has enabled many solutions to be able to weave computation seamlessly into the physical space, yet there are not many design driven creations, since the tools are not made for designers who realize these ideas.

3.2. Internet of Things

With the advent of mobile smart phones and a saturation of internet connectivity, more and more internet connected things are commercially available such as Twine\textsuperscript{20}. Twine is a small internet connected electronic device. The device has sensors embedded in it that can be used for programming simple web applications that use the Twine sensor readings to trigger actions. Another internet connected object that is commercially available are Nina Blocks\textsuperscript{21}. Nina Blocks forms a system that enables simple communication with internet connected physical actuators and sensors. The system consists of a base station that connects wirelessly to a range of sensors and actuators. The system provides an API for programming new applications as well as an application that enable interactions with the connected objects.

Smart Things\textsuperscript{22} is a more sophisticated home automation system that enables the worldwide operation of home appliances. Every object that is connected with the Smart Things base station can be operated with a smart phone anywhere where an internet connection is available. The Philips Hue\textsuperscript{23} is another example of a simple internet connected light bulb that is commercially available. With an XBee\textsuperscript{24} radio, the bulb connects to a basis station and from there via ethernet to the local network and the internet. The bulb receives commands and sends status messages using the JSON\textsuperscript{25} protocol. Networking protocols are specifically designed for internet connected things such as the Extensible Messaging and Presence Protocol (Jabber/XMPP)\textsuperscript{26} a message oriented communication system for objects and users. It is an open platform based on XML and uses a client server system to connect users and objects to the system. The system is decentralized, meaning that a user can
connect via its server to a second user on another server, similar to the way an email is sent via a user’s mail server to the mail server of a second user. Other networks such as the AllJoyN\textsuperscript{27} follow a Peer-to-Peer networking concept. As a communication layer, the Internet \textsuperscript{0}\textsuperscript{28} explores a simple communication protocol independent from its carrier. A simple package encoding used by such a protocol enables local low power computation. The paper, *Social networks for lonely objects*\textsuperscript{29} explored the benefits of connecting objects spatially with each other to support the user in daily decision making tasks. From an Internet of Things perspective, interfaces, hardware and network systems have been explored in depth, however the related work falls short in examining how interfaces have effectively connect with objects.

### 3.3. Augmented Reality

The previous two sections have shown that there are many ways to fill the world with digitally supported physical objects, however there are problems with direct interaction. The concepts of interactions follow the paradigms of a desktop computer more than the interaction with physical objects. Augmented Reality as a visual connection between the physical and the digital is a field that demonstrates the potential to fill this gap.

*Object-oriented video: interaction with real-world objects through live video*\textsuperscript{30}, for example, shows that a real image with augmented screen interaction can provide a better intuitive interface compared to operations with a graphical abstracted representation of the physical object. Such interaction is used in a *Knowledge-Based Augmented Reality*\textsuperscript{31} that shows the Augmented Reality supporting maintenance of systems by an augmented knowledge data base. The maintenance of a computer printer is presented as an example. One step further, *Augmented Reality in the Psychomotor Phase of a Procedural Task*\textsuperscript{32} explores the psychomotor aspects of a head mounted display superimposing augmented reality assistance information onto physical objects. This work shows that superimposed assistance information allows a user to perform faster than if this information is shown on
a normal screen. Finally, the *Augmented Assembly using a Mobile Phone*\textsuperscript{33} shows a mobile application connected to a server application to provide step by step guidance for solving complex assembly tasks by means of superimposed 3D graphics.

Today such a system can be built with *Qualcomm Vuforia*\textsuperscript{34}, an Augmented Reality tracking framework that can be used with a variety of mobile systems. This tracker enables the user to track features in normal images, so that many generic objects can be used as a tracking marker. Mobile applications enable browsing through real spaces with *Layar*\textsuperscript{35} which is such an Augmented Reality browser application. One can generate virtual content that can be superimposed on magazine pages or other markers. A user can later use the Layar browser to view these contents. *Junaio*\textsuperscript{36} is also an Augmented Reality browser that allows the user to place information on top of buildings and other trackable objects. A user can then browse through the content superimposed onto the real world. Interacting with Augmented Reality is still a difficult programming task.

*DART: A Toolkit for Rapid Design Exploration of Augmented Reality Experiences*\textsuperscript{37} allows a designer to use the Macromedia Director (similar to Adobe Flash) to generate AR content in a very simple way. The programming of a DART application takes place on a two dimensional user interface on a computer but allows the fast prototyping for Augmented Reality applications. *The Immersive Mixed-Reality Configuration of Hybrid User Interfaces*\textsuperscript{38} illustrates the flexible configuration of physical interfaces in relation to 3D manipulations on a screen. A physical magic wand is used as the tool to connect interfaces with interactions on the screen. In a video recording of the interaction, data flow visualization illustrates created connections.

Another work that focuses on making the interaction with Augmented Reality more simple is *Immersive Authoring: What You Experience Is What You Get (WYXIWYG)*\textsuperscript{39}. This work describes an AR authoring tools named iATAR, a system that allows the user to manipulate the animation of 3D content augmented onto an Augmented Reality marker. This work focuses on entertainment content only and not the interaction with everyday objects. Finally another simplification system is represented by the
3D-Registered Interaction-Surfaces in Augmented Reality Space\textsuperscript{40} that shows a simple way to generate Augmented Reality interfaces. Adobe Flash can be used to generate AR interfaces, while at no point the user has to deal with difficult programing.

Some work explores the interaction with objects such as Remote-Touch: Augmented Reality Based Marker Tracking for Smart Home Control\textsuperscript{41}. Similar to the teletouch\textsuperscript{42}, this work explores a system to remotely operate a light bulb. It uses infrared beacons to identify the light bulb for interaction without mapping an interface onto the light.

There are many projects that try to use Augmented Reality to support the home entertainment system such as A Unified Remote Console Based on Augmented Reality in a Home Network Environment\textsuperscript{43}. The paper describes a unified remote console for consumer products augmented into the video image of a mobile device. The remote console shows the functionality of a generic remote control for each individual device once the camera of the mobile device is pointed at the consumer product.

CRISTAL: A Collaborative Home Media and Device Controller Based on a Multi-Touch Display\textsuperscript{44} is an augmented reality remote control system that shows a user’s living room from a bid’s eye perspective projected onto a living room table. On this table a user can interact with all consumer products shown in the room. A user for example can program visually the area in which a cleaning robot should move, or drag and drop a movie onto a TV so that the TV plays back the movie.

Other related work includes An augmented reality system for teaching sequential tasks to a household robot\textsuperscript{45}. This work shows a control system for programming everyday tasks performed by a robot. A user makes a photo of each task with the objects to be interacted with in sight. Through the identification of an augmented reality marker, a user can tell the robot certain actions. By serializing such task, the robot can be programmed.

All of this work has some sense of intuition of usability. However this usability has been explored with Evaluating the Benefits of Real-Time Feedback in Mobile Augmented Reality
This research shows that real time feedback systems deployed through tangible interfaces shows benefits compared to graphical user interfaces only.

Finally, there is work that blurs the boundaries between the physical and the real in a kinetic sense such as an ARPool table. When a user places his billiard cue on the table in order to move a billiard ball, the system analyzes the position of these objects and shows projected onto the table the most likely paths of all balls moving on the table. In robotics, AR has been used for controlling complex machines with TouchMe: An Augmented Reality Based Remote Robot Manipulation a virtual representation gets superimposes onto a physical robot. Once the virtual robot is moved by the use of a touch screen, the physical robot moves to match the same position of the virtual robot. A live feedback system is employed with exTouch: Spatially-Aware Embodied Manipulation of Actuated Objects Mediated by Augmented Reality. ExTouch is an Augmented Reality exploration for a operating vocabulary of moving objects such as robots using a visual touch interface and a visual feedback system loop. By focusing an iPad camera onto the object that should be moved, a sliding interface appears and a user can use his finger to slide the slider to a new position. The moving object then physically moves to this position until the physical object and the virtual slider are superimpose again.

In conclusion, all related work shows the benefits of combining the digital with the real. The Ubiquitous Computing field illustrates strong explorations on how to digitalize the environment, whereas the Internet of Things reaches into a networked user-driven digitally supported world. In this world, interfaces are controlled using mobile devices but without a convincing direct connection to the objects. The feeling of an object as one entity is missing. Everything is present everywhere at the same moment. The Augmented Reality space shows that it is beneficial to combine the physical and the real in such a way that the boundaries get blurred. Some research shows promising explorations into this field.

Smarter Objects reaches into all of these three fields. It uses the omnipresence of ubiquitous computing and the concepts of the Internet of Things, but solves interaction problems using Augmented Reality methods. This thesis shows a unique and
comprehensive fusion of internet connected ubiquitous objects programmed with an interface that uses Augmented Reality techniques for enabling a simple interface.
4. Design

Now that I have described the motivations for this thesis and the related work it builds upon, I want to discuss the design of Smarter Objects and the Reality Editor that is used to digitally interact with them.

In this chapter, I explore what the transformation into a Smarter Object means for the objects that surround us. I first discuss which kind of interaction with physical objects would be beneficial so that this interaction can seamlessly be supported, then I describe the design of the digital interface that is used to support the physical object with a digital layer. Finally the chapter discusses the benefits of such a design solution.

4.1. How do we use Physical Objects?

To understand the benefit of combining physical objects with virtual objects, I carefully examined how we actually interact with objects while using them. I determined that there are three major phases when interacting with everyday objects, and articulating those phases might help us understand the relative benefits of a Tangible and a Graphical User Interface. Finally, this understanding also leads to some ideas for combining the benefits of physical and virtual objects.

Let’s discuss the example of a kitchen toaster (Fig. 3). The first thing we do when using an object is understand its meaning and functionality. For a toaster this means that we first carefully look at the toaster. We want to understand all its features, where the buttons are so that we can interact with it effectively. Once the object, in this case the toaster, is understood, we want to customize it. We want to set it to our needs and preferences. With the toaster this means that we want to move the timing dial to the perfect position so that
the toast always has the desired level of browning. During the Setup and Understand phases, the object has our complete visual, mental and physical attention. Understanding and Setup of the toaster did not take much time, but we now have internalized how the toaster works and externalized our desired behavior of this object. The toaster can now be a part of my personal environment and I can operate it without much focused attention. The primary interaction with a toaster or any other object is its daily operation. This action only needs minimal visual attention, as the object is fully understood and is working as expected (Fig. 4). Since we took time to understand the object, this interaction quickly becomes memorized in gestures, skin feeling and muscle memory. The user does not need to carefully look at the toaster while operating it.

In contrast, graphical user interfaces are based on visual interaction rather than the tangible operation we are used to with everyday objects. For example, to turn on the radio on a smartphone, one needs to look at the display while an old-fashioned radio could be operated by muscle memory. Graphical user interfaces have an advantage, as they are highly flexible and very detailed. They can show text and images and can adapt to a variety of functionalities.

Figure 3. First a user needs to understand the functions of a toaster, then set the right settings and finally operate it with a physical slider to slide down the bread.
Unlike a tangible interface, a graphical interface can be self-explaining and can be used as a medium for communication in the same way it can be used as a control system to operate systems. Its operation is very dependent on our eyes and attention but can be very flexible and not limited by a physical design that is hard to change for new use cases. Following this thought, we can look at the hybrid object consisting of a graphical and physical interface that I want to create to make the following assumption: The understanding of an object is best supported by a graphical user interface. It can show very detailed explanations of how an object “wants” to be operated. The Setup phase benefits most from a graphical interface as it involves interacting with the object’s buttons or sliders to customize the object’s functionality. On the other hand, the setup interaction is performed only rarely. A graphical user interface not only helps to simplify the physical appearance of an object (how many sliders and buttons it needs) but also allows flexible modification of the object’s functions. For example, if I want my toaster to have a calendar functionality that I can program to automatically toast bread when I wake up, I could just add this functionality to an existing toaster without needing to add a physical button.

Another good example is a television remote control, as a user wants to have his eyes on the screen while operating it. Therefore the remote should be a tool fully operated by muscle memory. However, common remotes do not only provide buttons that can be used with the muscle memory to operate the TV, they also provide buttons for all setup procedures. Figure 5 illustrates how many buttons are actually used for setup procedures and how many are needed for watching TV. The right image shows the remote without the
volume and program buttons. All of the visible buttons are used for the setup of the TV. In contrast, the left image shows all buttons removed except the ones a user uses while watching TV. If we were to use a graphical UI as a remote, e.g. a smart phone, then setup and understanding would be easier and could hide for the time such functionality that is not used. But when it comes to the operation of the TV, the graphical user interface looses its benefits. The need for visual attention generates a conflict. The user wants to keep his eyes on the TV but cannot perform this task while operating through a graphical user interface. At the same time, those operational interactions most likely are very simple and usually do not demand much attention such as volume up or down. The most important benefit of a physical interface is that it can be operated with muscle memory. In the case of the TV remote, we can see that only four buttons are needed to actually operate a TV and since they are physical and a user can feel them, he can focus all of his attention on the TV while using the interface.

This discussions can be extended to any other object such as a kitchen blender, a radio, desk lights and cars.

### 4.2. Direct Mapping

Looking at the related work, there are different approaches for interacting virtually with the physical object. The most likely solution for everyday objects is to separate the virtual interface from the physical object. This simple step works for a few objects but encounters a fundamental problem once it gets scaled to an amount envisioned with ubiquitous computing. What if I have 20 lights, 5 radios, a refrigerator, 10 doors, a car and maybe even
sensors on my body that operate in this spirit? The interaction becomes very difficult when it is separated into a graphical user interface that is operated on the computer or touch device and a tangible user interface that is operated on the object. I would need to memorize all the connections between objects and software. For example, if one has 20 lights in a home, with a graphical user interface on a computer screen or smartphone, one would represent the lights by numbers, lists or with a categorization of symbols, but never mapped to the actual position in one’s home (Fig. 6). The lights could be named in the kitchen simply "light in the kitchen". But what if there are 3 lights in the kitchen? Of course they could have the name light one, two, and three. But it starts to become complicated, as the user needs to keep a mental image of the actual position mapped to the number in the mind. If the system gets even more complex with multiple lights, doors, radios and kitchen devices, it becomes impossible to keep all positions and relations in one’s mind. This shows that it becomes difficult to operate objects with separate interfaces.

