G. Niezen, B. van der vlist, J. Hu, and L. Feijs, "From Events to Goals: Supporting Semantic Interaction in Smart Environments," in 2010 IEEE Symposium on Computers and Communications (ISCC), Riccione, Italy, 2010, pp. 1029 – 1034.

# From Events to Goals: Supporting Semantic Interaction in Smart Environments

Gerrit Niezen, Bram J.J. van der Vlist, Jun Hu and Loe M.G. Feijs
*Department of Industrial Design*
*Technische Universiteit Eindhoven*
*Eindhoven, Netherlands*
{*g.niezen, b.j.j.v.d.vlist, j.hu, l.m.g.feijs*}*@tue.nl*

***Abstract*—When we connect smart devices to one another we open up many new possibilities. One interesting possibility is to support high-level semantic interaction without requiring multiple steps on multiple devices. In this paper we investigate how ontologies, runtime task models, Belief-Desire-Intention (BDI) models, and the blackboard architectural pattern may be used to enable semantic interaction for pervasive computing. An initial demonstrator was developed to visualize and manipulate semantic connections between devices in a smart home environment. The demonstrator provides a way for users to physically interact with devices on a high level of semantic abstraction without being bothered with the low-level details.**

***Keywords*-Semantic Web; user interaction; smart home; ontologies; blackboard architectural pattern; task model; BDI model**

## I. Introduction

As computers disappear into smart environments [1], novel human-computer interactions will be needed to deal with the peculiarities of their environments, including invisible devices, implicit interaction, and real, virtual, and hybrid interactions [2].

In the conventional GUI genre, designers have typically developed prepackaged solutions for a predetermined interaction space, forcing users to adapt to their specific interaction protocols and sequences. In ubiquitous computing, the interaction space is ill-defined, unpredictable and emerges opportunistically [3]. There is the risk of engendering a mismatch between the system's model of interaction and the user's mental model of the system. In these conditions, new interaction techniques must be devised to help users to construct helpful mental models, in order to minimize system and user model mismatches.

A related issue is how pervasive computing differs from the sequential nature of traditional GUI interaction. The single point of control that is usually available in such interfaces naturally leads to a sequential organization of interaction. One step inevitably leads to the next; as an example, consider a dialog box that refuses to let you do anything else until you click either `OK` or `Cancel`. When we interact with a smart environment, it is not only the parallel nature of the interaction with the physical world that is different, but also *the many different ways that we might map our tasks onto the features of the environment* [4]. Another difference is that these are not necessarily single-user interactions, but multiple users interacting in the same smart space at the same time.

SOFIA[1] (Smart Objects For Intelligent Applications) is an European research project within the ARTEMIS framework that attempts to make information in the physical world available for smart services - connecting the physical world with the information world. The goal is to enable cross-industry interoperability and to create new user interaction and interface concepts, to enable users to benefit from smart environments.

The real payoff of being able to connect smart devices to one another is that it becomes possible to support high-level services, that would usually involve multiple steps on multiple devices [5]. From a user's point of view, streaming music from a mobile device to a home entertainment system is a single high-level task. In practice there are multiple steps involved, and if the devices involved are from different manufacturers, the user needs to learn the operational details of each device interface in order to perform the task. Universal Plug-and-Play (UPnP) with its device control protocols is not considered an adequate solution, because it has no task decomposition hierarchy and only allows for the definition of one level of task [6].

In this paper, we investigate various models and technologies that may be combined to allow for semantic interaction in smart environments. We also describe an initial demonstrator of a system that implements our first steps towards enabling semantic interaction in a smart environment.

In particular, we consider whether ontologies may be utilized to infer high-level tasks, goals and activities from low-level commands and events (and vice versa). We also look at runtime task models and task-centered interface design to see how these may be applied to pervasive computing scenarios. We also consider whether the blackboard architectural pattern, used by SOFIA to enable a smart space-based computing environment, may be combined with runtime task models and the Belief-Desire-Intention (BDI) model, to result in a usable software architecture for pervasive computing.

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

## II. Background

### A. Architectural patterns for pervasive computing

Existing architectural patterns for software like Model-View-Controller, Document-View and Presentation-Abstract-Control are considered to be inadequate when trying to design software architectures in the pervasive computing domain. Pervasive computing needs new kinds of mechanisms to meet flexibility to change the purpose, functionality, quality and context of a software system [7].

