G. Niezen, B. van der vlist, J. Hu, and L. M. G. Feijs,
"Using semantic transformers to enable interoperability between media devices in a ubiquitous computing environment,"
in International Workshop on Self-managing Solutions for Smart Environments (S3E 2011)
at the 6th International Conference on Grid and Pervasive Computing (GPC 2011), 2011.

Using semantic transformers to enable interoperability between media devices in a ubiquitous computing environment

Gerrit Niezen, Bram van der Vlist, Jun Hu, and Loe Feijs

Department of Industrial Design, Technische Universiteit Eindhoven
Den Dolech 2, 5612 AZ Eindhoven, The Netherlands
{g.niezen,b.j.j.v.d.vlist,j.hu,l.m.g.feijs}@tue.nl

Abstract. One of the aims of ubiquitous computing is to enable "serendipitous interoperability"; i.e., to make devices that were not necessarily designed to work together interoperate with one another. It also promises to make technologies disappear, by weaving themselves into the fabric of everyday life until they are indistinguishable from it. In order to reach this goal, self-configuration of the various devices and technologies in ubicomp environments is essential. Whether automated and initiated by context-aware entities, or initiated by users by creating *semantic connections* between devices, the actual configuration of the various components (based on their capabilities) should be performed automatically by the system. In this paper we introduce *semantic transformers* that can be employed to enable interoperability through self-configuration mechanisms.

Keywords: semantic transformers, interoperability, self-configuring systems, Semantic Web, ontology

1 Introduction

A key goal of ubiquitous computing [13] is "serendipitous interoperability", where devices which were not necessarily designed to work together (e.g. built for different purposes by different manufacturers at different times) should be able to discover each others' functionality and be able to make use of it [2]. Future ubiquitous computing scenarios involve hundreds of devices, appearing and disappearing as their owners carry them from one room or building to another. Therefore, standardizing all the devices and usage scenarios a priori is an unmanageable task.

One possible solution to solving the interoperability problem through self-configuration is being formulated as part of a software platform developed within the SOFIA project[1]. SOFIA (Smart Objects For Intelligent Applications) is a European research project that attempts to make information in the physical

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

G. Niezen, B. van der vlist, J. Hu, and L. M. G. Feijs,
"Using semantic transformers to enable interoperability between media devices in a ubiquitous computing environment,"
in International Workshop on Self-managing Solutions for Smart Environments (S3E 2011)
at the 6th International Conference on Grid and Pervasive Computing (GPC 2011), 2011.

world available for smart services — connecting the physical world with the information world. The centre of the software platform developed within SOFIA is a common, semantic-oriented store of information and device capabilities called a Semantic Information Broker (SIB). Various virtual and physical smart objects, termed Knowledge Processors (KPs), interact with one another through the SIB. The goal is that devices will be able to interact on a semantic level, utilizing (potentially different) existing underlying services or service architectures.

Ontologies lend themselves well for describing the characteristics of devices, the means to access such devices, and other technical constraints and requirements that affect incorporating a device into a smart environment [2]. Using an ontology also simplifies the process of integrating different device capability descriptions, as the different entities and relationships in the SIB can be referred to unambiguously. Because communication via the SIB is standardized, integrating cross-vendor implementations is also simplified, and technical incompatibilities can be captured by the ontology.

Next to "serendipitous interoperability", another key goal of ubiquitous computing is to make technologies — as from a user's perspective they are still dealing with technologies — disappear, and weave themselves into the fabric of everyday life until they are indistinguishable from it [13]. To reach this goal, self-configuration of the various devices and technologies in ubicomp environments is essential. Whether automated and initiated by context-aware entities, or initiated by users through establishing *semantic connections* (as introduced in [12, 11]) the actual configuration of the various components at a lower level should happen automatically. In this paper we introduce *semantic transformers* that can be employed to enable interoperability between different devices in a ubiquitous computing environment. This is done using self-configuration mechanisms that utilize device capability descriptions to determine how devices may interoperate, either directly or through semantic transformers.

2 Related Work

Various technologies have been developed to discover and describe device capabilities in order to solve the interoperability problem. Universal Plug-and-Play (UPnP) with its device control protocols is one of the more successful solutions[2]. However, it has no task decomposition hierarchy and only allows for the definition of one level of tasks [9].