Interaction with physical objects is most satisfying when the interaction that is performed directly maps to the action that is following\textsuperscript{50}. As a volume slider is moved up, the music gets louder. It is therefore important that the virtual and tangible user interfaces connected with an object are directly mapped. Since our world is a three-dimensional space, such a mapping must be performed in this three-dimensional space. Furthermore, for a perfect mapping of objects with data and programs, the used interfaces must follow the physical rules of our space\textsuperscript{51}.

One approach is to add a touch display to every object -- but this is too costly. An alternative solution explored in this thesis is to associate a graphical interaction directly mapped onto the object that needs to be controlled so that no mental relation between object and virtual interface needs to be remembered. A minimum amount of abstraction
and mental demand is achieved when a user has direct view on the object of interest and manipulates it on the spot, or where the user keeps an image in a device that shows the object and its surroundings. This is a direct mapping of operation.

The Reality Editor is a tool which supports programming Smarter Objects using such a direct mapping of operation. It thereby changes complex operations and programming into more intuitive tasks.

### 4.3. Interaction with Smarter Objects

The first thought for the interaction with Smarter Objects was a fusion of a virtual and a physical object. If we are able to combine a virtual with a physical object meaningfully, we can give the physical object functionality normally only known for virtual objects. This means for example that such a Smarter Object can inherit all the possibilities one can perform with virtual objects formed by object oriented programming. Within an object oriented programming language an object can inherit from other objects functions. These functions can be linked and remixed or an object can call functions of many other objects to present their functionality in a more simplified interface.

Therefore Smarter Objects and the Reality Editor have to provide, at its simplest form, the combination of a virtual with a physical object. This means whatever is happening to the virtual object has real-time influence on the physical object and vice versa. By combining the two components of a Smarter Object together, the Smarter Object reaches inseparably into both worlds and the interfaces can be truly mapped directly.

Since virtual objects can communicate with each other to form a computational system, the simplest form of communication
between Smarter Objects is a direct connection between these objects. Besides setting up one object with a direct mapped virtual interface, the Reality Editor should be able to make simple connections between two objects. The simplest connection that we can perform with electronic devices in the physical space is to connect them with a wire. This concept is visual and simple. For example, I can plug a electronic guitar with a wire into its amplifier. This allows me to combine the object guitar with the object amplifier.

If it were possible to connect more than one object, a simple radio can be presented with such a system. A tuner can be connected with a volume knob and then with a speaker. These objects virtually combined form a radio. On the physical level these objects could be seen separated, so that the radio could be controlled from within the kitchen, but a heavy speaker could remain in the living room. Or the radio could be connected with more than one speaker so that a full sound system can be distributed throughout the home. Also the tuning part could be given to a child so that he can chose the songs he likes, but the parents could control the volume of the sound.

Another concept for Smarter Objects is to combine objects not only with one and another but also with virtual logic. One such
logic is the simple conditional construct IF THEN. This construct determines if the output from virtual object A has met a defined condition, whereby the virtual object A receives its values from physical object A. If the condition has been met, then this small program continues with forwarding the data output from virtual object B to the virtual object C, so that the physical object C reacts on the inputs from physical object B if the physical object A has the right condition. In this case the IF THEN construct can be seen as a virtual switch for physical objects. Within the Reality Editor, the IF THEN construct can be represented with a virtual object placed close to the graphical interface of a Smarter Object.

With the Reality Editor, this virtual IF-object can be used to switch a connection by the use of a sensor, measuring if a door is open or close. A volume knob that regulates the volume from a tuning source for music can be connected to the IF object and to the input of a speaker. Here we have designed a simple program that demonstrates a system that would stop the music whenever the door opens. This can be used in an office if one enjoys listening to music in his office without disturbing colleagues. Whenever the office door opens the music will stop. This simple example shows the potential that lies in Smarter Objects that would allow one to reprogram the physical reality surrounding the Reality Editor user.

### 4.4. Data-Flow Interface

As previously noted, the connection between objects can be as easy as connecting a guitar with a wire to an amplifier. The connection between many physical objects and virtual objects would enable many new use cases for physical objects. The way such an interface will look like lies in the explanation of such concepts.
Smarter Objects should be able to be patched together or connected as easy as connecting a guitar to an amplifier. As there are many software concepts that use the concept of drag and drop or the use of computer programming languages, there are patching systems available such as pure data\textsuperscript{52} and MaxMSP\textsuperscript{53}. They work similar to the concept of patching a guitar to an amplifier or patching up a home entertainment system. These systems are called data flow programming systems, as you can see visually how the data flows through your patches. This is the closest representation of what a physical patch would do when connecting from one physical electric device to another.

Since this concept of patching wires can be used to connect objects in the physical space as well as to write programs in a data flow programming environment, it could be applied to Smarter Objects. While looking into different possibilities, it provided the most intuitive experience. With the Smarter Objects the data flow programming is used to patch virtual wires between physical objects to make the physical objects part of a virtual program.

In Figure 12, we see two small data flow programs represented. The left data flow program shows two switches (B) connected to an IF object (C). If the condition of the one switch is met, then the value of the second switch gets bridged to another object that can represent number values. Both switches are off, so that an object representing the result as a number shows 0 (D). The right program has two IF objects (C) that are connected with each switch (B). Since the switches are now both on, the numbers generated by a random number generating object (A) are bridged through the IF objects (C) so that the numbers that are generated gets displayed.

Figure 12. Image from the GUI of a Data-Flow Program.
For making such a programming environment possible with physical objects, every function of the physical object would need a specific spot that indicates a tag that can be used to draw connections from one to another device. Such tags could have in and outputs so that a user knows how to connect the objects. These tags could also indicate what kind of in or output the tag holds. For example, a kitchen mixer could have a switch that only allows 4 different speed settings. If I want a much more precise control for the mixer, I could make a connection from one of the nice dials found on a radio to the motor of the kitchen blender. This would enable me to control the kitchen blender with another interface, one that could better match my expectations. From a conceptional perspective, all essential elements of an editor for objects placed in the real physical space can be explained with these concepts.

4.5. Elements of the Visual Design

Now that I have discussed the beneficial separation into a digital and a physical user interface, the importance of direct mapping so that I can fill the physical world with Smarter Objects without the overwhelming need for memorizing virtual and physical connections, and the kind of interactions that are important for Smarter Objects and how they will be visualized, I want to show the design of the Graphical User Interface that enables the virtual use of Smarter Objects.

When building a physical radio or a desk lamp it is clear where to draw inspiration from in order to understand how such an object should appear. There are hundreds of examples all
around us in the physical world. We can modify those objects to test new functionalities, even before creating a complete new object ourselves. Finally, we can use those explorations to generate a new unique object.

But what are the things that we can draw inspiration from in order to form virtual direct mapped interactions with Smarter Objects? Since such Smarter Objects are new, there are no other established references that show a state of conceptualization or design in this field. One source that can be drawn from as inspiration are science fiction movies. Consider the introduction of multitouch interfaces with the iPhone -- there is consistent inspiration from science fiction. Similarly, the movie Minority Report\textsuperscript{54}, imagines fictional interfaces that motivated designers and engineers to think in new about touch interaction.

![Figure 14. Screenshot from the Movie Iron Man](image)

The interfaces found in science fiction have become a part of a general cultural memory, and new technology is often compared with science fiction. But such science fiction movies also inspire the creation of new technologies. One vision that is present in almost all modern science fiction movies is the idea of virtual interfaces being tangible and present in the room as if they were physical objects, or surrounding a person without a visible screen. This concept of interfaces peaked in the Iron Man\textsuperscript{55} movies. The movie shows a sophisticated concept of interaction with such virtual and tangible interfaces. Interfaces that are pixels in free space might be difficult or even impossible to create. With related research in mind this kind of interface is to far away as that a virtual pixel interfaces could become tangible and interactive in the free physical
space. At this time, the closest functioning research for such interfaces are projects in the MIT Media Lab Tangible Media Group. Projects like the Sublimate\textsuperscript{56} and Recompose\textsuperscript{57} show tangible pixels, but these pixels lift out of a table that gives them physical strength. Since these tables are stationary, it might not be functional to have virtual tangible pixel interfaces as an interface for everyday objects, but this thesis shows that it is possible to augment physical objects with virtual objects so that the physical world becomes supported by the virtual space.

Comparing the graphic design from Iron Men with those of Avatar\textsuperscript{58} by James Cameron, Prometheus\textsuperscript{59} by Ridley Scott and interfaces from the movie Tron: Legacy\textsuperscript{60}, we can see, over and over, one appearing color theme as well shapes indicating the look of science fiction. This look is designed for interfaces that are not bound to a screen but floating in physical space. A clear color language indicates information in white, neon blue or saturated yellow colors. These color themes separate the interfaces perfectly from the world which they get overlaid. This is where science fiction crosses the lines to research, because when exploring the possibilities with overlay graphics for camera images, overlay interfaces need to appear in shapes and colors that divide the interface clearly from the underlaying image. When analyzing the screenshot from Avatar one can see how nicely the interfaces separates from the brown and green colors of the tree. In other movies, white is used to create an underlying frame for interfaces such as in Prometheus.

Figure 15. Screenshot from the Movie Avatar
Looking at the used shapes, we see a lot of simple round shapes as well simple polygons. The interface needs to be complementary to nature. Instead of showing a multicolor icon world as we can see it in traditional graphical user interfaces, these overlaying interfaces must compete with a multicolor world that is full of shapes of all kinds.

Clear simple shapes are not present in the real world, so that these shapes always pop out and provide a good contrast for interactive elements. We can see this kind of language as the foundation for the visual style in Joseph Kosinski’s version of Tron (Tron: Legacy) where the full set is one big sterilized computer interface.

Figure 16. Screenshot from the Movie Prometheus

Figure 17. Screenshot from the Movie Prometheus

Figure 18. Screenshot from the Movie Tron: Legacy
Based on these interface explorations, I designed the Reality Editor interface components to follow a clear visual language of simple circles and polygons as well blue, yellow, clear, red and green colors. Every color has its domain. Green indicates positive events, whereas red indicates negative events. Yellow indicates active buttons and value representations. A light blue is used for the outline of all elements. White is used to support shapes as a background or to indicate empty space for interactions and value representations.

4.6. Benefits of Smarter Objects

I have explored the main components that need to be considered to design Smarter Objects as well as the visual and interaction design for the Reality Editor. As a last open question I want to discuss the physical appearance of a Smarter Object.
A Smarter Object meaningfully divides the interactions with an object into a virtual and a physical interface by distinguishing between understanding, setup and operational tasks. This distribution has tremendous impact on the appearance of the design for a physical object. When looking at an object like an amplifier (Fig. 20), we see different sizes of buttons and different places where those buttons are located. In a good design, the design of the interface tries to guide the user.

More frequently used buttons and knobs have bigger sizes or are located at significant locations. Less frequently used buttons usually are smaller and located more in the background as we can see in the upper device in Figure 20. From a GUI perspective, such user guidance can be seen for example in the software MacPaint.61

In the 1980s and 1990s, electronic devices such as VHS recorders were designed in a way that less frequently used function are hidden behind a flap that could be opened to access those functions. This presentation of primary and secondary functionalities led to very clear designs that could be easily operated. But this solution is not optimal, since sometimes a secondary function becomes primary for a certain user. Furthermore functions that are less used need to be very simple to understand since they are never operated and not memorized by the user. In the case of the VHS recorder, these functions were hidden because of their complexity, which made operation of the device even more complicated.
The language of interfaces evolved as a consequence of too many features. Designers exposed important functionality on the front panel and hid additional features behind sub-menus that can be accessed by pushing certain buttons a couple of times. With Smarter Objects and the distribution of interaction, those set up procedure functions can be moved to a virtual interface, similar to hiding functions behind a flap. Since they are now virtual, a GUI can be used to give detailed explanations about their functionality. A user can operate functions that are complex and less used, since the GUI explains the functionality at any time as it is one of the benefits of GUI. In addition, unlike a fixed designed VHS recorder, with Smarter Objects the physical buttons can also be easily reprogrammed so that a user can choose a desired setup supported by a GUI. This leads to a very clean design. Looking back at the bottom image of Figure 20, we can see how such a redesign could change the design of an object.

Aside from an on and off switch, an amplifier might only need one dial to choose a program and one for volume. The two dials are flexibly reprogrammable, so that a user can set up the device to his desired functionality. The amplifier can now be much smaller and the interface itself can change radically. For example, the limited amount of dials could become the casing itself, where a user sees a solid case with an on and off switch and two extruded parts on top of a solid base that can be rotated. This object would still represent all the features of the first described amplifier, but unlike that amplifier there is no need to expose the controls by the use of a flat control panel that holds all the necessary buttons.

Figure 21. A simple amplifier with dials for volume and program.
If there were no Smarter Object such a step would complicate the control of an amplifier. Since the understanding about the object and its functionality has shifted with the Smarter Object to a GUI, a user can look up all the explanations and functionality in a GUI that tells him at any moment how to use the interface. The user can make changes supported by the GUI and only need to memorize the simple operation of the physical object. Compared to the complex interfaces of VHS recorders or washing machines that require a manual to understand the functionality, such a Smarter Object could have more features, operated in the physical world and be self explaining by the use of the GUI.

The Smarter Objects approach simplifies the physical features of an object thereby liberating the design of an object from the function of a shape so that a designer has more freedom to experiment with shapes.
5. Technological Implementation

In the previous chapter, I have discussed all the design elements that need to be considered to build programmable Smarter Objects. In this chapter, I explain the technological components that enable programming with Smarter Objects and the Reality Editor. First I provide an overview of all the necessary components and later I explain these components in detail.

