

# Competitive Agents for Intelligent Home Automation

Timo Michalski  
tmichalski@techfak.uni-  
bielefeld.de

Marian Pohling  
mpohling@techfak.uni-  
bielefeld.de

Patrick Holthaus  
pholthau@techfak.uni-  
bielefeld.de

Center of Excellence Cognitive Interaction Technology  
Bielefeld University, Bielefeld, Germany

## ABSTRACT

Technologies that aim to achieve intelligent automation in smart homes typically involve either trigger-action pairs or machine learning. These, however, are often complex to configure or hard to comprehend for the user. To maximize automation efficiency while keeping the configuration simple and the effects comprehensible, we thus explore an alternative agent-based approach. With the help of a survey, we put together a set of intelligent agents that act autonomously in the environment. Conflicts between behaviors, identified with a secondary study, are thereby resolved with a competitive combination of agents. We finally present the draft of a user interface that allows for individual configuration of all agents.

## ACM Classification Keywords

I.2.11 Artificial Intelligence: Distributed Artificial Intelligence

## Author Keywords

smart-home automation; agent-based intelligence; online survey; user interface

## INTRODUCTION

Recently, smart environments are receiving a growing amount of interest in research, and computer science in particular [18]. Especially assistance systems, e.g., for elderly people and those who need a constant monitoring, play a significant role [2, 14]. As functionality becomes more sophisticated and the number of devices increases, a key challenge lies in the reduction of complexity for the naive user. Thereby, interfaces must be designed that make efficient use of all available functionalities while simultaneously being simple to control. As a possible approach, we are investigating the use of simple autonomous agents that control the smart environment.

Nowadays, functionality in smart environments is primarily configured manually and knowledge of available features of the environment is needed. Commonly available systems, such

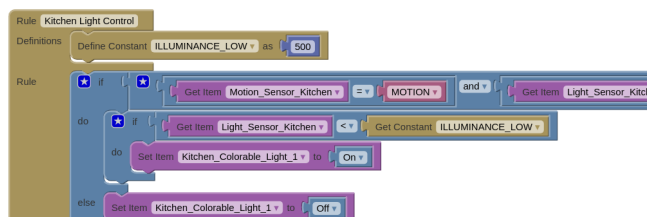


Figure 1. *HABmin*<sup>2</sup> configuration interface for the *OpenHAB* automation software. This example shows a complex rule that controls lights based on motion and illumination inside the kitchen.

as *openHAB*<sup>1</sup> (cf. Figure 1), provide the user with a comprehensive interface that can configure and control the majority of devices on the market. At the same time, such interfaces are still complex to handle [17] and imply an effortful configuration of static rules by the user. Due to an increasing density of smart features, this complexity is also rising rapidly. Concerning this, expert knowledge of the underlying system is mandatory to efficiently exploit the capabilities of a smart home. Therefore, the aim of this work is to dramatically reduce the user's required knowledge about technical details and to simultaneously increase the effectiveness of smart features.

Smart technologies, e.g., energy saving mechanisms, often come with a reduction of comfort for their users and vice versa [6]. A key question is how to individually determine the (perceived) level of control that each occupant has over the functionality [3]: Allowing the user to configure each device or function is overly complex and possibly leads to a leverage of the intelligent automation idea. Fully autonomous systems on the other hand often only have very limited configuration possibilities or lack them completely. Learning-based approaches, for example [13], might provide an effective use of smart features but actions might be neither easily configurable nor fully explainable to the inhabitant anymore. In order to find a compromise between feature effectiveness and configuration comfort, we propose an intelligent home control system based on goal-driven competitive agent behaviors.

Similar to systems like *MavHome* [9], the overall system is composed by a set of agents that try to reach a simple goal while they are active. By allowing users to toggle and configure each agent, functionalities remain transparent while the autonomous agent can still exploit the whole bandwidth of

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*HAI '17*, October 17–20, 2017, Bielefeld, Germany

© 2017 ACM. ISBN 978-1-4503-5113-3/17/10...\$15.00

DOI: <https://doi.org/10.1145/3125739.3132616>

<sup>1</sup><https://openhab.org>

<sup>2</sup><https://github.com/cdjackson/HABmin>

configuration possibilities. In consequence, occupants can use a simple interface to configure high-level goals that are active for a certain amount of time instead of simple trigger-based actions. Agents are thereby competing for interactive resources that are provided by a domotic abstraction in order to minimize agent interdependencies and required knowledge about the environment.