Current RDF-based schemas for representing information about device capabilities include W3C's CC/PP (Composite Capability/Preference Profiles) and WAP Forum's UAProf (The User Agent Profile) specification. UAProf is used to describe the capabilities of mobile devices, and distinguishes between hardware and software components for devices.

A number of ontologies have been developed for ubiquitous computing environments that may potentially be used to describe device capabilities. Chen

[2] http://upnp.org/sdcps-and-certification/standards/sdcps/

G. Niezen, B. van der vlist, J. Hu, and L. M. G. Feijs,
"Using semantic transformers to enable interoperability between media devices in a ubiquitous computing environment,"
in International Workshop on Self-managing Solutions for Smart Environments (S3E 2011)
at the 6th International Conference on Grid and Pervasive Computing (GPC 2011), 2011.

et al. [3] defined SOUPA, a context ontology based on OWL (Web Ontology Language), to support ubiquitous agents in their Context Broker Architecture (CoBrA). The ontology supports describing devices on a very basic level (e.g. typical object properties are `bluetoothMAC` or `modelNumber`), but it has no explicit support for modeling more general device capabilities.

Ngo et al. [8] developed the CAMUS ontology in OWL to support context awareness in ubiquitous environments. Their device ontology is based on the FIPA device ontology specification[3], with every `Device` having the properties of `hasHWProfile`, `hasOwner`, `hasService` and `hasProductInfo`. Devices are further classified into `AudioDevice`, `MemoryDevice`, `DisplayDevice`, or `NetworkDevice`. For audio, the `hasParameter` property has the `AudioParameter` class as range, with subclasses like `ACDCParameter`, `Intensity` and `HarmonicityRatio`. Unfortunately it does not define a notion of completeness, and the ontology is thus not considered generic enough for general use in ubicomp environments.

The SPICE Mobile Ontology[4] allows for the definition of device capabilities in a sub-ontology called Distributed Communication Sphere (DCS) [10]. A distinction is made between device capabilities, modality capabilities and network capabilities. While the ontology provides for a detailed description of the different modality capabilities, e.g. being able to describe force feedback as a `TactileOutputModalityCapability`, there are no subclass assertions made for other device capabilities. Most physical characteristics of the devices are described via their modality capabilities, e.g. a `screenHeight` data property extends the `VisualModalityCapability` with an integer value, and the `audioChannels` data property is also related to an integer value with `AcousticModalityCapability`. The input format of audio content is described via the `AcousticInputModalityCapability` through an `inputFormat` data property to a string value.

It is not clear whether the modality capabilities should be used to describe the actual content that may be exchanged or the user interaction capabilities. As an example, if a device has an `AcousticOutputModalityCapability`, it is not clear whether the device can provide user interaction feedback (e.g. in the form of computer-generated speech or an audible click), or that the device has a speaker.

3 Semantic Connections

As described in the introduction, *semantic connections* were introduced as a means for users to indicate their intentions concerning the information exchange between smart objects in a smart environment. The term semantic connections is used in the context of the SOFIA project to refer to meaningful connections and relationships between entities in a smart environment. We envision these connections as both real "physical" connections (e.g. wired or wireless connections that exist in the real world) and "mental" conceptual connections that

[3] http://www.fipa.org/specs/fipa00091/SI00091E.html
[4] http://ontology.ist-spice.org/

G. Niezen, B. van der vlist, J. Hu, and L. M. G. Feijs,
"Using semantic transformers to enable interoperability between media devices in a ubiquitous computing environment,"
in International Workshop on Self-managing Solutions for Smart Environments (S3E 2011)
at the 6th International Conference on Grid and Pervasive Computing (GPC 2011), 2011.

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

Semantic connections exist in both the physical and the digital world. They have informative properties, i.e., they are perceivable in the physical world and have sensory qualities that inform users about their uses. The digital counterparts of semantic connections are modeled in an ontology. There may be very direct mappings, e.g. a connection between two real-world entities may be modelled by a `connectedTo` relationship in the ontology. Semantic connections can exist between the following entities: artifacts, smart objects, sensors, UI elements, places, (smart) spaces and persons. Semantic connections have properties like directionality, symmetry, transitivity, reflexiveness and modality.