5.1. Overview

Smarter Objects can be seen as a system of three computational hardware components. A server, the Smarter Objects themselves and a touch device. The Smarter Objects server is a central computer that negotiates the data of all Smarter Objects and touch devices that run the Reality Editor. The server holds a list of all present objects and their functionalities so that it can change the functionalities of the Smarter Objects once it receives commands from the Reality Editor. The server also collects all the inputs from the Smarter Objects in real time and forwards these inputs to the Reality Editor if necessary.

The Smarter Objects, from a technical perspective, provide the user with the physical user interface. The inputs from the interface are digitalized and if a change to the interface occurs, the Smarter Objects sends this changes to the Smarter Objects server. The actuators, such as lights, motors or screens of the
Smarter Objects, are connected to the serve, that controls the behavior of the Smarter Object at any time. The Smarter Objects hold an Augmented Reality marker on its surface so that the Reality Editor can directly map a virtual interface onto the physical Smarter Object.

The final computational hardware component is the iPad or any preferred touch device. In our case the iPad is preferred as it is a simple tool to program for with Open Frameworks. This device has a camera on its back side so that a camera image gives the illusion that a user could look through the device. It has a touch screen on the front that enables finger interaction with the screen. The device runs the Reality Editor and therefore needs to be able to process the camera image to generate three dimensional images in order to map the virtual interface onto the physical object. It is wirelessly connected with the Smart Reality Editor and mirrors the objects that are stored in the server so that it can use this information to visualize all connections and interactions. Finally with its touch interface, it receives control commands from the user and decides if these commands need to be forwarded to the central server.

Aside from the computational hardware components, this chapter also discusses the communication that occurs between the components, the interface that lets the Reality Editor combine the physical object with a virtual object and the functionality of the Reality Editor in detail.

### 5.2. Interaction Elements

In order to make objects interprogrammable, we need to understand what component an object consists of and how these components should communicate between all the Smarter Objects components. Let’s return to the example of an amplifier. The amplifier’s interface consists of a display that gives feedback about controlling the features of the amplifier. We
can see this display as a media source that can represent pixel information to form text. The amplifier has various audio inputs and outputs that can follow various audio standards. We can summarize all of these video and audio inputs and outputs as media streams. Beyond those inputs and outputs, an amplifier has several buttons. A button has two states: on or off. There are buttons that can be pushed to hold one of the two states until it is pushed again, or such buttons have a state only as long as they are pushed. Buttons always trigger actions that influence systems. An amplifier has another interface, represented by rotation knobs or sliders. These sliders have many different states depending on which position they are moved to. In contrast to an action, these interfaces can represent motions that a user can perform over time, or they can be moved to hold a specific state of the slider. These sliders can have mechanical components that make them snap into increments or they can be moved seamlessly. These are all the essential parts an interface can consist of, and using these essential interaction components, any device interface can be constructed.

As another example, even a touchscreen itself could be made up of these components. The touchscreen consists of a screen, representing a medium and an action, that is performed once the finger touches the surface. The action stays active as long as the finger is touching the surface. Furthermore, the touchscreen inherits two sliders: one for the x and one for the y axis. These sliders change their value once the finger touches and moves over the touchscreen surface. Any kitchen device, light switch or complex machine can be broken down into these components. This means that if I generate a system that can understand communication media, motion and action between interfaces, actuators, speakers and screens, every object can potentially become programmable.

5.3. Data Structure

Based on the previous discussion, I simplified the data structure for Smarter Objects so that the interconnectivity between Smarter Objects can be solved simply. I separated the data communication for the Smarter Objects system into media streams and value streams.
A value stream consists of an integer value between 0 and 255 that is sent 30 times per second. If the value stream represents a button that only has the states on and off, the on represents 255 and the off represents 0. A slider uses the full range between 0 and 255. A Smarter Object always consist of 6 input and output values. Every Smarter Object always sends or receives 6 values. If an object only has one input, then only one of these 6 values contains data. This structure simplifies all possible objects into one value stream for each object. The Reality Editor always has internally identically defined in and outputs, so that a minimum of complexity and errors occur when connecting Smarter Objects.

5.4. Network

These values are either communicated from an object to the server using a USB serial connection, a wifi connection with a communication protocol called Open Sound Control (OSC)\textsuperscript{62} or an XBee\textsuperscript{63} connection using JSON\textsuperscript{64}. The data communication between the server and the Reality Editor has more complexity, as it communicates which objects it sees and which objects should be connected. The Smarter Objects themselves do not know about their connection with other objects. The server knows these connections and changes the behavior of each Smarter Object accordingly.

Such a structure is called a star networking structure. This means that the system cannot work if the central server fails. To enable stable Smarter Objects that will always function, this kind of structure is not the most beneficial, but functions well for prototyping Smarter Objects. There are well developed frameworks\textsuperscript{65} that enable direct communication between devices, such as peer-to-peer\textsuperscript{66} networks, however a peer-to-peer network makes the overall system more complicated and is difficult to use as a prototyping foundation for Smarter Objects. Therefore such a system is not used in this thesis; instead, the Smarter Objects components are all connected in a local network.
If the Smarter Objects server is exposed to the internet, a user can communicate with his objects from remote locations or combine objects in different locations. For example, a light switch in a vacation home could be coupled with the light switch in one’s home so that the home has a similar light pattern to prevent thieves from breaking in.

5.5. Server

The Smarter Objects server consists of two separate servers. One server handles the communication with all Smarter Objects and is written with Processing\textsuperscript{67} which is a simplified Java environment, and the second server is written with the C++ development framework OpenFrameworks\textsuperscript{68} and contains the Data-Flow logic and communication with the iPad. There are two reasons for this communication division. First, the Reality Editor and the server both need to be written using the same foundation, since they mirror the logic for the Data-Flow programming on each side. Since the Reality Editor itself can only be programmed with C++ and OpenFrameworks, the server needs to follow the same path.
Processing is a very simple and stable environment for interacting with hardware. It is used to communicating among all kinds of inputs and outputs via USB, Wifi or Xbee, Serial, OSC or JSON. In addition to the connections, the Processing server also holds the audio playback for a radio, which is realized with the minim library.

The processing server communicates with a very simple command structure over OSC with the OpenFrameworks server. The OpenFrameworks server only needs to hold the Data-Flow logic so that it can be cleanly programmed. The Data-Flow logic consists of a list of objects. Each object inherits a basic object that defines all in and outputs. If an output of an object is combined with an input of another object, the input of the second object references the value that represents the output of the first object. During each run cycle, the server first looks to see if there are new commands from the Reality Editor or changes that come from the processing server, then it calls each object on the list and runs it routines. While running these routines, the server sends its commands for changing behaviors to the processing server.

The processing server receives all commands from the OpenFrameworks server and changes its values accordingly, then it runs its main routine that defines how each Smarter Object behaves. Finally, at the end of each cycle, the processing server sends back the states of the Smarter Objects to the C++ server in case changes occurred.

### 5.6. Electronics

The digital electronics that are built into a Smarter Object in order to make this object smarter need to complete three tasks. First, it needs to be able to process inputs and outputs to interface with all the functionalities a Smarter Object provides. Second, it needs to be able to process these inputs and outputs in a way that it can send them via network connections. Looking at a Smarter Object from a manufacturing perspective, a
manufacturer can use all kinds of microprocessors and communication modules to make an object “smart” and enable it to communicate with the Smarter Object server. However, some companies might not have the advanced electronics expertise necessary to do this. Not only do the devices need to be tested for electromagnetic compatibility (EMC) but they also need to incorporate microprocessors and code that make the device compatible with a Smarter Object and the Reality Editor. Objects that could be used with the system are typically ordinary objects such as a light switch or a kitchen mixer, which normally work without any microprocessor capacity.

The question is what would make it easiest for such companies to offer compatibility with the system proposed in this thesis? A promising solution may be modules that a company could simply place into its devices. With a clearly specified interface, a company would not need to do anything more than connect the pins of the module with the pins of the input or output devices. In order to keep such manufacturers independent from a third party and allow the companies to sell their products at the same standard cost, it would be beneficial for these modules to be connected with the object after the object is purchased by the user. In such a case, a company would only need to connect its inputs and outputs to a standardized connector found on the device. A user would then be able to plug the module onto the standardized connector to make the object smarter.

Figure 26. Illustration, how a wifi and microprocessor module could be connected with an electronic device
In recent years, a new category of modules that can be used to support this concept has entered the market. SD cards for cameras have started to support wifi built into the cards. Photos that are shot and stored on these cards can be automatically sent to a remote computer. These cards have enough computation on board to process a variety of inputs and outputs. A new specification for SD cards called SDIO\textsuperscript{71} even specifies the use of SD cards for inputs and output data streams. With such a standard, a manufacturer of Smarter Objects only needs to build an SD card slot into its devices that can hold the SD card of a Reality Editor manufacturer. A manufacturer could also produce one compatible card that connects to the Reality Editor and the card can be placed into all of its objects, without the need for individual testing.

Apart from the SD card standard, the company electric-Imp\textsuperscript{72} created a small form factor card that has the shape of an SD card but uses the casing and the pins for their own set of inputs and outputs. This card also holds a wifi connection and a microprocessor. Similar to any of the other cards, this card does not allow full control for programming, as these cards need to be programmed with a proprietary online programming environment and language. Such an environment cannot be used for demonstrating the potential of Smarter Objects and the Reality Editor. However, this development shows that a modular form factor is beneficial for Smarter Objects and could lead to easy manufacturing of such products.

The solution that was found for prototyping wirelessly connected Smarter Objects is a combination of a wifi or Xbee module and an Arduino compatible microprocessor developer board. Arduino\textsuperscript{73} is a development environment for electronics that allows the programming of Atmel microprocessors\textsuperscript{74} with a simplified language built upon the C programming language.
Since there is a large developer community around the Arduino environment, there are many examples that help to connect the wireless module with the microprocessor specifically using the WiFlyHQ library\textsuperscript{75}. In order to use OSC (Open Sound Control) for communication between the prototypes and the server application, I used a modified WiFlyHQ version called ArdOSCforWiFlyHQ\textsuperscript{76} that has the OSC library included. On top of this foundation, the Smarter Objects could be prototyped.

When I was searching for the right hardware to prototype the Smarter Objects, I designed an Arduino compatible circuit board specific for the Smarter Objects prototyping process using the Atmega32U4 processor. This board has generic inputs and outputs so that it can operate actuators and has inputs that can be used for all sorts of purposes. But since there are many useful hardware options for developing Smarter Objects already available, as I have shown it in this chapter, developing a specific hardware circuit board is less relevant, so I decided to use standard Arduino boards. I connected these boards either directly to the inputs and outputs of my objects or combined them with an additional custom circuit board. I adopted the latter solution in cases where the electronics of the object became more complex.
5.7. Graphical User Interface for the Reality Editor

The Reality Editor is written with C++ and uses Open Frameworks (oF) as the framework for simplifying the code. OpenFrameworks has build in simplifications for OpenGL and all kinds of networking capabilities including OSC. For other simplifications, the OpenFrameworks community creates so called add ons that are simple wrappers for all kinds of C++ libraries.

As one of its main add-ons, the Reality Editor uses ofxQCAR, an oF add-on that enables the simple use of Qualcomm’s Vuforia Augmented Reality tracker. As a foundation for the Reality Editor, the add-on first places the video image from the iPad camera as a background onto the Reality Editor viewport. Then it changes the OpenGL projection matrix of the viewport to synchronize the projection matrix identically to the iPad camera perspective. It then analyzes the video image stream in order to find markers in the images. When a marker is located, the ofxQCAR add-on transforms the OpenGL model ViewMatrix

Figure 29. Schematic of the Reality Editor
accordingly to the analyzed position of the marker, so that the origin of the model ViewMatrix appears to be at the center of the marker seen in the video background. The Reality Editor can now place a virtual object directly mapped on to a physical object.

In order to place such a virtual object onto the physical object, I used an Image Rendering Utility for OpenGL\textsuperscript{78} to easily integrate png\textsuperscript{79} images into the OpenGL-scene. With these png images, I could create complex graphical user interfaces with a small geometry count, so that the iPad can always render the scenes in real time.

In order to interact with these virtual objects, the touch interaction performed in the x and y screen coordinates needed to be transformed into the present OpenGL coordinates. This task is performed by a class called Matrix Calculation Utility for OpenGL\textsuperscript{80}. Once the coordinates are transformed, an of function called ofInsidePoly()\textsuperscript{81} is used to determine if the touch collides with the object that should be interacted with. For example, a simple button can be placed on the marker. With the ofInsidePoly() function, the Reality Editor can determine if the finger touch lies over the button. From here, the Reality Editor can handle the touch interaction as if it were a normal screen coordinate interaction.

In order to animate the graphical interactions performed with Smarter Objects and the Reality Editor ofxTweener,\textsuperscript{82} another of add-on is used for animating the interactions. For example, if a button triggers a change in the interface, the interface can change with a discreet acceleration or deceleration. Interaction can feel elastic or stiff. OfxTweener enables such animations. Using ofxTweener, the Reality Editor interface looks more natural and not so digital.

The Reality Editor mirrors the data flow logic that was described in the Server section, which consists of a list of objects. Each object inherits a basic object that defines all in and outputs. During each run cycle, the Reality Editor first searches for new commands from the server, then it calls each object on the list and runs its routines. Based on the markers that are detected with Vuforia, the Reality Editor decides which of the objects are drawn. Each object holds all elements that represent the visual appearance and the interaction
with the graphical user interface. Once the data flow processing has ended, the visual result is rendered to the ViewPort.