The approach used in SOFIA is to make use of a blackboard architectural pattern to enable cross-domain interoperability. It also makes use of ontologies to enable interoperability without requiring standardization. The first core component of SOFIA's interoperability platform (IOP) is called Smart-M3 and an open source implementation is available online[2].

Given a set of smart devices, a blackboard may be used to share information between these devices, rather than have the devices explicitly send messages to one another. If this information is also stored according to some ontological representation, it becomes possible to share information between devices that do not share the same representation model, and focus on the semantics of that information [8].

SOFIA takes the agent, blackboard and publish/subscribe concepts and reimplements them in a lightweight manner suitable for small, mobile devices. These agents, which are termed Knowledge Processors (KPs) can operate autonomously and anonymously by sharing information through blackboard spaces (see figure 1). The Semantic Information Broker (SIB) is the information store of the smart space, and contains the blackboard, ontologies, reasoner and required service interfaces for the KPs or agents.

### B. Task models

A task model is often defined as a description of an interactive task to be performed by the user of an application through the application's user interface. Individual elements in a task model represent specific actions that the user may undertake. Information on subtask ordering as well as conditions on task execution is also included in the model [9]. In traditional UI design, task models are used only at design time and then discarded [5]. A task-based user interface uses a task model at runtime to guide the user.

A task is commonly defined as an activity performed to reach a certain goal. A goal of a task is considered to be a specific state that is reached after the successful execution of a task. Tasks vary widely in their time extent. Some occur over minutes or hours (like listening to a song or watching a TV show), while others are effectively instantaneous, like switching on the TV.

[2]Available from http://sourceforge.net/projects/smart-m3/

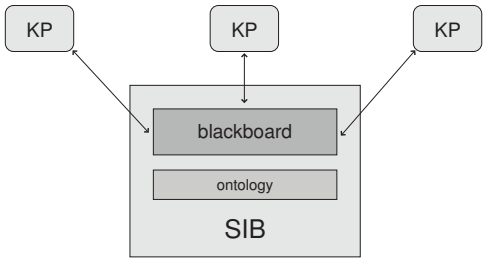


Figure 1. SOFIA infrastructure model

ANSI/CEA-1028 [5] uses a single uniform task representation, compared to other representations where high-level tasks (goals) are separated from low-level tasks (actions). It does so at all levels of abstraction, providing more flexibility when adjusting the level of granularity. We consider this a worthwhile option to explore as this allows for the simplification of the model.

In [10] it is suggested that task models should be based on an ontology that describes the relevant concepts and the relationships between them, independently of any used graphical representations. This also allows for different visualizations of the same task model.

Task decomposition is the most common ingredient of task models. This creates a task tree or hierarchy that can easily be modeled by an ontology. The most important purpose of a task is that it changes something, otherwise it has no reason for existing.

Van Welie et al [10] state that task models should be able to represent the psychological, social, environmental and situational aspects of agents and their tasks. This is why we consider runtime task models a good fit for the Belief-Desire-Intention (BDI) model used for constructing intelligent agents.

### C. Intelligent agents and the BDI model

The BDI model is a philosophical model of human practical reasoning originally developed by Michael Bratman [11], with a number of successful implementations and applications in the agent research community [12], [13]. It could be argued that the BDI model is somewhat dated, as the principles of the architecture were established in the mid-1980s and have remained essentially unchanged since then [14].

A desire is the motivational state of an agent, with a goal having the added restriction that multiple active desires must be consistent (e.g. concurrent desires of "going to a party" and "staying at home" is not possible). When an agent commits to a specific plan with subgoals (based on a belief, or the informational state of the agent) it needs the capability to reconsider these at appropriate times when the world dynamics change. These committed plans and procedures are called intentions, or the deliberative state of the agent.

When building intelligent pervasive computing systems, it may be useful to model computing entities as agents. Chen et al. [15] defined SOUPA, a context ontology based on OWL, to support ubiquitous agents in their Context Broker Architecture (CoBrA). The context ontology covers contexts in the office/campus environment, but it has no explicit support for modeling general contexts in heterogeneous environments.

In SOUPA, agents are defined with a strong notion of agency, which is characterized by a set of mentalistic notions such as knowledge, belief, intention, and obligation. In the SOUPA ontology, both computational entities and human users may be modeled as agents. SOUPA uses the MoGATU BDI ontology to express the beliefs, preferences, intentions and desires of an agent or a user, which makes it possible to rank the priorities of plans and goals.