Crucial to our approach is to make the gap between user goal and action smaller. If we consider streaming music from one device to another, "streaming" now consists of multiple actions that do not necessarily make sense to a user, but are necessary from a technological perspective. In our view, this single high-level goal should have one single high-level action, or at least as few actions as possible. The actual configuration of the devices, i.e., matching device capabilities, transforming content or information to the format that is accepted by the receiving device and negotiating passwords and permissions should happen automatically, based on the user's goals and the constraints of the environment. Our earlier work on mapping the user needs to the actual configurations of the devices uses an heuristic approach at mostly syntactic level [5–7]. In this work this is improved by the semantic transformers. In the following two sections we describe how to model device capabilities and relate them to user actions through the use of semantic transformers.

4 Semantic Media ontology

The Semantic Media ontology, shown in Figure 1, is an application ontology that allows for describing media-specific device capabilities and related media content. A mobile device may be described as follows:

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

The system configures itself through ontological reasoning based on these media type descriptions (as described in Section 6). A media player event of type `PlayEvent`, that would be generated when the mobile device starts playing music, is described as follows:

G. Niezen, B. van der vlist, J. Hu, and L. M. G. Feijs,
"Using semantic transformers to enable interoperability between media devices in a ubiquitous computing environment,"
in International Workshop on Self-managing Solutions for Smart Environments (S3E 2011)
at the 6th International Conference on Grid and Pervasive Computing (GPC 2011), 2011.

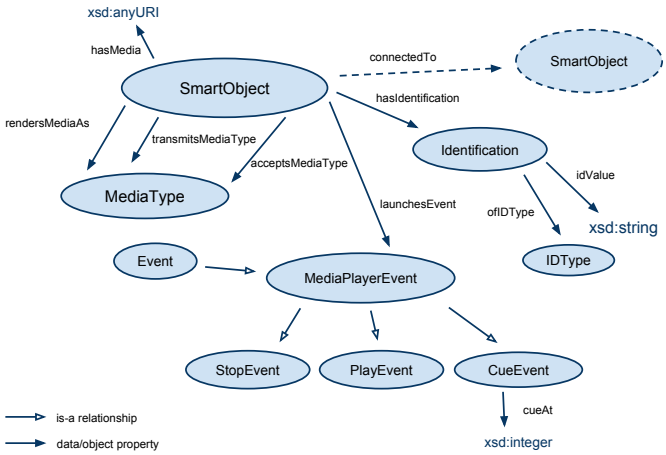

Fig. 1. Semantic Media Ontology

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

Smart objects may be connected to one another using the `connectedTo` relationship. When a device receives an event notification, it first verifies that it is currently connected to the device that generated the event, before responding to the event.

Smart objects may be connected to one another directly if there is a semantic match between transmitted and accepted media types. Otherwise a semantic transformer (introduced in the next section) will have to be be introduced to transform the shared content, while still preserving the actual meaning of the connection.

5 Semantic Transformers

The term *semantic transformers* is used in the context of the SOFIA project to refer to virtual devices that transform user actions into interaction events and perform matching and transformation of shared data and content. To this end, the work done for touch and pen-tablet interaction by the recently established W3C Web Events Working Group was taken as a starting point [1].

A semantic transformer is responsible for interpreting data coming from different smart objects and data sources into possible user goals, and maps them onto the plurality of available services. This facilitates a paradigm change from today's function-oriented interaction to a future of goal-oriented interaction.

User-action events are high-level input actions which capture and report the intention of the user's action directly, rather than just reporting the associated hardware input event that triggered the action. This high level of abstraction

G. Niezen, B. van der vlist, J. Hu, and L. M. G. Feijs,
"Using semantic transformers to enable interoperability between media devices in a ubiquitous computing environment,"
in International Workshop on Self-managing Solutions for Smart Environments (S3E 2011)
at the 6th International Conference on Grid and Pervasive Computing (GPC 2011), 2011.

enables developers to write applications, which will work across different devices and services, without having to write specific code for each possible input device. The W3C Web Events Working Group defined four conceptual layers for interactions (for touch- and pen-tablet interaction) [1]:

physical This is the lowest level, and deals with the physical actions that a user takes when interacting with a device, such as pressing a physical button.

gestural This is a middle layer between the physical and representational layer, and describes specific mappings between the two; for example, a "pinch" gesture may represent the user placing two fingers on a screen and moving them together at the physical layer. This may map to a "zoom-in" event at the representational layer.

representational This is the highest level of abstraction in the event model, and indicates the means by which the user is performing a task, such as zooming in, panning, navigating to the next page, activating a control, etc.

intentional This layer indicates the intention of the task a user is trying to perform, such as viewing more or less detail (zooming in and out), viewing another part of the larger picture (panning), and so forth.