If a user touches an output of the data flow buttons, the Reality Editor saves a reference to the object and its output value. To visualize this process, the ViewPort renders a line from the button to the position of the finger. Once the finger releases the touch action above the input of another object, the previously memorized output value gets sent to the server as a source reference for the input of the second object. The server has now saved this connection and treats the object list accordingly. Each processing cycle, the Reality Editor now draws a line between the input and output of the two objects that are connected.

By drawing such connections, a real object can transfer its functionalities to another object. For example, an old radio could have a very appealing look and feel, but the new sound system has better functionality and sound. With the Reality Editor, all the dials and buttons of the old radio can be used as outputs and mapped to the inputs of similar functions within the sound system.
6. Prototypes

6.1. Markers

In order to directly map virtual interfaces onto physical objects, I needed to find a method to combine the virtual with the real. Augmented Reality trackers have shown to be the most promising solution. Such trackers demand Augmented Reality Markers to combine the real with the virtual. An Augmented Reality Marker is a reference image that is known to tracking software analyzing video streams. This chapter discusses the chosen tracking software and the development of appealing markers.

There are many different Augmented Reality trackers such as AR-Toolkit\textsuperscript{83}, PTAM\textsuperscript{84} or ReacTIVision\textsuperscript{85} that are commercially or freely available and the technology such trackers use has common foundations. The software searches in a grayscale image for specific contrast rich points that can be recognized over time in a video stream. Such points are permanently compared with the points found in a reference image’s so-called marker.

The algorithms of an Augmented Reality tracker are optimized to find such specific patterns in any geometrical orientation as well as a variety of light conditions. Good trackers also provide error correction so that the tracking becomes very robust.

The Reality Editor employs the Vuforia tracker\textsuperscript{86}. Vuforia is a feature tracker. This means the tracker analyzes a captured image for high contrast edges. Each edge is therefore a feature.

Vuforia has a database of images that have been analyzed upfront and are known to the tracker in size and orientation. The tracker\textsuperscript{97} searches in the captured image for features that match the features saved in the feature database. This database is called a feature database.
dictionary. Vuforia has internal algorithms that let the tracker look for matching features in the captured image very fast, so that those features can be analyzed in real-time.

Since Vuforia is an Augmented Reality tracker, it can not only find matching features, but also analyze the 3D-orientation of the found marker. As a result, a virtual 3D-object can be placed on top of a video stream so that the 3D-object appears to be in the scene connected to the real object that has the matching features.

Features found in a reference image representing a marker are unique and Vuforia assigns a unique ID to each marker so that it can be accurately identified. A precise identification requires an adequate number of features on the tractable image, so that the unique identification has enough points for differentiation. In addition, the tracking stabilizes to reveal invisible points.

It is important for the features to be in a non-repeating pattern so that the tracker can determine the orientation of the trackable object, otherwise a 3D-object might be inaccurately projected onto the marker. If all of these guidelines are followed, then the Vuforia tracker can be used for robust matching of physical objects with virtual objects.

From a design perspective, we need to take a look at the specific use case the Smarter Objects demand. A Smarter Object might be a mass produced product. This means that all versions of the same product will be identically produced. Markers need to be able to become part of a production cycle. It is possible that a user owns more than one copy of the object. In such a case, the marker needs to be altered during production so that every product has a unique marker and gets registered correctly by the Reality Editor.
A tracker like Vuforia is specialized to track features found in images. This means that it is specialized for detailed images. It might not be desirable to have products with images or a lot of useless details, since this could disturb the product language and usability. Smarter Objects need to be able to be combined with many different product design guidelines. A marker needs to have a pattern that is appealing to the eye and can be woven into many different product designs.

Based on these problems, I have performed a couple of marker tests, material tests and explorations shown in Figure 9.

Figure 31. A set of marker explorations used to find a good artificially generated result.

The first tests (Fig. 31A) should help to determine how complex a marker must be to be compatible with the tracking framework. The first marker has a very simple structure with some smudged lines, so that the pattern is not repeating. However, the structure of this marker is too simple for being tracked as not enough features can be detected. The marker also has fuzzy edges that prevent a good detection. The second marker test (Fig. 31B)
attempted to maximize the number of edges. This test shows that rather than having too many undefined edges, it is more important to have them unevenly distributed over the picture with details in different areas. This test has too many repeating patterns and therefore the tracker cannot identify this marker.

The next test (Fig. 31C) alters the previously used marker with a random pattern. Since the patterns are still too evenly distributed, this marker does not offer better trackability. The fourth test (Fig. 31D) aims to simplify the marker by the use of a random honeycomb structure. This structure provides a lot of edges and a non-repeating pattern. This structure is also less disturbing to the eye and it could be easily generated artificially with randomizing algorithms. The marker has been generated in a way that many honeycombs form solid colored shapes. The structure can become more trackable when separating the individual honeycombs visually. Therefore, I combined the previous version with a random detailed image in the following test (Fig. 31E). The fifth test (Fig. 31F) shows a lot of features and represents an efficient marker to use with the Augmented Reality tracker. I designed this last pattern for a Smarter Object that explored extended functionality of a light-switch (Fig. 32).

This exploration shows valuable information about the patterns that can be used with the Vuforia tracker; however, they are not as appealing to the eye, since the honeycomb pattern does not match well with the design of the light switch.

After these experiments, I examined a marker’s core requirements and determined that triangles in different shapes and sizes would best suit an artificially formed shape that serves as a surface structure (Fig. 33). Triangles can be easily tracked and remain visual in the background, so it does not disturb the user from the functionality of the objects. Such markers exist on a radio that was built to illustrate this thesis (Fig. 34). The radio will be explained in more detail later on.
A radio’s speaker is typically hidden behind a fabric or a grid to prevent it from being damaged. I replaced the conventional protective structure with a triangle patterned marker so that this structure can serve two purposes at the same time. As previously noted, the structure must be random and non repeating. In the following exploration (Fig. 35), the marker has a stronger mix of big and small shapes and different patterns which makes the marker even more stable for tracking.
These patterns enhance the simple aesthetic design of a desk light (Fig. 36) while also providing a perfect marker for Smarter Objects. Since these markers are formed from simple geometrical shapes, it is easy to randomly generate them in software. Through laser engraving, printing, screen printing or stickers, they can become part of the mass production process of commercial products.
6.1.1. Natural Materials

Where else can we find appealing non-repeating structures that are interesting to look at and can serve the needs of the Vuforia tracker? In nature, we find unlimited examples of detailed non-repeating structures; structures that are interesting to the eye. It is valuable to adapt these structures for markers used in mass production.

The bark of a tree, for example (Fig. 37), has a lot of edges found in a topologically similar but geometrically different structure. The structure itself is a very specific identification for the individual tree. The two trees shown in Figure 37 are wonderful examples of strong Augmented Reality markers, as they have non-repeating patterns, but the patterns still allow us to identify them as different trees. This knowledge of visual identity is a very important exploration, as it allows us to identify different nature-inspired patterns to be used as markers in the mass production of Smarter Objects. The structure of the tree bark could become a central design identification of a product, to differentiate it from other products on the market, as well as indicate that such a product is in fact a Smarter Object.

In contrast, when we look at bark mulch (Fig. 38), such natural structures are very chaotic. But if we step closer, the mulch produces good details. This means that it might also be important to think about the different distance a marker is seen from and take care of it. If a house becomes a Smarter Object, then it can have bigger structures that can be seen from a
distance, but if we step closer to power outlets or water taps we might see another marker structure to identify these objects.

Stones are another structure that is well defined from both a distance and close up (Fig. 39). We can see many details on each stone that can be used for detecting features as well as shadows in between the stones that produce edges that are visible from a distance. Stones produce one of the best marker examples. They do not trigger a user’s attention or disturb it with too much chaos.

But how can these explorations be transferrable to Smarter Object production or even the material which a product itself will be formed from? To find some answers to this question I want to introduce some material explorations that have promising characteristics for designing Smarter Objects.
The first exploration looks at the structure of textiles. Fabrics have structures similar to those of tree bark. They have an overall identity defined by the way they have been knitted, but the structure itself becomes deformed by the way it is used or by the process of hand manufacturing or the structure of the yarn itself. This is of course the case the more loosely the fabric is knitted. Fabric can be combined with electronics, but can also become part of a production cycle for products of solid nature. For such a production, fabrics can be combined with resins forming a composite material (Fig. 40). They can also be combined with bio plastic materials that, in combination with the yarn fiber, generate a solid composite structure. Such composites can then become part of a solid product case or just a decoration element, but serve an important role in enabling an interaction with the graphical direct mapped interface.

Similar compound materials can also be formed from felt-like materials or natural fibers such as insulation wool (Fig. 41) or fiber boards consisting of natural fibers transformed into a solid state through the use of resins (Fig. 42).
Another promising construction material is Paper Mache as a compound with resins and plastics (Fig. 43). In such a combination, it forms extremely strong surfaces. It generates surfaces that can be made of many different colored paper sources. Smarter Objects formed from such a material are very sustainable and very lightweight, easy to recycle and they can be used for products that can decompose in nature.
Chipboards (Fig. 44) can be used to form Smarter Objects on a bigger scale, using colored resin and wood chips. If these materials are used for large scale constructions, they can help to augment construction work or buildings in general. Looking at this structure, the material is similar to marble or granite and looks very stone-like. One can think about compounds of stone chips with concrete as well.

Finally, I want to introduce a material called liquid wood\textsuperscript{89} (Fig. 45) into the Smarter Objects material exploration. This material can be used in traditional injection molding machines. It therefore can become part of any normal production cycle. The material is made from actual wood and generates naturally occurring patterns related to the fiber structure and the molding process. The material can have multiple colors and can be tweaked to form more contrast in its appearance. Objects formed from this material will be very environmentally friendly and trackable by the Reality Editor tracker.
Since research on the use of new materials for mass production is not the core of this thesis, I searched for some material that let me mimic the structure of a natural material. I found that paper produced from natural fibers (Fig. 46) have similar optical surface properties that can be used as the essential building material of a Smarter Object.

In our case, the paper has wood chips that have been combined into the paper structure. This paper can easily be glued on top of an existing product or a shape to mimic a material made of natural fibers. In the case of Figure 47, I was experimenting with a product line of kitchen devices. Products made of such a material can become an interesting "natural" product line.
Finally, I also investigated surfaces that are created in an artistic manner (Fig. 48). Sometimes this object can be a collector’s edition of something or represent a specific memory. In our case is it is a couch in the MIT Media Lab. This couch has been covered with graffiti. These shapes and text are another good example of a functioning marker that weaves perfectly into a real space without changing the general appearance of environments and objects.

There are many ways to weave an augmented reality marker into everyday objects. Design preferences determine how to combine such surfaces with the design of appealing objects. Such structures, when inspired by natural shapes, can leave a good memory with the objects and provide ample space for unique product design. Unlike other Augmented Reality markers such as the ARToolKit\textsuperscript{90} or even QR-Tags\textsuperscript{91}, the Augmented Reality markers discussed in this chapter are more natural and weave better into the appearance of everyday objects so that a Smarter Object becomes a real opportunity for enhancing mass products.
6.2. Radio and Speaker

The first Smarter Object that I prototyped was a radio. I chose the radio because when it was first invented, it was very simple. Over time it evolved to become very complex. Ultimately, with the advent of the digital age, the radio became a media player. Such a media player has a lot of functionality formed around the simple core experience a user needs: select the preferred music and change the volume.

To enable a fast prototyping process, I used clay to develop the physical interface. With this clay I wanted to explore in which way the objects can have a specific form that represents the new functionalities of a Smarter Object. Since the design of an object changes substantially once it has a virtual counterpart, this was a meaningful exploration.

Figure 49. Different shapes made from clay.
Aside from the use of a more traditional concept of a radio with a dial knob and a tuning knob, I also explored interfaces involving sliders that could be slid around the speaker for volume and tuning, and a shape that had sliders and rotation knobs in combination with the speaker. After a short evaluation of these shapes in clay, an organic shape with rotation knobs was selected for further development.

For fast prototyping of the working prototype shape, I used the software Cinema4D\textsuperscript{92} for polygon modeling of the radio and the software Pepakura Designer\textsuperscript{93} for unfolding of the polygon model into a planar cutting plan. This plan could then be prepared with Adobe Illustrator\textsuperscript{94} for cutting thin wood with a laser cutter. The cut-out shapes are glued together with hot glue to form the shape of the radio. With this prototyping method I can generate low polygon shapes in minimal time. I designed two iterations of the wooden radio. The first iteration gave me an idea of where to place the buttons and how to change the shape so that the buttons would hold nicely.

Figure 50. Lasercut parts for a radio
In order to interact with the radio in any digital possibility, I wanted to create rotation knobs that could rotate infinitely. After some research, I could not find a product that has such properties. The solution was to glue two rotary potentiometers together. Using an Arduino compatible teensy board, the two potentiometers were read with an analog digital converter (ADC). Since the potentiometer can measure a physical range that is greater than 180 degrees, I could define by software that if one is moved beyond 180 degrees, the other potentiometer starts to get used, so that I get a full 360 degree reading. The Arduino board sends the changes on the potentiometers to the server application via USB and a simple serial protocol.
I connected the radio to the Smarter Object Processing server previously described and programmed the server in such a way that the tuning knob will hold 8 tuning spot that can be chosen from when rotating the knob. The volume knob regulates the volume. If the maximum volume is reached, the volume is limited even while radio knob can still be rotated. In a practical test, this has no influence on the usability, since the audio is the feedback for the limitation. In addition, the loudspeaker of the radio was connected to an amplifier and the amplifier was connected to the Processing server as well, so that the music can play through the radio speaker.