When the goals, plans, desires, and beliefs of different agents are explicitly represented in the ontologies, this information allows them to share a common understanding of their "mental" states, helping them to cooperate and collaborate. If we are then able to represent the human user's mental states in the ontology, it may help software agents to reason about the specific needs of the users in a pervasive environment. In SOFIA, a reasoner may also be used for truth maintenance, belief revision, information consistency and/or information creation [8].

### D. Related ontologies

*1) SOUPA:* Agent/person ontologies are used to describe actors in a system, where actors include both human and software agents (or computing entities). In SOUPA a computing entity is characterized by a set of mentalistic notions such as knowledge, belief, intention and obligation. The properties of a person agent includes basic profile information (name, gender, age etc.) and contact information (e-mail, phone number, mailing address etc.) SOUPA references several classic domain ontologies to do this:

- FOAF - expresses and reasons about a person's contact profile and social connections with other people;
- MoGATU BDI - describes an abstract semantic model for representing and computing over a user's or an agent's profile in terms of their prioritized and temporarily ordered actions, beliefs, desires, intentions and goals. SOUPA uses this model to help independent agents to share a common understanding of their "mental" states, so that they can cooperate and collaborate. The agents also help to reason about the intentions, goals, and desires of the human users of a system.

As stated earlier, we would like to expand on this concept. According to Ye et al [2], a set of lower independent profile ontologies should be built, each of which would reflect the characteristics of one aspect of a model of a person. These profile ontologies can then be customized and combined to satisfy particular application requirements.

*2) CAMUS:* One of the major goals of context-aware computing is to provide services that are appropriate for a person at a particular place, time, situation etc. In CAMUS, context entities and contextual information are described in the ontologies [16]. For the entities related to agents, there is a top level concept called `Agent`. It has been further classified into `SoftwareAgent`, `Person`, `Organization`, and `Group`. Each `Agent` has a property `hasProfile` associated with it, whose range is `AgentProfile`. An `Agent` is also related through the `isActorOf` relationship to an `Activity`. The Device ontology is based the the FIPA device ontology specification, with every `Device` having the properties of `hasHWProfile`, `hasOwner`, `hasService` and `hasProductInfo`.

While we do not necessarily agree with having organizations and groups being direct subclasses of the `Agent` class, it makes sense to distinguish between software agents and persons, and also to link a profile to each one. The BDI model may form part of the user/agent's profile.

## III. Interaction Tile

### A. The scenario

In the context of the SOFIA project we have developed an initial demonstrator that demonstrates an easy way to visualize and manipulate semantic connections between devices in a smart home environment. A video of the scenario is available[3] and the description is as follows:

"Mark is relaxing at home when his friend Dries arrives. Dries comes with a portable music player loaded with his favourite songs. He wants to play some of his recent collections for Mark. Mark's home is equipped with a sophisticated surround sound system. They decide to enjoy the music from the music player on the sound system. Mark uses his Interaction Tile to see if he can connect Dries's music player to the sound system, which is connected to the home network. The interaction tile indicates that a connection is possible and Mark picks up the tile and shakes it to make the connection.

All the smart devices in the home have a cube-like representation that can be used with the interaction tile. The interaction tile shows the connection possibilities with a high level of semantic abstraction, hiding the complexity of the wired or wireless networks. By interacting with the objects, semantic connections can be built, redirected, cut or bypassed.

Dries starts streaming his music to the environment. Now the room is full with Dries's music and they both enjoy listening to it. Recently Mark has installed an ambient lighting system that can be connected to the sound system and renders the mood of the music by dynamic colour lighting in the room. Mark uses the objects again to create