In the following section, we will put our contribution into context by describing the current situation as well as alternative approaches aiming to provide convenient home automation techniques. We then present our concept of competitive agents for intelligent home control. Firstly, we describe the acquisition of a reasonable set of intelligent behaviors and functionality categories with the help of an online survey. Afterwards, we discuss the combination of agents by assigning them categories and identifying possible conflicts and priorities with a second user study. The final part introduces our draft of an interface that enables users to easily enable intelligent behaviors and assign them to locations inside the environment.

### ALTERNATIVE APPROACHES

In the following, a brief overview of related work in the area of smart home automation techniques is presented. Different approaches to an increased effectiveness of a smart environment are discussed and decisions that lead to our agent system are pointed out. A common way to introduce smart behaviors to an interactive environment is by adopting a rule-based approach as it was done by [12] and [5], for example. As stated by Ur et al [16], most use cases can be covered by such trigger-action rules. But as complexity rises, configuration and customization of these rules are also getting more difficult. Since in most cases hardware devices must be configured directly, such systems require prior knowledge to create automation rules that effectively exploit features of the environment. Especially for users without a deep technical understanding, manual configuration of such rules often is overly complex [17] and conflicts with daily activities, for example family life [1].

In order to simplify the configuration problem, there are approaches to home automation that incorporate the learning of actions based on triggers or certain conditions inside the environment. In this context, for example, artificial neural networks (ANNs) have the potential to provide excellent support especially regarding monitoring and controlling aspects of a smart home [4]. Sági et al also use a neuronal network to reduce the energy consumption of a smart environment by optimally scheduling device activities [15] without interrupting the inhabitants. Meng and Lu for example, generate customized services by learning rules depending on the current context in order to satisfy the users' expectations [13].

Multi-agent based systems aim to reduce the workload for users by abstracting from device and rule configuration to more general goals [7]. In *C@sa*, an interactor agent serves as a natural interface for inhabitants and dispatches tasks to device agents that control the environment [10]. In the *MavHome* project [8], a set of agents with a specific goal acts collaboratively to compile an intelligent behavior of the smart home. Again, machine learning techniques are applied for decision making inside each agent and prediction of activities.

### COMPETITIVE AGENTS FOR INTELLIGENT CONTROL

In this work, we present our own agent system for the intelligent control of a smart home that aims to increase the inhabitants' living conditions on several dimensions. For our approach, we make use of a research environment that resembles a common home, the Cognitive Service Robotics Apartment (CSRA), as a reference system [19]. It provides a high density of sensors and actuators to perceive the users and modify their surroundings. With a homogeneous service-oriented software architecture and abstraction, it enables the agents to explore and modify the available components of the environment. Thus, it facilitates abstract services, which are independent of the actual device that is offering a service [11].

As Wolter et al. point out that it is important that system actions are explainable to the user, especially in case of a non-trivial decision [18]. Following this line of argumentation, we refrain from learning techniques for decision making in order to not lose transparency regarding activation and deactivation of agents. Therefore, we use explicit initiators of agents, called triggers which are enabling a certain goal that is understandable for the inhabitants. In the rare case of confusing actions, the possibility remains to identify the intended goal and possibly disable the responsible agent. In order to reduce the amount of relations between single agents and thus modeling effort, agents in our system independently try to fulfill their own goal. A scheduling component thereby resolves possible conflicts externally, allowing for new agents to be integrated seamlessly.

In the following, we will first point out how we determine a first set of agent functionalities that aims towards supporting occupants in everyday situations. Furthermore, these functionalities are assigned to functional categories. Afterwards, a way to detect and resolve conflicts of agents is discussed, before the resulting architecture and the user interface of the system are presented.