![Smarter Object Radio Prototype](image)

Figure 53. Smarter Object Radio Prototype

Shunichi Kasahara helped me to program the first graphical interface for the Smarter Object radio. As the first prototype, this interface visualized the physical selection on the radio tuning knob and enabled the reprogramming of the radio spots within the reality editor. Therefore, we placed an mp3 selection above the radio loudspeaker. The mp3 selection is generated from a mp3 list that is sent from the server to the Reality Editor. A user can drag a song from this list and drop it on to one of the 8 tuning spots. Once released, the Reality Editor sends the change to the server and the server reprograms its audio selection.
This first prototype worked as expected. But I wanted to bring the Smarter Object concept closer towards the connecting of objects, so I built a speaker using the same methodology as I did it with the radio. The speaker contains an Augmented Reality marker similar to the radio marker but without physical control knobs. I connected the speaker to the radio amplifier as well, and placed the amplifier inside the speaker so that the full setup appears nice and clean. Luckily, the amplifier was a stereo module so that within the Smarter Object

Figure 54. A song gets dragged from the mp3 list on to the tuning knob.

Figure 55. Smarter Objects Speaker
server I could simply pan between the left channel and the stereo to reflect the interaction that should show how to connect two objects. Within the Reality Editor, my colleague programmed a line that can be digitally drawn from the radio to the speaker with the sweep of a finger. Once drawn, it remains connected between the two devices, and with a swipe the user can delete this connection. Each state of this line is sent to the server which uses the information to play sound on the left audio channel if the objects are connected, and stereo if they are not?

### 6.3. Overlapping Interaction

While interacting with the first prototype, there appeared to be a problem that I describe as overlapping interaction. Within the graphical interface, the list that represents the mp3 songs could be moved up and down with the finger. But to drag a song, a user needed to rest the finger on this list for a moment to activate a second functionality that allow the
user to drag and drop a song onto the tuning knob. After a few demonstrations, it became evident that this overlapping interaction is very unintuitive.

This fact can be explained with an example: usually, a physical object does not change its behavior when it is held for a longer period of time. A glass does not make water hot when it is held, nor does it turn colors. Every function in the physical world has exactly one way of working. For example, a car always has the function to drive. A bench always can be expected to serve as a seat. We expect that our surrounding has no functionality change as long as everything looks the same. When we look at similar behaviors in nature such changes in functions or appearances of an animal can only be found to trick other animals. Animals use this to lure their prey into a trap to eat them, since the prey is confused by the situation and misbehaves. These concepts are counterproductive for modeling intuitive interaction. I can use many different versions of a glass, but then I need to understand that I hold a new version of the glass in my hand. Translated into an interface, this means that every interface needs to have exactly one function or a clear interface that tells me how the function of the object has changed, or can be changed. This finding has resulted in later prototypes which include a mp3 list on the iPad screen that can be moved, as well as a button for each song that can be dragged and dropped, so that all interactions are visible and are not overlapping each other.

6.4. Memory Bar

When presenting the radio and speaker to users, one major question arose during these presentations: Can I take the interface with me or how can I operate a Smarter Object if I am not physically near it?

Since in many ways the Smarter Objects blur what is virtual and what is physical, computer adventure games like Lucas Arts’ Monkey Island provide a wonderful solution for the same questions. In an adventure game, a user must interact with a virtual environment and
objects in order to solve problems. These problems can manifest themselves in systems that are broken so that the user needs to solve a complex puzzle until the system works again or he needs to take objects from one place and combine them with object he finds in another place. In order to perform such tasks, adventure game designers developed an interactive visual clipboard in which a user can store objects by picking them up from the virtual environment with a pick up command so that they are stored in the visual clipboard or virtual backpack. If one of the collected items is to be used, the user can make the command “use,” then click on the item and place it on another item to make use of it.

This visual clipboard is an essential tool for adventure games that are essentially a purely virtual version of the Reality Editor. Those games allow the manipulation of virtual systems within a virtual world, whereas the Reality Editor is used to manipulate virtual systems that are in connection with the physical world.

Transforming and applying the concept of an adventure game visual clipboard into the Reality Editor means that it must be possible to store physical objects as pictures in a memory bar so that a user can easily recognize his stored objects. Once such pictures are stored, the user is able to simply click on the still images to open a full screen image that has the interfaces augmented on top of the still image.
Figure 58 shows a prototype of the Reality Editor with a memory bar. A user can rest his finger on the screen for three seconds and an image appears underneath the finger. This image can now be dragged and dropped onto the memory bar, where it gets stored together with all Matrix transformation data for the interface, and the interfaces that are needed for the scene until a new image overrides it.

In order to jump between the still images from the memory bar and the camera’s live stream, the Reality Editor prototype has two additional buttons that enable movement between a frozen and a live image. With this solution, a user can operate the Smarter Objects from a remote location.

Figure 58 also shows an updated graphical interface that I have designed and programmed. It reflects the previous findings as well the design guidelines found in the design chapter.
6.5. Lights

To prototype the interaction between many Smarter Objects, I decided to repurpose two desk lights to form additional Smarter Objects. With these prototypes I wanted to explore how difficult it would be to repurpose an ordinary electronic product into a Smarter Object. Furthermore, light and music generate a good contrast for demonstrating the diverse applications for Smarter Objects. Based on my marker explorations, I added a pattern to the lights which serves as a marker and looks similar to the pattern of the radio. I chose a similar pattern so that the Smarter Object appear as one system and so that the marker communicates the same Smarter Objects foundation with every object.
For making the lights Smarter, each desk light use a Philips hue\textsuperscript{97} light that can be controlled in color, hue and brightness wirelessly over the Smarter Objects network. The Hue uses the JSON\textsuperscript{98} protocol to enable networked communication with the lights. Based on this protocol I connected the Hue light to the Smarter Objects server. For this purpose I used and modified functions that have been written for the processing HueDisco project\textsuperscript{99}.

In order to connect the light switch of the desk light with the Smarter Objects server I detached the light switch from the 120V circuit that would ordinarily switch a light bulb on and off, and connected it to an Arduino board that is connected with the server application via a USB serial connection. Since the light bulbs no longer need to be switched on and off directly via switching the 120V, the socket for those bulbs is connected directly to the power plug so that the Philips hue is powered permanently. This way, the functions of the smarter light can be fully controlled by the Smarter Objects server.

Within the Reality Editor, I designed a user interface that let me change the color and saturation within a color-picker that is mapped onto the light. Since the color picker only
works in two dimensions, x and y, respective saturation and color, a third slider underneath the color-picker is used for controlling the brightness. A small rectangle beside the slider visualizes the color choice that is sent to the Smarter Object server so that the server can forward this changes to the light. With this prototype, one can use the light switch on the light to switch the light on and off, but the Reality Editor can be used for changing the color of the light.

6.6. Mental Object Model

While prototyping the connections that can be performed between many Smarter Objects a new problem appeared. Using the Vuforia tracker, the Reality Editor only worked when all objects are in sight. Once an object is out of sight, the reference to the object is lost. The more Smarter Objects I built, the more it became difficult to keep them always visible.

To solve this problem, the Reality Editor must store a memory of where the other object are placed in the room. With such a mental model, a user could connect objects that are too far away from each other to be seen in the same video image. For example, if one wants to connect a toaster to the house front door lock, so that he can make sure that all devices in one’s home are turned off once the house is locked, a user can touch the Smarter Object output tag on the door and then walk, while keeping the finger touched on the display, to the toaster and release the finger on the input tag representing the off button.
I prototyped a version of this mental mModel for the Reality Editor and James Hobin helped me to realize the code that is used in the Reality Editor. The mental model interaction in the Reality Editor follows this logic: If the object disappears from visibility, the object snaps to the edge of the screen in the form of a bar. The bar indicates the direction in which the Reality Editor calculates the estimated position of the object. This way all connections between objects can be visible to a user and the user can determine to which devices the visible device is connected by following the lines.

Since a rough functionality for position estimation is sufficient, this mental model functionality uses the gyroscope and the acceleration sensor of the iPad to perform a positioning estimation of the object. Unfortunately, the acceleration sensor of the iPad is very inaccurate in its resting position so that it is difficult to generate a full position estimation by only these two sensors. The working prototype of the mental model showed to be very reliable with the idea of the bars that snap to the edges once an object is not visible. The positioning in the space not visible to the iPad camera needs to be further researched. A beneficial system would be the use of an OpticalFlow where the motion performed with the device holding the Reality Editor gets estimated from the motion of a few camera pixels, so that the gyroscope sensor of the iPad can be supported by rough motion estimation.

Figure 62. Mental Model Live Demonstration
With the mental model, all objects are always present for the system and a user does not need to care if the objects are visible at the same time when interacting with more than one Smarter Object. With this foundation, the Data-Flow programming can be prototyped.

6.7. Data-Flow Prototype

As a first step for the Data-Flow prototype, I separated the programming from the interactions that are only performed with the functions of the object itself. This separation is represented with a new button selection that enables switching between the two modes.

By separating the Data-Flow programming, the interface became more clear and less crowded with interaction options. This helped to eliminate failed touch inputs that could interrupt a programmed connection. Instead of simply connecting one object with another object, the new prototype that explores the Data-Flow programming separates the radio and the speaker into all of its interaction components. In addition, these components obtain tags for inputs and outputs.

With such a separation into all components, the radio is represented with a tuning knob that has an output that is connected with the input of a volume knob. The output of the volume knob is connected with the input of the speaker. With this simple program, the tuning knob gets bridged through the volume knob to the speaker. The source that reaches...
the speaker represents the music source processed by the volume object component, so that the final signal that goes to the speaker represents the music regulated by the volume. This program can be extended to the speaker by connecting the output of the radio to the input of the speaker.

6.8. Improving the Data-Flow Prototype

After exploring the possibilities with the Data-Flow programming prototype, two new concepts arose that helped to simplify the interaction.

First, I realized that a user always starts connections between two objects with the output source that gets dragged over to the input of the second object. Additionally, the input and output tags appeared to not be very intuitive. Therefore I designed a simple version of the input and output tags to just one round button. By default, all tags are outputs. If a user
touches one of the tags all the other tags switch to inputs. Since the new tag is round it can hold a connecting line on the outline instead of the center point. The lines themselves became a color gradient from yellow being the output and blue being the input.

In addition to simplifying the inputs and outputs, a simple concept for logic appeared to be more intuitive than the original concept for placing a virtual objects in the virtual space next to the Smarter Objects. Instead, the tags themselves can hold the logic that is needed for programming Smarter Objects. Inspired by water barriers, the following example can explain the concept: Let’s consider 3 water barriers that are aligned in a row so that if all barriers are open the water can flow from 1 to 3.

If the first water barrier is opened so that water can flow through the barrier but a second one is closed, the water cannot reach the third barrier. The same happens if the second barrier is opened but the first is closed. In a computer program this would be translated as IF <condition> AND <condition> THEN.
For our example, this would mean that instead of creating a virtual IF object, we can just bridge the source through the object that would trigger the passing of our signal. If there is a source in the first place, and the condition of the second object is matched, then the third object would receive the source from the first object.

Beyond this simplification, we can also construct a second scenario. If we would connect the first barrier directly with the third and the second directly with the third as well, water would reach the third barrier if any of the first two barriers are open. In a computer program this would be translated into IF <condition> OR <condition> THEN. And finally one output source can simultaneously be connected to many input for multiplying the source.

With this simple concept of bridging and multiple connections, many Data-Flow programs can be prototyped without the need for additional virtual objects and therefore without unintuitive complexity.

Figure 67. Reality Editor final iteration
7. Use Case Scenarios

Now that I have described all necessary concepts related to Smarter Objects and the Reality Editor I want to introduce some use cases in the form of snapshots from an imaginary future existence in which Smarter Objects have become a normal part of everyday life in society.

7.1. Waking Up in My Home

It is 8:00 am in the morning. I have this nice old radio that I found at a flea market last weekend when I went for a walk. Using the Reality Editor, I have connected its volume with one of my calendars that I called “Hello World”. The Hello World calendar holds all my time settings for waking up in the morning. “Good Morning world”, I silently whisper as I open my eyes. Last week I placed a color sensor on my roof and connected it with a little toy my children almost threw away. They say they don’t use it anymore, it’s old. It is a little projector that shows the time and can alter the colors by rotating a wheel. I connected the red, green and blue values from the color sensor on the roof with the color setup of the toy and now I use it to show me the morning sunset in my room. The clock just told me that I can spend another 10 minutes in bed as there is still enough time for all necessary tasks before leaving the house.

I connected the brightness of the toy to my calendar as well. I also repurposed the wheel that was used for altering the color to control my window blinds. When I finally touch it to wake up, it always gives me a satisfying feeling as I have nice memories from seeing my children play with it. The window blinds open smoothly the more I rotate the wheel. I have connected the heating system of my car to this wheel as well. Before I had a Smarter Car I
had an automation system that always used up so much energy. Now the car knows when I start my day. I also connected my coffee machine to it. I like that the coffee machine reacts to this action, because it will be made fresh the moment I arrive in the kitchen, even when I wake up at different times each morning.

Other things in my home start to operate as well. I have always been disturbed by the washing machine’s spin-drier program at the end of the washing cycles. I used to wake up a couple of times because of the noise. To resolve this problem, I connected the spin-drier program with another calendar named “Sleep Time” and also connected the calendar with the wheel on my time projecting toy. This little program indicates if it is “sleep time” and the wheel closes the window blinds. The spin-drier program no longer disturbs me when I am trying to sleep.

When I wake up and walk in to the bathroom, I alway walk on nice warm floor tiles. It used to be very expensive to heat all the floors in my house, but now I can use the Reality Editor to perfectly select which tiles on the floor are actually on my path each morning. That was not clear to me when I first moved into the house, but over time I always walk the same paths and I love to make small selections and find the right floor heating preferences. I can carefully and individually select which floor tile should have what kind of temperature at which time. I can also install nicely designed temperature regulators, but I add a Smarter Objects module to it so that I can operate it from distance or connect it to some sensors.