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

Table 1. Examples of representational events in a smart environment

In Table 1 examples of possible representational events are defined. A representational event has more than one possible entity that it can be performed on, as well as more than one possible entity that triggers it, i.e., there exists some ambiguity. Only when no ambiguity exists as to which entity it is performed on, as well as the action which the user is trying to accomplish, we refer to it as an intentional event. Semantic transformations occur between physical actions (such as pressing a button or doing a gesture) and representational events, as well as between representational events and intentional events.

Building on the semantic transformers as they are applied to user actions, they can also be used to map and transform shared content between smart objects. The next section describes how this can be achieved.

6 Implementation

We illustrate our implementation of semantic transformers by means of a use case scenario, where the semantic transformers were introduced as part of a smart home pilot (as defined in the SOFIA project). In this use case scenario, media content is shared among several devices in a smart home setting. Music

G. Niezen, B. van der vlist, J. Hu, and L. M. G. Feijs,
"Using semantic transformers to enable interoperability between media devices in a ubiquitous computing environment,"
in International Workshop on Self-managing Solutions for Smart Environments (S3E 2011)
at the 6th International Conference on Grid and Pervasive Computing (GPC 2011), 2011.

can be shared between a mobile device, a stereo speaker set (connected to the smart space through a Sound/Light KP) and a lighting device that can render the mood of the music with coloured lighting.

In the use case scenario, a Bonding Device [4] renders these light effects, but it accepts only RGB values, and cannot render music directly. The Bonding Device, created by Philips Research Eindhoven, provides a means to connect friends and siblings living apart, allowing them to share experiences, and stay in touch in a new way. The Bonding Device consists of two identical devices that enable indirect communication between relatives or friends at remote locations, by means of detecting human presence and activity at one side, and rendering this information at the other side, to establish a feeling of social connectedness. It responds to both implicit and explicit behavior of the user, e.g. when a Bonding Device detects the presence of a person and his/her proximity to the device, this information is rendered with particular light patterns at the remote side. Additionally it can render lighting effects to communicate the mood of music played by the friend or sibling at one side of the Bonding Device, and have the lighting effects mirrored on the remote Bonding Device.

The Bonding Device is described using the Semantic Media ontology as:

```
BondingDevice rdf:type SmartObject
BondingDevice acceptsMediaType RGBValues
BondingDevice rendersMediaAs Lighting
```

For the implementation of semantic transformers we consider the following scenario: "Mark and Dries start listening to music on a mobile device. They wish to render the music on a lighting device for some visual effects. They establish a semantic connection between the mobile device and the Bonding Device and the light effects are rendered on the Bonding Device."

The Bonding Device accepts dynamic lighting information in the form of a stream of RGB values. What makes this scenario interesting is that the mobile device itself is not capable of transmitting these RGB values, but the Sound/Light KP (a virtual device in the smart space) is. The Sound/Light KP acts as a semantic transformer, converting the music stream generated by the mobile device into the RGB values required by the Bonding Device. From the user's point of view, the only *required* connection is that between the mobile device and the Bonding Device, while the smart space takes care of the rest.