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

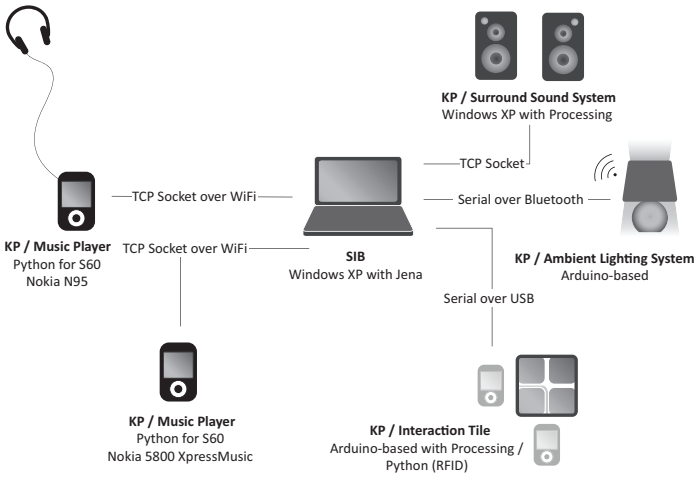


Figure 2. An overview of the demonstrator

another connection and now the room is filled with Dries's music and colourful lighting effects.

Mark's roommate Sofia comes back from work and decides she wants to watch a movie on the TV. She seems somewhat annoyed by the loud music. Mark and Dries do not want to bother her and they again use the objects to re-arrange the music stream. Now the music is streamed to Mark's portable music player while also playing back at Dries's. It is also connected to the ambient lighting system directly, bypassing the sound system. They both are enjoying the same music using their own favourite earphones (and the colourful lighting effects), but without loud music in the environment. Now Sofia can enjoy her movie without any disturbing music."

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

The interaction tile, inspired by Kalanithi and Merrill's "Siftables" [17], was designed to explore the connections and interaction possibilities and manipulation by direct manipulation, and by making simple spatial arrangements. The interaction tile visualizes the various connections by allowing a user to explore which objects are currently connected, and what connections are possible. Coloured LED lighting and light dynamics visualize the connections and connection possibilities between the various devices. By means of putting devices close to one of the four sides of the tile, a user can check if there is a connection, and if not, whether a connection is possible.

A more detailed description of the interaction tile and the demonstrator is available in an article entitled *Semantic Connections: Exploring and Manipulating Connections in Smart Spaces*, also submitted to this workshop [18]. A visual overview of the demonstrator can be seen in figure 2.

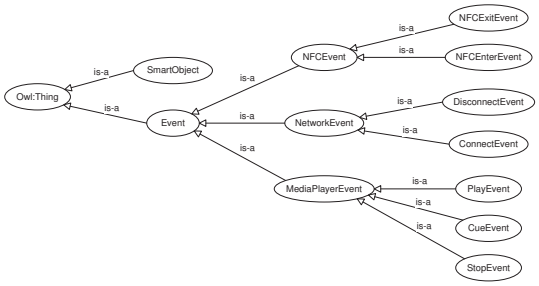


Figure 3. Ontology indicating rdf:type relationships

### B. Ontology used in scenario

The ontology used for the prototype was created in OWL, the Web Ontology Language used to build expressive ontologies for the Semantic Web.

While developing the ontology, we realized that the most promising way of describing low-level interactions seemed to be to describe them in terms of interaction events, which are traceable, reversible and identifiable. An interaction event in the smart space consists of an event ID, timestamp and other related information (e.g. the position of the cube next to the interaction tile). For the scenario described, we distinguished between a number of events that can be seen in figure 3. As a next step we want to determine how we can use the semantics and expressivity of ontologies to infer higher-level tasks and goals from these interaction events.

A notable object property used in the ontology is the `connectedTo` property, which is both *symmetric* and *irreflexive*. Irreflexive properties are a new feature in OWL 2. A symmetric property is its own inverse, which means that if we indicate a `connectedTo` relationship from device A to device B, device B will also have a `connectedTo` relationship to device A. Another way to think of symmetric