We have a very open house and the heat circulates a lot in our home. We often have disagreements when my children adjust the heat. We always regulated the heat using different settings and it was difficult to find an mutually agreeable household temperature. Luckily we came up with a wonderful solution. We took an old portable radio apart so that we had the dial left and build a small wooden cat around it. We bought another virtual object called “am I home?”, placed it next to the wooden cat and connected all our GPS phones to it. We set the “am I home” distance to about 1 kilometer and finally connected the “am I home?” object to the dial of the wooden cat. We then connected the dial to the heating system and we disconnected the dial on the heating system and all virtual
connections that would enable direct control of the heat. Once we made all of those settings, my wife and I virtually locked the heating system with a finger written signature on the touchscreen of my phone that runs the Reality Editor. Now whenever we are all out of the house, the heating system stops, and if one person comes home it starts again even before opening the door. If someone feels like it is too cold in the home, now we always communicate with each other and search for the wooden heat cat. Since we started using the wooden heat cat we have much less discussion about the standard temperature in the home and whenever someone feels like the temperature is unsatisfactory, we go for a wooden heat cat hunt and then talk about if we can change it. The other day, I placed a Lego Mindstorms temperature sensor onto the wooden heat cat and connected it with the dial as well. Now we just need to place the wooden heat cat into a room and the sensor measures the temperature and adjusts the home accordingly to the perfect temperature in that room. I know our heating system is a mess, but we found a very good solution thanks to Smarter Objects.

I am alone in the house right now, so I took the wooden heat cat with me into my bedroom. In the evening we usually place it in the living room, where we found to be the best spot for everyone. I go to the bathroom, take a shower, brush my teeth and find out the temperature outside. I connect the temperature of my shower to an outside temperature sensor. A virtual delay object spends twenty seconds to make sure that the shower has enough time to heat up. Then the shower tells me the outside temperature, and after another twenty additional seconds it stops again. With the memory in mind that it is Fall and the feeling on my skin that the temperature outside must be very cold, I no longer want this program to run. I get out of the shower, take my phone and disconnect the temperature setting and any connection of the shower to anything else using the Reality Editor. I disconnect all connections except one.

This one is actually useful. I installed the Smarter Grid power outlets in my home and with my Reality Editor I can program them to stop working once the electricity costs have reached a certain monthly limit. Since I am using a Smart Grid, the prices are changing by the minute. If I use the cheap cycles I save a lot of money. So I connected the water heater of
my shower to the power outlet switch. I don’t need to connect a physical plug; a virtual connection is enough. It saved us a lot in the last months. But other than that the shower should not be controlled by anything. I like it as it is now that it is Fall.

Now that I am dressed, I go in to the kitchen, drink my freshly brewed coffee, put the washed cloth into the dryer that has finished its drying cycle since the spin-drier program started when I woke up. I put on a warm jacket as I know that it is cold outside and get into my nice warm car that already started the engine once I approached it. As I connect my car with another “am I home?” virtual object and set the distance to ten meters, the car starts automatically and unlocks, so that I just have to start driving away from my home to start saving energy.

7.2. Sunday Morning Lawn Care

On a warm Sunday afternoon I just came home from a nice breakfast meeting with my friends. I walk from my car to the door and check with my phone on the lock to ensure that my virtual social network group object is connected to the door. I have invited friends over to watch the big finals game on our TV this evening. In the past, it was a nuisance to open the door for every guest while the game was running. Today, I just invite them to my personal social network group and since I connected the virtual social group object with the door lock, each one of my friends is granted access once they are close to the door. They just need to have their phone with them.

Before the game starts, I want to get my garden in shape. First, I want to make sure that later when everyone comes nobody gets wet from the lawn sprinklers. I have a Smarter lawn sprinkler that I not only connected with a couple of moisture sensors distributed throughout the lawn so that the water only sprinkles if the lawn gets dry, but I can also point my mobile device at them and get a virtual radius showing where the actual water will fall. I selected this specific feature because I want to keep the walkways dry. All I need
to do is draw the outline of the area in which the water should fall down to the lawn in the Reality Editor. I can do a similar process with the lights. I can point my Reality Editor on the garden lights during the day, and I see the radius light up at night. Last summer I used this feature to distribute the lights perfectly in the garden, now I just use it to generate a nice dramaturgy for the evening. I start with a little brightness of the lights at the street and as you approach my house, the lights get brighter. I just need to draw a bigger radius in the Editor and it will light up a bigger space. Since I have a color sensor mounted on my roof, I can connect one of the colors with a virtual inverter object to the brightness input of my garden lights, so they start to turn on once the sun gets dark.

As a last step I just want to enjoy mowing my lawn. I open my garage and the lights in the garage turn on, since I connected the Smarter Lights with the d garage doors. I also connected a motion sensor to it that I bought the other day. I needed to find the right spot for this sensor as I always worked on my hobbies behind my van and the sensor could not detect me from where I placed it the first time. Since it is a Smarter Object it is quick and easy to place it in another location. I also use this sensor to turn on the radio in the garage. I connected it with a little dial that I placed on my work bench that regulates a delay. So the radio does not always starts when I use the car, but I can regulate the delay when I work in the garage for an extended period of time.

Since I am in the garage to take out the lawnmower, I use my Reality Editor to see if there is enough gas in the mower. The Editor also shows me the average gas use of the last four uses. This is a nice tool. My garden needs around five liters for a full session. Only four liters are in the tank, so I fill it a bit more. I also check to see if all the patterns for the lawn plates are still fine. Last time I used the English Garden settings but the lawn turned out to be too flat. I use the Reality Editor to adjust these settings with a nice visual representation of the plates, but since I have all the standard connections of the lawn mower set, I can use the mower’s settings dial to easily switch between the programmed settings when I operate it. This settings will now hold for a couple of months.
As I take out the lawnmower I see that my tool shelf has a loose screw. So I take the screwdriver and since the screw is already loose in its hole and I don’t know how much force I can bring on it, I hold the Reality Editor on the screwdriver and request a special setting. I set it to wood and screws but with the special tag loose screw. The screwdriver is now set to the perfect torque and my shelf is safe. I like this little virtual object that helps me with the torque. The screwdriver manufacturer did not consider such features but manufactured this wonderful Smarter Object screwdriver. Some people in an open online community board just programmed this virtual object so that I can combine it with my screwdriver. The whole community helped to collect all the right torque settings so that I have a prepared choice for almost all situations. From week to week as the community helps to define this object more and more, the virtual object gets better and better. This makes me love my screwdriver more and more.

I also connected the lawnmower with the sprinkler system, so that if the lawnmower turns on, the sprinkler stops working. I can just go out on the lawn and don’t need to worry about anything else other than if the English Cut works perfectly.

### 7.3. Watching the Game

The evening has arrived and the game is about to start. Three of my friends have arrived but five more are on the way. Since they have automatic access to my home for the evening, I don’t have to worry about them being here on time so that I have to open the door while the game could have an important moment. But there is something else I still want to check. I have a bunch of external hard drives and I want to find the one where I save sports videos. I have three of them and as I recall only one holds enough free space for today’s game.

I can use the Reality Editor to point on these unpowered drives and the Editor gives me a status about how much space is left on the drives. I can swipe through a couple of
visualizations and one actually shows me the capacity in hours of recorded video. Whenever the hard drive is used, a cloud service keeps track of the capacity. If I un-power it, I can use the Reality Editor to read the capacity right on the drive.

The same also applies with SD-Cards. I have many vacation SD-Cards. I like to keep them like photo books or old film rolls. Each SD-Card holds photos of a vacation or an event. I like to search through them, and the Reality Editor shows me the content projected on to the card. It is wonderful to look at all my computer artifacts in this way. Everything tells me a story about what is stored on it. While searching for this one hard disk, I found an SD-Card from my last vacation with my family in the Alps. I can use the Reality Editor to quickly browse through some thumbnails. I take a pen and finally write “vacation in the Alps” on it. Next time I don’t even need the Reality Editor to find these memories when I search for them.

But now let’s get back to the game. I have this very simple remote with five buttons. For the game, I used the Reality Editor to program Smarter Remote buttons to the four live views that my TV provides me. The fifth button is just connected to mute. This is all I need for the game. I have connected my audio system to the TV using the Reality Editor and since the kitchen is far away, I also connected the radio in the kitchen with the audio output of the TV. Nobody ever wanted to go to the kitchen and get the food for fear of missing a critical play and I don’t want to have a house full of screens. But now at least we have an audio link to the kitchen, so hungry viewers can stay in touch with the game.

My TV represents everything a nice medium should be. Just a screen, nothing more. In earlier days, the TV had all these outputs and unloved buttons that were required for nothing other than to program it. Now I can use the Reality Editor to program the TV and I can connect the screen to objects around me. For example, a few years ago I got this wonderful amplifier. I love its design. It has wonderful dials that give me the right feeling when I crank the volume up. No TV remote ever could give me this feeling. So I just connected this dial to my TV volume as well. Even when I have the amplification function
switched off, I can still use its dials to control other things like my TV. I can modify my entertainment system however I want.

Two months ago we played a car racing game with our game console and just for fun everyone searched the home for objects that could function as a game controller. I used my favorite amplifier dial, but someone else came with the kitchen blender, another one used the analog button of my screwdriver. We all connected these functions with the controller input of the console using the Reality Editor. One of my children even brought one of his walkie talkies and connected a virtual object that was called “give me the audio pitch” with the walkie talkies and then this object with the console. We had so much fun. When I remember how the kids made all these “newwwwaaaaaoooooowwww” sounds to control the car, it just gives me a smile.

I also had fun with my son the other day when he received a new electric guitar for his birthday. We tried the same idea again since it worked so nicely. We connected every object with a unique virtual audio effect generator object and daisy changed them all so that the guitar signal runs through all these virtual objects and then into the amplifier. He had the maximum expression with all his moves operating the physical objects. In the end, the whole family was engaged to wind screwdrivers, push light buttons or twist all sorts of kitchen devices, since those objects were connected to the guitar altering the sound. We all had a part and everyone used the Reality Editor to try to creatively patch the whole system in new ways, or to use one of the kitchen tools as a instrument itself.

7.4. My Surgery in the Hospital

I broke my arm. I was hiking in the woods and slipped off a stone when I wanted to cross a river. Luckily it happened at the beginning of the hike and I could get back to the hotel quickly. On my way, I called the hotel and the reception called the local ambulance. The ambulance came just in time and we headed to the hospital.
I was surprised to see how much the Reality Editor and Smarter Objects have transformed the hospital environment. From old movies, I always saw tons of wires connected to the patient’s care station. Every nurse and every doctor always asked for the name and birth of the patient to avoid making procedural errors in treatment. But some accidents happened because of weakly designed technologies. I read in Wikipedia that such accidents were quite common before hospitals and doctors started using Smarter Objects.

The moment I got into the ambulance, they taped a temporary tattoo on my arm and on my chest. I asked them where all the wires are, the ones that I am used to seeing in old movies. With a laugh the doctor said “We are not in the Stone Age. Today we don’t have these anymore. Everything is wireless and personalized.” He showed me how he uses the Reality Editor. First he transferred the identity from my citizen identification card onto the tattoos on my chest and arm. Then he used the Reality Editor to define the procedures that need to be done with my broken arm. He made virtual marks right on my arm and locked them with his personal finger written signature. He said, from now on, no failure can happen anymore. If someone would accidentally start the wrong surgery, the machines will stop working even before beginning. Only this arm can be treated, since the surgery tools automatically read your individual tattoo signature and only procedures related to a broken arm can be executed. Of course, if the vital signs change or a override of the signature happens by two other doctors, then this can be changed.

He then used the Reality Editor to connect the tattoos with a display in the ambulance so that they show my vital signs, but he also connected them to a special Smarter jacket he wears. “It is a jacket that will tell me, through physical vibrations, if some of my signs are out of range, he said. When I wear this jacket I will never lose my attention for your critical data.”

I asked why he is not using a head mounted displays for all of these procedures, and he answered that these things are very difficult to use in stressful group work situations. “We had a lot of complications in the hospital with vital signs that had been inaccurately
communicated,” he explained. “Nobody knew what was going on. It is much better to have a screen for this, so everyone can see your vital signs. I can point at them and communicate the next treatments better. But you will see one of these devices tomorrow when the doctor comes by your bed. Since we use the Smarter Objects technology, the doctor never needs to take paperwork with him or check your heart rate, blood pressure or temperature. With his head mounted display all he needs to do is look at your tattoo and all of this information will be presented not only at the moment, but over time as a report. For you as a patient, it is very convenient because you can sleep without concern about ripping of wires. You can walk around and take a shower. Your vital signs are always recorded.”

We arrived in the hospital and the doctor told me that his assistants will bring me to the surgery room that already is prepared. I asked him if he must detach the virtual patch to the tools in the ambulance, but he said no, there are two kinds of links in the medical sector. They can be preprogrammed. The ambulance has only short range links. This means if your wireless signal gets out of reach, the connection will automatically detach.

The hospital is very clean and organized. There are no wires, no tool that need to be touched by many hands. As a result, germ have a hard time to spread. There are no door handles that needs to be opened. Smarter Objects have been used to customize every aspect of the hospital daily routines. I asked one of the nurses, how it can be so clean and organized when they have all of these patients and surgeries. She said that the hospital gets better and better every day. “It is like constructing with Legos,” she explained,”If there is, for example, a door that needs to be opened automatically before the patient arrives, we can take the motion sensor from this door and move it into the room. We can now just move through the floors without any restrictions and we make adjustments to optimize our routines every day.”

“For your arm for example, we just prepared one of our surgery robots. It will look for the tattoo on your arm once we have connected the arm with the robot using the Reality Editor. If you go to the X-Ray machine, the machine will scan your tattoo and from there the system has all your data to start. Of course we have to connect you manually to the right
machines. With Smarter Objects and the Reality Editor, we can easily distribute the workload throughout the hospital. For example, there are two other doctors that have patched into the robot, so they can supervise the procedure with their head mounted displays, or whatever else they prefer to use. They will also stay connected with you for the whole procedure and check up tomorrow when you are done."