#### Agent Functions and their Categories

As the first step of our work, we wanted to figure out in which aspects potential for automated functions is considered. One of the main aspects of smart homes is to increase the *comfort* for inhabitants, as pointed out in [9, 18]. On the other hand, automated behavior can benefit the *energy efficiency* to reduce the power consumption [15]. Additional, *assistance*, *safety* and *security* are widely discussed in research as well [6, 7, 20]. With our work, we are focusing on a common smart home and want to enhance aspects in the everyday life. Therefore, our interest is set to the question, what potential residents think should be increased by automated behaviors. The categories of main interest are forming the dimensions of our system, in which the automated behaviors are interacting. Thus, the agent goals, or functions, that should benefit these aspects, are closely coupled to these dimensions. To determine the categories and to generate a set of functions for our agents, we conducted a first online survey with 52 participants. Our approach for the survey was to have an initial educated guess of the categories and a small set of agent functions. We wanted to have our initial guess approved and afterwards extend it with the survey's result. Based on the categories mentioned above,



**Figure 2. Participants' agent to category mapping.** One column represents the percentage of assignments for a single agent to a functional category, adding up to 100% each. Percentage values are visualized on a scale from white (0%) to dark green (100%).

our initial assumption of the main dimensions is set to *comfort*, *assistance*, *energy saving*, *safety* and *security*. Participants of the survey were first asked to name categories without knowing our initial guess and afterwards, to rate the importance of each of the initial categories. The result shows that the initial categories were all rated important for improvement and no other category, that was not a sub-category, was identified.

The second aspect of the study was to identify new agent functions. An initial set of 12 agent goals with respect to our initial categories was created. The participants were asked to think of automated functions that would be an improvement, without being primed by our ideas. Afterwards they had to rate the impact of randomly chosen functions of our set to our initial categories. Due to this, they got an idea of how we think a function might look like and we gave them the chance to name further ideas. With the survey, 18 new agent goals were created, adding up to a total set of 30 goals. With respect to the CSRA the set was reduced to 19 agent goals that can be realized within the environment. These distinct functions were presumed to be distributed among the five categories. In the following, two simple agent functions are presented that will be used as an illustration example in the next sections:

- A) Keep lights turned off, if no one is present
- B) Generate a random light pattern, if no one is home

### Conflicting Agents and Categories

The complex behavior of the smart home control is emerged from the combination of several agent behaviors. Since a competitive approach of a multi agent system is used, conflicts between agents might occur. A conflict appears, if at least two agents that are following partly or generally excluding goals are triggered at the same time. In order to combine agents in an expectable way, conflicts have to be identified and resolved.

This was the major aspect of the second conducted survey, which has 126 submissions. For investigation, a pair-wise comparison of 19 agent goals was made and the participants had to identify whether two of them might be conflicting. In case of a conflict, they additionally had to point out which one is more important to them and shortly describe how they, as a human, would handle the conflicting situation. With the user preferences, we wanted to figure out a priority scale of the agents to resolve conflicts in a way inhabitants would resolve them on their own. Another aspect of the second survey was to find out which agent goal is assigned to which category of enhancement. We assumed, it is possible to find priorities between the major categories in a large scale. So, the participants

were asked to assign every agent goal to one of the categories. As seen in Figure 2 it is observable that *assistance* is not an independent category on its own, but always seen in combination with a second one, like for example *comfort-assistance*. Due to this result, the dimensions of the system were reduced to *comfort*, *energy saving*, *safety* and *security*. The result of the survey confirms a large-scale priority between the different categories and additionally a small-scale priority between the different agent functions of a category. The importance of the categories was rated as follows: *safety* > *security* > *energy saving* > *comfort*. This means, that every agent of the *safety* category should have a higher priority in case of a conflict than any agent of the other categories. Also, the category internal conflicts were resolved to a priority scale. In conclusion, a priority was assigned to every agent.

In our architecture agents do not know about other agents that might be activated and triggered at the same time. Therefore, conflicts are arising if two initiators are trying to modify the same resource. Initiators are system components like an agent or an inhabitant itself. Hence, a scheduling and arbitration mechanism for resources modification that resolves conflicts by priorities was integrated into the system. The system resolves conflicts with rejecting and canceling actions of lower prioritized initiators to allow higher prioritized ones to apply their actions to the resource. Not only agent conflicts are handled, but also conflicts with other initiators. To ensure that the inhabitant will never lose control of the environment, a human initiator will always have a higher priority than a system component. Depending on the strategy of an agent, the whole action is canceled or subgoals with less devices are achieved, if it gets rejected or canceled.

As an example, the two agent goals mentioned in the previous section will end up in a conflict, if all inhabitants are leaving the home. Agent *A* would already be active in locations where no inhabitant is present. As soon as everyone leaves the environment, it would be triggered in every location and agent *B* would be triggered as well. Agent *A* tries to keep all lights turned off to save energy, as *B* wants to turn some of them on to scare off burglars. Since agent *B* is part of the *security* category which is prioritized higher than the *energy saving* category of agent *A*, it will get the right to modify the lights of the environment.