The Sound/Light KP is described as follows:

```
SoundLightKP rdf:type SmartObject
SoundLightKP acceptsMediaType Audio
SoundLightKP transmitsMediaType RGBValues
SoundLightKP  hasIdentification id4321
id4321 ofIDType IPAddress
id4321 idValue "192.168.1.4:1234"
```

The stream of RGB values is sent via a separate TCP/IP connection, so the Bonding Device needs to know whether the source device is capable of communicating via TCP/IP. Since smart objects in the smart space can be identified

G. Niezen, B. van der vlist, J. Hu, and L. M. G. Feijs,
"Using semantic transformers to enable interoperability between media devices in a ubiquitous computing environment,"
in International Workshop on Self-managing Solutions for Smart Environments (S3E 2011)
at the 6th International Conference on Grid and Pervasive Computing (GPC 2011), 2011.

using their IP address and port number, we can use the identification information to infer a `communicatesByTCPIP` data property that can be read by the Bonding Device. To relate the `SmartObject` directly to the `IDType`, we use a property chain:

$$\text{hasIdentification} \circ \text{ofIDType} \sqsubseteq \text{hasIDType}$$ [5]

We then infer the `communicatesByTCPIP` data property by specifying a `TCPIPObject` subclass:

```
Class: TCPIPObject
    EquivalentTo:
        hasIDType value IPAddress,
        communicatesbyTCPIP value true
    SubClassOf:
        SmartObject
```

In order to determine the media source for the bonding device, we first need to perform semantic matching of the media types. We first define `isAcceptedMediaTypeOf` as the inverse property of `acceptsMediaType`, and then define the following property chain:

$$\text{transmitsMediaType} \circ \text{isAcceptedMediaTypeOf} \sqsubseteq \text{convertsMediaType}$$

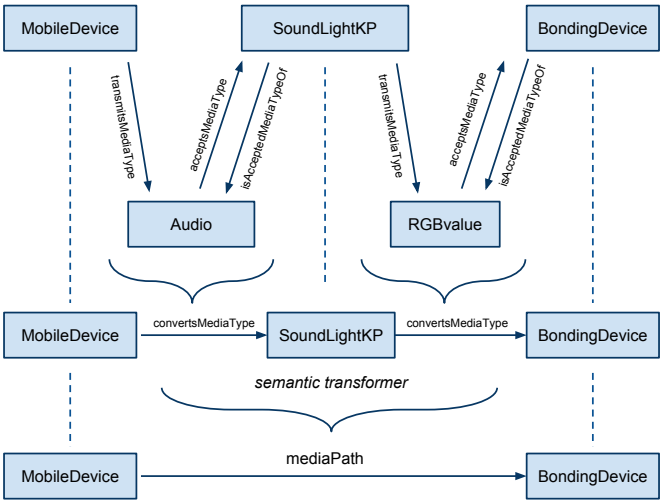

Fig. 2. Inferring the media path

This allows us to match media types between smart objects. We can then infer a *media path* between the mobile device and the Bonding Device with the Sound/Light KP acting as a semantic transformer using another property chain:

[5] The concatenation of two relations R and S is expressible by $R \circ S$, while $R \sqsubseteq S$ indicates that R is a subset of S

G. Niezen, B. van der vlist, J. Hu, and L. M. G. Feijs,
"Using semantic transformers to enable interoperability between media devices in a ubiquitous computing environment,"
in International Workshop on Self-managing Solutions for Smart Environments (S3E 2011)
at the 6th International Conference on Grid and Pervasive Computing (GPC 2011), 2011.

$$\text{convertsMediaType} \circ \text{convertsMediaType} \sqsubseteq \text{mediaPath}$$

To then determine the media source itself we have to use SWRL (Semantic Web Rule Language) [6], as the expressivity of OWL does not allow for inferring the media source if there are more than one `convertsMediaType` relationship linked to the Bonding Device:

```
convertsMediaType(?x1,?x2) ∧ convertsMediaType(?x2,?x3) ⇒ mediaSourceSO(?x3, ?x2)
```

We can also infer whether a device is a Semantic Transformer or not using:

```
Class: SemanticTransformer
    EquivalentTo:
        (canAcceptMediaTypeFrom some SmartObject) and
        (convertsMediaType some SmartObject)
    SubClassOf:
        SmartObject
```

The end result is that the Bonding Device responds to the mobile device's media events (based on the Semantic Connections `connectedTo` relationship), but uses the Sound/Light KP as a media source for generating dynamic lighting. The `connectedTo` relationship between the mobile device and the Bonding Device should only be possible if a media path exists between the two devices. Figure 2 illustrates the entire process of inferring the media path from the original media type definitions.

7 Conclusion

Judging from the experience of implementing the semantic transformers, such an approach to solving interoperability problems appears promising. In simple use cases, we found that the mechanisms developed and presented in this paper show promising results.