The procedure was quite impressive. Robots and humans worked hand in hand as I had a local anesthetic. I fell asleep in my hospital room with the good feeling that staff are somehow connected to me and I do not even have to call someone for help is something is out of standard levels.

The next day, a doctor came by to checks my arm. The tattoo that is underneath the gypsum that keeps my arm in shape is now connected with the gypsum itself. One of the doctors has linked the two objects before the procedure. As the doctor looks at my arm wearing his head mounted display, I know now that he can view all my vital signs starting from the time when I entered the ambulance. He sees all the parameters of the surgery and if my arm is in the correct parameters. He explained to me that I can use my Reality Editor now to view these vital signs as well. He then said that I can go home and that I should see him or another doctor to take off the gypsum in 6 weeks. I can see this also with a timer that is automatically part of the gypsum virtual object. As I hold my Reality Editor on my arm, I can see all of this information. The timer is a big green number counter. The doctor said if the day comes close to the time when I should contact a doctor to take off the gypsum, the counting down number will turn red. I then just click on the number to choose to connect with him or connect to a local doctor that can help me.

Once I returned home after my vacation, I needed to tell all of these adventures to everyone. But another thing was equally fascinating to me. Since the gypsum is a real Smarter Object, I can connect every aspect of it to all other Smarter Objects. I can use my vital signs to regulate the house light colors. For example, the other day I used it for my meditation session to become even more relaxed by controlling the brightness with my heart beat.
My wife was very concerned, so I let here make a memory shot of my gypsum so that she can see all the vital signs of my arm anytime. I connected the time with the vibration motor in my phone. I also bridged the function that will make the counter appear red to the red color of the little toy on my bed that usually shows the sunset colors. Now I get reminded in the morning once I need to take action with the gypsum. It’s good that I can use Smarter Objects to let people stay connected to my health and to remind me if an important event comes up.

Now I can understand the benefits of Smarter Objects in the medical sector. If I had a pacemaker, I can just have brief check ins with the doctor or give my whole family access to my vital signs. I can even let the ambulance automatically be called when something is wrong. I can connect the battery of my pacemaker to all sorts of objects in my home that would alert me if something malfunctions. I could connect all kitchen appliances with my pacemaker battery, so that they stop working and I need to call someone to fix it because I am hungry. I then would be reminded about this battery even if I had forgotten why these devices do not work in the first place. But I know that everyone else would remind me.

If I had an artificial leg I could use the Reality Editor to easily tweak the settings of the limb directly on the object and on the go. It would be very easy to make adjustments because all the interfaces are projected directly onto the leg using the Reality Editor. I could also localize the program settings of such legs so that once I am at, for example, the sports field they act a bit different as if I would climb up the stairs to the second floor in my house. I could connect the legs with other devices to be able to recognize things through vibration or I could even use the artificial leg to control a computer game. I could connect the battery of such legs to the light switches in a house to make the lights to an ambient display that shows me when the batteries of the legs need to be charged again. I would never forget it in the evening since the house would be dark.
8. Evaluation and Discussion

The creation of physical objects that are combined with virtual objects raises many questions which were answered or explored in this thesis. The following chapter examined these questions.

8.1. Critique and Evaluation by Professional Designers

To understand if I found good answers to these questions, I conducted two evaluation sessions with designers from IDEO. The first session which was attended by Michelle Sinclair and Todd Vanderlin allowed me to introduce the designers to my work. The purpose of the second session was to discuss the thesis in more detail. Michelle Sinclair and Todd Vanderlin attended the session again, joined by Colin Raney.

In the first session, when I showed Michelle and Todd the benefits for design as I see them, Michelle gave this initial response: “I really like the idea of outsourcing some of the things that you don’t use that often.” She also liked the concept of remapping the possibilities of what buttons can do. But later on, we also spoke about the possibility that some users might prefer an object with all functionality visible. From my perspective it seemed like a cultural issue. For some, the visibility of features is important, for others solely the functionality. Showing off the latest new technology to friends might play a role in user’s preferences, and since an augmented interface has not been introduced this way, it might remain a question to be explored with actual products on the market.

Another quote from Michelle was: “Right now I buy objects and from that point onwards, they decrease in value. They get old and then they die. But now with Smarter Objects, I
could buy an object and it gets smarter over time as opposed to decreasing in value and utility over time."

A more critical point was the idea of exposing markers to the user. The first question that Michelle asked was if her whole house would become filled with triangular shapes. Todd mentioned that at the moment these particular markers might be necessary, but in the future it could be another technology, like invisible markers. And this assumption is probably right. Looking back at the history of Augmented Reality markers, this technology already developed very far and in the future we will see better algorithms that would make the marker itself obsolete, or at least the surface will not need to hold such a top layer structure. But once such objects become ubiquitous or if you could just spray the marker on objects, you could have almost another sense for your environment, Michelle noted.

From a design perspective, Michelle liked the idea that I am not expecting every product designer or company to learn how to program for Smarter Objects or to learn how to integrate the electronics into their products. I explained the idea of modular SD-Cards that can operate Smarter Objects services through a separate company. Outsourcing the hard work to another entity is fascinating in Michelle's view. This supports one of the assumptions for this thesis that it is important not to overload a manufacturer or a designer with technology.

Todd found it a bit difficult to operate the programming mode with two hands. He asked if there is a possibility to make a snapshot to take the interface with him. Since I have been asked this question many times before, I showed him the memory bar that enables a snapshot of an object. The memory bar served his wish nicely.

But from the first session the question remained: when will be the critical point that the technology allows even more freedom to track objects of any shape and not only those with visible markers.
During the second session, Colin mentioned that he sees two bigger ideas in the thesis. One idea is the virtual reality editing and the second is the one of eliminating 99 apps for all internet connected objects. But for such a universal tool, there needs to be a universal dictionary in which a user can publish and find the virtual interface that matches its object.

For Colin, the fact that on the one hand Smarter Objects should become ubiquitous and on the other hand they demand a very specific pattern is in conflict. If there would be a couple of different methods for interacting with Smarter Objects, so that the designer could choose other options besides markers, then this ubiquity would be more likely possible. This comment draws back to the marker related discussion from the first session, but also points to another assumption that I made in the first place. I asked about the possibility to manufacture Smarter Objects, since this is the direction where this should point to. First Michelle answered that from a branding perspective, if everything has the specific look, a customer could identify a Smart Object based on this appearance. She thinks that this has something beautiful to it.

Todd thinks that there would be a solution for manufacturing Smarter Objects if there is a toolbox for designers that has a set of 50 different marker patterns, or that helps designers to design such a pattern or surface and as you design this surface the toolbox gives you a confidence level of how trackable your surface is and then you could upload your pattern. At this point, it would be too abstract for a designer to use such technology. If you can create something that makes it easy for a designer to use Smarter Objects as a part of their designing methods it would work. As Todd noted, “Give a designer a tool like Photoshop, Indesign or Rhino that helps them to understand that better.” This last comment was a wonderful affirmation that the concept of Smarter Object as I have formed it is on the right track.

There are other important questions that came up during this evaluation or discussions while forming this thesis that demanded some strong answers. I want to discuss some of those questions in more detail with the following subsection of this chapter.
8.2. Communicating virtual connections

One of the biggest questions I want to address is how a user can see the virtual connections that have been established between physical objects and their functions. One assumption is that once there is a certain level of complexity of connections in one’s home, a user cannot keep track of what the programs in the home actually do. Furthermore, what if there are many family members in a home and someone else programmed the objects? How can they all keep track of the Smarter Objects functionalities in the home? I want to explore this question with a hardware exploration and a theoretical example.

As a hardware exploration, I investigated the best intersection of the physical and the virtual. A touch sensor is such an example. It is an action-triggering button, but has no tangible feedback other than the feeling of touching the surface of an object. With this in mind I created a prototype that adds to every sensor, switch, knob and actuator a touch sensor onto the object surface. This touch sensor is also paired with an LED to provide visual feedback on each sensor. This visual feedback now can be used to visualize the connections made between Smarter Objects. To trigger the visualization of such a connection, a user touches the touch sensor and all connections with the sensor start to blink. This way a user can see where there are connections between objects in his home as long as they are in the same room and visible to the user.

Figure 68. The electronics and shape of the Sensor-Box.
I built a sensor box (Fig. 68) that holds a light, a temperature sensor and three touch sensors each paired with an LED. The box itself holds a microprocessor and a battery, including the charging circuit.

As a counterpart to the sensor box, I also repurposed a kitchen blender following the same principle. Since the mixer drains a lot of watts that need to be seamlessly controllable, I paired the mixer with a stage light DMX controller that can seamlessly control the energy of 1000 W by 120 Volts. DMX is a standard protocol for light equipment. It builds upon the RS485 standard for eliminating electromagnetic errors in data transmissions. The mixer has five buttons that were repurposed to be freely programmable. The buttons are each grouped with a touch sensor and an LED light to show the network connections. The system is connected with a wifi module and an Arduino Compatible development board.

While building this additional interaction capability, another question came up: Does this concept actually improve the user experience? Does such a system help a user become aware of the virtual connections made in the home?

First of all, this implementation builds up complexity and require a lot of predefined surfaces for the design of Smarter Objects. By doing so, this concept contradicts the idea of Smarter Objects actually providing more freedom vis-a-vis designing interaction.

Second, a straightforward connection between two objects is the simplest form of connected Smarter Objects. But such systems can be more complex if we consider the free programmability of all system aspects. For example, the music of a radio could be bridged through a switch so that the switch interrupts the connection when it is turned off. This simple program would make it much more complicated to understand just from blinking LEDs how the system will actually behave. If such a system becomes more complex and different functionalities can be represented with it, limitations of such a one pixel representation would be reached quickly. This is where the theoretical example can help to answer these questions.
Is it necessary to have a perfect system that always shows how it is connected so that whenever someone makes a change to the system, every other user will see this change? There are many things that we use every day that hide the full behavior of their systems. For example a light switch is not directly connected with a light, but we know which light switch turns the light on and off. Or all of the systems that work together to make a car work. The light switch on the wheel turns on the front lights. The break that make the car stop. Systems that do not tell us in the first place how they work, but we learn how to use them.

Of course, if we would be able to change the way the car works with a system like Smarter Objects, anyone else who uses this car should know what changes I made. Most of the time, I would make such a change to improve a system to my desire that previously malfunctioned. Most of the time, an object like a car is used by a set of people that are trusted by the owner of the car. Since the owner of the car wants his car to be functioning in the future as well, after he gives the keys to someone else, he will explain carefully to this person how the car actually works. If he made changes to the settings of the car, he will first inform every person that has access to the car about the changes he made. He will then ensure that the changes are understood so that the driver and the car are safe.

Another solution could be to personalize the Smarter Objects by the user. The wireless signals from a smart phone can be used to determine the distance between the User and the Smarter Object. This way each user can program Smarter Objects to their desire or the Smarter Object can inform a user about the changes another user has made. For example, the car could prevent the driver from driving and provide a message on the driver's smart phone. After accepting the changes, the driver could start driving.

Systems like a car should always be designed in such a way that human error cannot appear. A user can of course forget to tell everyone about his changes or another user falls back into previous internalized habits. Smarter Objects therefore should never be used for remapping important interactions, such as the breaks in a car, the functions of the steering wheel or emergency lights. However Smarter Objects can be used for making adjustments
to such important interactions. The steering wheel could be adjusted to different driving settings. It would be the role of the designer to design the interactions with Smarter Objects so that the objects are safe to use.

The same model can be used for the lights in one’s home. A traditional light switch on the wall is detached from the light that is on the ceiling. More than one such traditional switch could control this one light, and these switches could be located on different floors or in many positions close to doors leading into the same room. The people living in the space communicate between each other how everything in the home is set up and built. If one would make a change to the light switches using the Reality Editor, the person would communicate these changes within the group of people so that everyone is aware of them.

This practice can also be used for the wiring of a home entertainment system. If objects are shared, the functionality of such objects are always shared as well. Most of our electronic objects already have hidden systems that can be altered to generate a desired functionality (e.g. preset stations on a radio). These alterations are already communicated within the social structure so that a virtual change to an object could be communicated just like any other communication about present objects. In fact, the Smarter Object, through the use of its virtual direct mapped interface, offers a much better resource for such communication, as everyone who has access to the Smarter Object can use the Reality Editor to easily understand the object’s behavior.

8.3. Connected Complexity

Another question that came up during the IDEO evaluation relates to connected complexity. If I have 20 different Smarter Objects and I interconnect them all, I can quickly produce a complex system that can generate many errors and mis-­‐operations. This problem can be addressed with a simple example. When I connect an electric guitar with an amplifier, I generate the connection of two objects. But this connection does not satisfy my idea of the
sound I want. I need two additional effects to alter the sound to my preferences. So I interrupt the connection between the two objects and connect the guitar to the first effect box, and then the output of this effect box to the second effect box. Finally, I connect the output of the second effect box to the amplifier to reach my goal. I make this modification even when I have additional choices for connecting more effects into the effect boxes or to the amplifier. But first it would not reach my expectations, and second I do not have more physical objects. The last point might sound trivial but defines an interesting point that separates Smarter Objects from virtual objects. One can generate infinite instances of virtual objects, so that the complexity of a system could also grow infinitely. Smarter Objects need to be bought and therefore a user obtains them for a specific reason. In the case of the given example, a user would only have the four described Smarter Objects that can be connected. The user’s 20 Smarter Objects are all purchased for a very specific reason or they get repurposed for a specific reason because an object has value to the user. Therefore the value of only one Smarter Object entity limits the complexity of connections.

8.4. Physical Gestures

Another issue that arose is that most of the interactions are performed better on a virtual interface. A radio, for example, has a better virtual representation and the music can be controlled from everywhere. I want to discuss this question with the affordances of objects and the general research exploration for gestural controls.