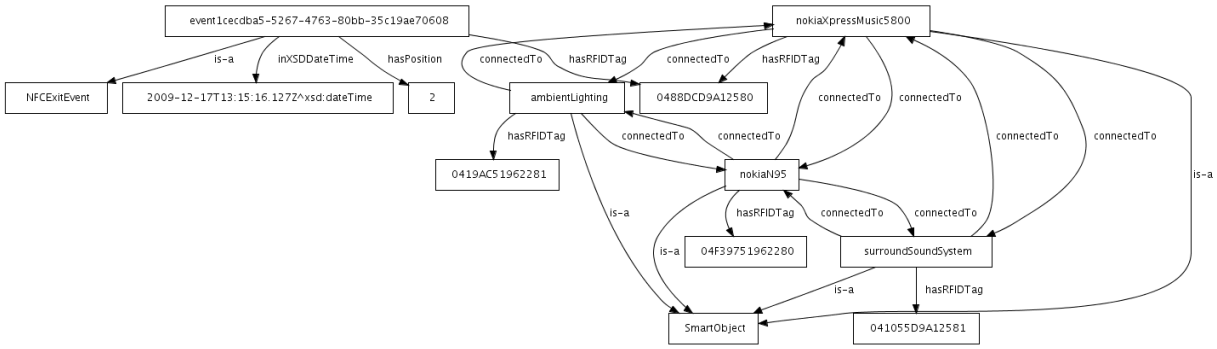


Figure 4. Example instance of ontology showing some individuals

properties is that they are bidirectional relationships. An example of an instance of the ontology used to store (and reason about) the necessary data is shown in figure 4.

An irreflexive property is a property that never relates an individual to itself [19]. This allows us to restrict our model by not allowing a `connectedTo` relationship from a device to itself.

To determine which other smart objects a specific device is connected to, a simple SPARQL query suffices:

```
select distinct ?object where{
deviceID semcon:connectedTo ?object }
```

To get the last event belonging to a specific device, including the position of the cube (representing the device) next to the interaction tile, the SPARQL query is a little bit more complex, but still surprisingly manageable:

```
 select ?position ?eventType where{
deviceID semcon:hasRFIDTag ?tag .
?event semcon:hasRFIDTag ?tag .
?event semcon:hasPosition ?position .
?event a ?eventType .
?event semcon:inXSDDateTime ?time .
FILTER (
?eventType = semcon:NFCEnterEvent ||
?eventType = semcon:NFCExitEvent) }
ORDER BY DESC (?time)
```

A big advantage of using SPARQL and a triple store is that it is easy to add additional constraints and/or specifics to the query, compared to a traditional SQL database where unions between columns and tables can get quite complex very quickly.

We consider all class instances in the triple store to form part of the BDI beliefs of the agent (or KP), from the `connectedTo` relationships between the smart objects, to interaction events that occurred in the past.

If we describe a sequence of actions (plan) in the ontology to achieve a certain intention, we may then use the interaction events to trigger the plan, update beliefs or modify goals. Goals may be defined in the ontology as desires, where we can add the necessary property restrictions to ensure that active desires are consistent.

If sequences of actions are sufficiently defined in the ontology, we may even be able to use a reasoner to infer subsumption hierarchies of plans based on the user's current actions, which in turn would allow us to determine the user's intentions.

## IV. Conclusion

When user interaction and computational intelligence are considered together with the "disappearing computer", pervasive computing becomes part of the broader concept of ambient intelligence. Marzano and Aarts [1] formulated the following five key technology features to define the notion of ambient intelligence:

- Embedded - many networked devices are integrated into the environment.
- Context aware - the system can recognize you and your situational context.
- Personalized - the system can tailor itself to meet your needs.
- Adaptive - it can change in response to you.
- Anticipatory - the system anticipates your desires without conscious mediation.

The SOFIA project tries to solve the interoperability problem by means of a blackboard-based approach. Some of the problems associated with current blackboard-based platforms are scalability and access rights. While our immediate goals does not involve solving these problems, they should be considered as possible constraints.

We consider context awareness to be one of the most important features of a smart environment, especially when

we consider a user's interaction with the smart space. Considering the parallel nature of our interaction with the physical world, any smart space will require context to help it make sense of the many different ways in which users map their tasks onto the environment.

Where the system tries to predict what the user is trying to accomplish, by being adaptive and anticipatory, we need to identify ways to give the users appropriate means to express themselves. The possibilities, available services and information that exist in the smart environment needs to be communicated in a meaningful way. Only if this is done correctly will users be able to build helpful mental models of the functionality the environment has to offer, set goals and make plans on how to act. By developing novel and meaningful interaction devices, the user can then perform the necessary actions and the system can in turn try to understand the user's goals and make the match to its internal models. We see a vital role here for the theory of *product semantics* [20], the study of how artefacts acquire their meaning and use its theories to define common concepts and semantics.

To be able to create a personalized environment, we consider both runtime task models and the BDI model to be important. Task models may be used to describe the user's actions, while the BDI model may be used to represent the psychological, social and situational aspects of the tasks. Once the task model is defined, the system can adapt to the user, by mapping the user's current activity or task to higher-level goals and intentions.