Using the the *Semantic Media Ontology*, we were able to define a smart object in terms of the media types it accepts and transmits. Based on these descriptions, semantic transformers can be used to transform media types in order to enable information exchange between devices that would normally not be able to communicate. With only a minimal set of device capabilities described, the system is able to perform self-configuration using ontological reasoning.

Acknowledgment

SOFIA is funded by the European Artemis programme under the subprogramme SP3 Smart environments and scalable digital service.

[6] http://www.w3.org/Submission/SWRL/

G. Niezen, B. van der vlist, J. Hu, and L. M. G. Feijs,
"Using semantic transformers to enable interoperability between media devices in a ubiquitous computing environment,"
in International Workshop on Self-managing Solutions for Smart Environments (S3E 2011)
at the 6th International Conference on Grid and Pervasive Computing (GPC 2011), 2011.

References

1. W3C Web Events Working Group Charter, http://www.w3.org/2010/webevents/charter/
2. OWL web ontology language use cases and requirements (2004), {http://www.w3.org/TR/webont-req}
3. Chen, H., Perich, F., Finin, T., Joshi, A.: SOUPA: standard ontology for ubiquitous and pervasive applications. In: Mobile and Ubiquitous Systems: Networking and Services, MOBIQUITOUS 2004. pp. 258–267 (2004)
4. Dadlani, P., Markopoulos, P., Kaptein, M., Aarts., E.: Exploring connectedness and social translucence in awareness systems. In: CHI 2010 Workshop on Designing and Evaluating Affective Aspects of Sociable Media to Support Social Connectedness. 10 - 15 April 2010, Atlanta, GA, USA (2010)
5. Feijs, L., Hu, J.: Component-wise mapping of media-needs to a distributed presentation environment. In: The 28th Annual International Computer Software and Applications Conference (COMPSAC 2004). pp. 250–257. IEEE Computer Society, Hong Kong, China (2004)
6. Hu, J., Feijs, L.: Ipml: Extending smil for distributed multimedia presentations. In: Interactive Technologies and Sociotechnical Systems, Lecture Notes in Computer Science, vol. 4270/2006, pp. 60–70. Springer, Xi'an, China (2006)
7. Hu, J., Feijs, L.: Ipml: Structuring distributed multimedia presentations in ambient intelligent environments. International Journal of Cognitive Informatics & Natural Intelligence (IJCiNi) 3(2), 37–60 (2009)
8. Ngo, H.Q., Shehzad, A., Liaquat, S., Riaz, M., Lee, S.: Developing context-aware ubiquitous computing systems with a unified middleware framework. In: Yang, L.T., Guo, M., Gao, G.R., Jha, N.K. (eds.) Embedded and Ubiquitous Computing, Lecture Notes in Computer Science, vol. 3207, pp. 239–247. Springer Berlin / Heidelberg (2004)
9. Niezen, G., van der Vlist, B., Hu, J., Feijs, L.: From events to goals: Supporting semantic interaction in smart environments. In: Computers and Communications (ISCC), 2010 IEEE Symposium on. pp. 1029 –1034 (22-25 2010)
10. Villalonga, C., Strohbach, M., Snoeck, N., Sutterer, M., Belaunde, M., Kovacs, E., Zhdanova, A., Goix, L., Droegehorn, O.: Mobile ontology: Towards a standardized semantic model for the mobile domain. In: Service-Oriented Computing-ICSOC 2007 Workshops. pp. 248–257. Springer (2009)
11. van der Vlist, B., Niezen, G., Hu, J., Feijs, L.: Design semantics of connections in a smart home environment. In: Chen, L.L., Djajadiningrat, T., Feijs, L., Kyffin, S., Steffen, D., Young, B. (eds.) Proceedings of Design and Semantics of Form and Movement (DeSForM) 2010. pp. 48 – 56. Koninklijke Philips Electronics N.V., Lucerne, Switzerland (2010)
12. van der Vlist, B., Niezen, G., Hu, J., Feijs, L.: Semantic connections: Exploring and manipulating connections in smart spaces. In: Computers and Communications (ISCC), 2010 IEEE Symposium on. pp. 1 –4 (22-25 2010)
13. Weiser, M.: The computer for the 21st century. Scientific American (September 1991)