Before the idea of gestural control, there was the computer mouse, the keyboard, joysticks and all kinds of other physical interfaces. Then touch interfaces were invented, and with the touch interface came the possibility to form all kinds of gestures on the touch surface. In the past years many free space gesture interfaces have been introduced such as the Kinect or the leap motion. In general, it looks like gestures are a beneficial feature to. Why is this important for answering the first question?
I would like to look at another gestural interface and compare it with the Kinect. Let’s take the Nintendo Wii\textsuperscript{108} and its free space gesture controller. The motion of this controller can be analyzed in space, so that a user can perform all kinds of gestures but needs to hold this controller. The controller has add ons that allow the controller to change its shape. For example, it can be changed into a steering wheel or a gun. The add ons have no identification themselves, they only provide the user with the affordances of a wheel or a gun. With such an add on, a user can take the wheel in his hand and knows instinctively what to do with this object, so that he can operate a racing game without any additional learning of a gesture. The user performs the gesture of a steering wheel automatically as the wheel tells him how to us it. The same applies for the gun. Without significant learning, the user can perform the gestures needed to operate a gun. Performing the same gestures with the Kinect, a user first needs to learn all the gestures necessary for operating a shooter or a racing game. The affordances of physical objects tell us their use and explain gestures that need to be performed with these objects. This means that Smarter Objects are actually digital gesture inputs that allow the interaction with digital information in a limitless amount of gestures. Gestures one does not need to memorize, since they are memorized in the physical space. Therefore Smarter Objects can be the logical consequence of the present development of gesture controls. Of course a radio can be operated perfectly with a touch screen. But the gestures to operate the interface for the radio need to be memorized in once mind. With a physical radio, one can use the externalized memory in the physical radio affordances as well the muscle memory in one’s body to operate the radio in a more intuitive way.

\textbf{Figure 69. Wheel Add-On for the Nintendo Wii Controller}
8.5. Markers in Everyday Design

Do I really want to have objects around me that carry a marker on top? We are used to clean shapes and clean surfaces. Is it optimistic to think that any object could have a marker on top?

This question arose in the IDEO evaluation and I want to discuss it using two different perspectives. First there is the technology perspective. Augmented Reality markers have evolved from very defined shapes to almost anything that has a detailed structure. The next steps in this path is the reduction of such detail necessary to detect a marker. There is a new set of ARTrackers evolving that tracks and maps a parallel object. This means that the tracker searches for known points on an object and continuously searches for new points that it can use to identify the position of the tracked object. This technology does not necessary need a marker since it can use any detail given in a scene. Combined with a simplified marker, these technologies together might enable the tracking of an object that only consists of the shapes that tell the affordances of an object, so that the object surfaces them self do not need to carry any marker structure.

With a second approach I want discuss the question from an art and design point of view. When we look at Oriental art we see lots of ornaments. These ornaments are omnipresent in this culture and minimally altered, these ornaments can be used as markers. From this perspective the appearance of objects can be informed by cultural development.
This means that such ornaments can be introduced into other cultures for connecting the virtual with the physical. Ornaments and surfaces that have no solid structure can be found in high class cars or every object made from wood. If future AR-trackers will be able to track markers with less details, every wood structure would be able to become a marker no matter how detailed. This means that future Smarter Objects could also be made from ordinary wood. It could change a cultural perspective about the things that surround us away from artificial materials toward natural materials.

The surface of an object always becomes a memory for its owner. Old metal boxes contain scratches from their use and other objects get personalized by painting on the once clean surface. Consider, for example, the Porsche Janis Joplin owned. Such aesthetic personalization can make an object more stable and trackable, while at the same time, an owner feels more connected to the owned object.

The concept of markers on every surface is optimistic, but technological development and the cultural ability to form new fashions suggests that there will be an intersection between culture and technology, where Smarter Objects become everyday objects.
8.6. Impact on Industrial Design

I like to draw the image of the first automobiles that looked like horse wagons, because at that time nobody could imagine what a self-driving wagon would look like. The car needed to evolve into the many shapes it has today. What does it mean to the design of an object to be virtual and not dependent on its own functionality at the moment it was created? From the evaluation came an interesting idea: in such a future, the designer could start to design interfaces only, instead of objects. One could design an interesting knob that a customer could use to control his music with.

Another relevant implication of Smarter Objects on industrial design is that the object does not lose its value over time, as it can be repurposed for other uses. The shape and the story of the object becomes more important than the latest features or the best functionality. This could give designers the freedom to explore the interactions with and the shape of objects in never seen ways.

But it raises some important questions. Who is the person that finally designs these objects? Is it an interaction designer or a product designer or a graphic designer? There would need to be a new category of designer that knows all the necessary product development intersections. It is therefore important to minimize the number of people who are involved in the process of developing Smarter Objects to preserve their integrity. One of the first ideas I had was to build a toolbox that enables designers to create Smarter Objects, which IDEO colleagues confirmed is a strong idea. A toolbox that supports designers with choices of materials, shapes and interactions with Smarter Objects would enable a low entry barrier for designers to play around with the possibilities of Smarter Objects. If this toolbox would be easy to use, similar to Photoshop or Rhino, designers can adapt Smarter Objects for their products. The key lies in the tools provided to designers and the simplicity of the Smarter Objects system.
8.7. What I have Learned

In the beginning when I formed the idea for Smarter Objects, I studied a lot of art. I studied the color and shape theories from the Bauhaus, I visited to the Museum of Fine Arts in Boston and looked at Egyptian hieroglyphs and I learned about the first paintings humans drew on cave walls in France thousands of years ago. I formed an artistic view on Smarter Objects even before I knew that Smarter Objects was the concept I intended to realize.

From these explorations I learned that objects have formed humans. The first objects that were painted on the walls in caves, to everyday objects that have been used to form written languages, and objects that have been abstracted to their essence to understand what the foundational meaning of an object is. From this foundational meaning, cubistic artists created the idea that every object in the world can be built from these basic shapes. The Bauhaus used such basics shapes and colors to explain the foundations of the being. This idea is impressively close to the foundation of a personal computer when looking at the specification of USB HID and the Computer with its Screen. As the Bauhaus found an understanding of the physical human world, the computer uses it for interacting with its virtual world. These interactions are out of balance (Fig. 73). There is too much pure action.
in the physical when pushing buttons and touch screens and too little surface of objects in
the physical when looking at them in a screen. As this has been the first abstract thought
about Smarter Objects, the goal was to bring these three shapes on one line to form a
balance between the physical and the virtual.

Using my design perspective, I began to analyze the use of everyday objects that surround
us. I learned how we actually approach the use of such objects, that we first understand
what we are dealing with, then we shape the object to a form that lets us interact with it
day by day. From this learning I could form the foundation that lead to the design of
Smarter Objects. With the design of Smarter Objects, I learned about Data Flow
programming and the benefits of Augmented Reality to be able to map virtual interfaces
onto physical objects. Exploring the possibilities of Augmented Reality, I understood how
important it is to map interfaces according to the actions the interaction should result in.

I have also learned about the importance of simplicity. I learned that simplicity in many
things combined can in fact produce a lot of complexity. Over and over I needed to reflect
on my work to learn how my interfaces should appear so that I do not visually overwhelm
the user. To find the most simple solution is a craft that I have refined during this thesis.
I learned a lot about this craft but it is a craft that one can never bring to perfection.

From an engineering perspective, I have learned a deep understanding about networks.
I learned about the package length of UDP and TCP/IP connections, a better understanding
about object oriented programming and the power of inheritance. I learned that object
oriented programming is much more generic usable than to be just used for the
programming of personal computer only. The concept of object oriented programming
shapes the industrial world that we live in and also the shape and interactions possible
with Smarter Objects. I have learned about many software libraries that I explored for the
purpose of this thesis, and used them to program the code for Smarter Objects. I learned
new hardware prototyping techniques and the limitations of Augmented Reality.
Finally, from a philosophical perspective, I have learned about the importance for humans
to possess stable, fast accessible and stable objects.
9. Conclusions & Future Work

Before I started this thesis, I was searching for a possibility to interact with everyday objects supported by a digital layer. I was amazed about the possibilities that Augmented Reality provides and it led to all of the explorations and prototypes previously discussed. I searched to find a balance between technology, design and philosophy. Sometimes I drove too deep into the technology part, creating systems that became complex and lacked a useful story. Sometimes the designer inside me was too enthusiastic about the limitless possibilities. But the foundational vision that lies beneath all of this work grew stronger and stronger while writing this thesis.

9.1. Function

Separating interactions to form a new view on objects and their interaction enabled the content of this thesis. Looking back at the prototypes and discussions, it is still questionable if the direct mapping onto objects is the best option, as sometimes you want to have the operation detached from the object. The memory bar is a powerful tool to provide something that is detached from an object but not as abstract as a list of pure object names. For the programming of objects, direct mapping onto the interface is essential for a simple understanding of the programming; this has proven to be very intuitive and easy to use. There is no other experience that makes it so simple to reprogram the functionality of objects and interaction between them. In the future, it will be interesting to research when it is valuable to have an abstract widget on a touchscreen, the Reality Editor or a physical object itself to change the setup for an object.
9.2. Technology

From a technological point of view, the system that drives the Smarter Object must be further explored. What kind of network, what kind of service provider and what kind of manufacturer creates the foundation for Smarter Objects? This thesis shows some starting points that can help to develop such systems and a close goal for future work will be the integration of standards and procedures that have been proven for the internet, such as HTML, server structures and network protocols. One interesting consideration is a peer-to-peer networking structure, whereby every object holds all the logic that is needed to run the full Smarter Objects system, instead of a central server providing this information. Such a system will satisfy the demand that a physical object needs to be a stable object. For example, if I pair a dial with an amplifier this connection should remain stable until these objects vanish, not until a central server disappears or an internet service gets bankrupt. Clear networking interfaces need to be defined so that all Smarter Objects will stay compatible with any other Smarter Object, just as every internet webpage ever produced can be still viewed using any web browser.

9.3. Design

From a design perspective, the Smarter Objects must result in a toolbox that makes it easy for designers to create new products. Such a toolbox must follow the guidelines of any other design tools so that it can become part of a design production pipeline. It will be interesting to develop new paradigms of interaction whereby a product can consist of only the functionality of an operation. We already have such a development, for example, with computer peripherals. I can choose from twenty different joysticks to combine my choice with my computer and operate any game with it.

What if I could easily choose from twenty different rotation knobs that I can combine with the light in my living room, my washing machine, my amplifier and so on without any
specific installation. A user could design the appliances in his home in any manner without the need to compromise design of his environment with the function of a tool. Right now we have many different brands with many different designs in our homes. Often the design does not fit to the design we would like to have. If the operational part is detached from the function, one can use one similar design for the operation of all devices. Companies could sell the pure functionality. For example, an oven could have only the hotplate, whereby all operational knobs are from another company and in a specific design.

9.4. Gestural Interfaces

Another interesting future exploration is the affordance of physical things and therefore externalized gestures. After writing this thesis I think that the research into gesture control will profit a lot from physical objects used for gesture inputs. These objects do not need to be a full electronic object, as they just need to communicate to and guide the user in performing the gesture. Right now, if a user would like to operate a game with the Microsoft Kinect, he would need to learn the gestures necessary for the game, but with a Smarter Object that is linked to a controller, the user can operate the interface without extra learning, as the Smarter Object tells the user how to operate it. The use case story “Watching the Game” gives some impression of how gestures with Smarter Objects can be used for gaming, but finally could support any gesture control.

9.5. Interaction

Another future exploration topic is collaborative interaction with Smarter Objects. Two user could, for example, be in the same room and look at the same Smarter Object with their Reality Editor. Both users would be able to interact with the Smarter Object as they see all interactions at any time. These users could also use two different appearances of their virtual interfaces, while interacting with the same Smarter Object. Such a skin can be
altered by the user so that every Smarter Object is personalized within the Reality Editor but physically the same. Since the physical object defines the function of the Smarter Object, even two different looking virtual objects in the Reality Editor will still have identical functionality. The user can also be on remote location and interact with a shared owned object. For example a child could save a racing car of a friend in his memory bar and take it home. He then could combine the performance of the friend’s car with lights in his room, so that both could start playing together even though they are not in the same room. The Smarter Object can also become kinetic. This means the Smarter Objects system can drive robots or motors. They can become a playful element in kids toys or a powerful tool to operate large machines.

9.6. Vision for Future Work

All of these conclusions and concepts for future work lead to one overall simple vision. In the future we will not have a physical reality and a separate virtual reality that can be seen with a computer that shows us augmentation placed on top of the physical reality. We will have one reality that reaches in the virtual and physical space at the same time. We will not be able to speak about them as separate.

Reality will become a real virtuality without any barriers between the physical and the virtual. As we will be more and more surrounded by objects that live in both worlds, the idea of separated spaces will disappear. We will have more and more computation around us, but it will feel as if computers as we known them today are less present in our daily life.

However we name it, we cannot call it Augmented Reality as this name space is mainly used for mapping data on to real objects or supports the understanding of the real space. It does not influence every aspect of our physical reality. It will be more than ubiquitous or internet connected machines, as every aspect of an object is part of it, visual appearance included. This discipline focuses on the product design of things as this is where all
disciplines merge into one. This development will need all aspects of the new discipline Mechatronics as it requires mechanics, electronics and computer engineering. We can call this development **Real Augmentation** as it is a computational support for reality that changes how physical objects behave, feel and appear. They will feel real.

Figure 74. Illustration for Real Augmentation
10. References


28 Krikorian, Raffi Chant. Internet 0. MIT. August 2004


47 “ARPool” RCV Lab at Queen’s University. August 8, 2013. <http://arpool.ca>


52 “Pure Data - PD Community Site”. puredata.info. August 8, 2013 <http://puredata.info>


95 “Teensy 2.0”. PJRC.COM, LLC. July 8, 2013


105 “Teensy 2.0”. PJRC.COM, LLC. July 8, 2013.


107 “Leap Motion”. Leap Motion, Inc. July 2013.