The BDI model approach focuses on the anticipatory aspect of ambient intelligence, where the system tries to predict what the user is trying to accomplish. We also hope to use the low-level events and command currently implemented in the system to automatically infer higher-level tasks and goals, with the final step being able to model the user's (and/or agent's) intentions using an ontology.

## Acknowledgment

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

## References

[1] E. Aarts, "Ambient intelligence: a multimedia perspective," *Multimedia, IEEE*, vol. 11, no. 1, pp. 12–19, 2004.

[2] J. Ye, L. Coyle, S. Dobson, and P. Nixon, "Ontology-based models in pervasive computing systems," *The Knowledge Engineering Review*, vol. 22, no. 04, pp. 315–347, 2007.

[3] J. Coutaz, J. L. Crowley, S. Dobson, and D. Garlan, "Context is key," *Commun. ACM*, vol. 48, no. 3, pp. 49–53, 2005.

[4] P. Dourish, *Where the Action Is*. MIT Press, Sep. 2004.

[5] C. Rich, "Building task-based user interfaces with ANSI/CEA-2018," *Computer*, vol. 42, no. 8, pp. 20–27, 2009.

[6] "UPnP standards," [Last accessed on 17 February 2010]. [Online]. Available: http://www.upnp.org/standardizeddcps/default.asp

[7] E. Niemela and T. Vaskivuo, "Agile middleware of pervasive computing environments," in *IEEE Annual Conference on Pervasive Computing and Communications Workshops*, Orlando, FL, USA, Mar. 2004, pp. 192–197.

[8] I. Oliver and J. Honkola, "Personal semantic web through a space based computing environment," Aug. 2008. [Online]. Available: http://arxiv.org/abs/0808.1455

[9] Q. Limbourg and J. Vanderdonckt, "Comparing task models for user interface design," *The handbook of task analysis for human-computer interaction*, pp. 135–154, 2004.

[10] M. van Welie, G. C. van der Veer, and A. Eliëns, "An ontology for task world models," *Proceedings of DSV-IS98, Abingdon*, pp. 3–5, 1998.

[11] M. E. Bratman, *Intention, Plans, and Practical Reason*. Cambridge, MA: Harvard University Press, 1987.

[12] M. E. Bratman, D. J. Israel, and M. E. Pollack, "Plans and resource-bounded practical reasoning," *Computational Intelligence*, vol. 4, no. 3, pp. 349–355, 1988.

[13] M. Georgeff and A. Rao, "A profile of the Australian Artificial Intelligence Institute [World Impact]," *IEEE Expert*, vol. 11, no. 6, p. 89, 1996.

[14] M. Georgeff, B. Pell, M. Pollack, M. Tambe, and M. Wooldridge, "The Belief-Desire-Intention model of agency," in *Intelligent Agents V: Agents Theories, Architectures, and Languages*, 1999, pp. 1–10.

[15] H. Chen, F. Perich, T. Finin, and A. Joshi, "SOUPA: standard ontology for ubiquitous and pervasive applications," in *Mobile and Ubiquitous Systems: Networking and Services, MOBIQUITOUS 2004.*, 2004, pp. 258–267.

[16] H. Q. Ngo, A. Shehzad, S. Liaquat, M. Riaz, and S. Lee, "Developing Context-Aware ubiquitous computing systems with a unified middleware framework," in *Embedded and Ubiquitous Computing*, 2004, pp. 672–681. [Online]. Available: http://www.springerlink.com/content/xy5be5aeuajwbeu0

[17] D. Merrill, J. Kalanithi, and P. Maes, "Siftables: towards sensor network user interfaces." Baton Rouge, Louisiana: ACM, 2007, pp. 75–78. [Online]. Available: http://portal.acm.org/citation.cfm?id=1226969.1226984

[18] B. van der Vlist, G. Niezen, J. Hu, and L. M. Feijs, "Semantic connections: Exploring and manipulating connections in smart spaces," in *1st Workshop on Semantic Interoperability for Smart Spaces (SISS10)*, Riccione, Italy, Jun. 2010.

[19] J. Hebeler, M. Fisher, R. Blace, and A. Perez-Lopez, *Semantic Web Programming*. Wiley Publishing, 2009.

[20] L. Feijs, "Commutative product semantics," in *Design and Semantics of Form and Movement (DeSForM 2009)*, 2009, pp. 12–19.