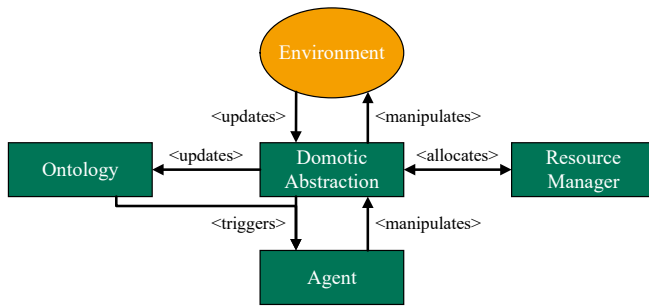
### Resulting Architecture

As we are using the CSRA as a reference, our concept was integrated in its software architecture. The main aspects of the architectural concept are visualized in Figure 3 and will be shortly described in the following. On one hand, we have the environment with all its devices building up the actual state of the smart home. The domotic abstraction layer<sup>3</sup> is the interface to the smart home platform<sup>4</sup> through which the current state of the environment is automatically discovered, represented and manipulated. Updates of states are also stored with corresponding time information in an ontology<sup>5</sup>. Agent functions can be triggered by current states represented by the domotic

<sup>3</sup><http://dal.basecubeone.org>

<sup>4</sup><http://basecubeone.org>

<sup>5</sup><http://ontology.basecubeone.org>



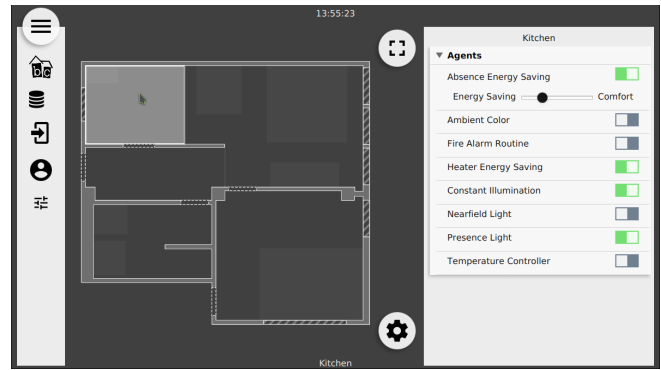
**Figure 3.** Communication architecture between system components (green) and the environment (orange). Agents manipulate the domotic abstraction layer, which in turn allocates resources and alters the environment and updates the ontology.

abstraction as well as from time dependent events of the ontology. For manipulations of the environment the domotic abstraction is providing services. The resource manager<sup>6</sup> is tightly coupled to these manipulation services as a resource must be allocated in order to get the right for manipulation.

With this architecture, a simple workflow example of the above mentioned agent *A* can be described as follows: The domotic abstraction is continuously updated with the current state of the environment with all its devices and other entities. A provider service is representing if a human is present in the given location the agent is assigned to. At a certain point, this state might change to *absence* as no resident is present in this location anymore. The trigger that is observing this state, is triggering the agent as the condition is valid for its execution. Then, the agent will generate the action to keep all the lights turned off in the location, using an operation service of the domotic abstraction. Resource allocation is requested at the resource manager and approved if feasible. Afterwards the action is applied to the devices of the environment. As long as the current state of the environment does not validate an agent’s trigger condition, the agent is not allowed to manipulate any entities anymore. Accordingly, possibly allocated resources are released to be used again by other initiators.

By now, the agent goals are described as a permanent and non-flexible behavior. Since this might not satisfy the users’ needs, a way to adjust these agents is introduced with respect to the dimensions of our system. Even though the category of an agent is fixed by its main intention, the emphasis of how to reach its goal can be adjusted. For every agent, it is identified in which dimensions it is interacting and how a change of the emphasis will influence the agent’s behavior. To give an example, agent *A*’s main intention is to turn off the lights, if no person is present, to save energy. The identified dimensions are therefore *energy saving*, but also *comfort*, as the reaction time of the agent is influencing the *comfort* aspect. Thus, a change of the agent’s emphasis on a scale between *energy saving* and *comfort* will change its reaction time. Completely set to *energy saving*, it will turn off the lights immediately, but will wait for an increasing number of seconds, the more the emphasis is set to *comfort*.

<sup>6</sup><https://github.com/pholthau/arbitration-service>



**Figure 4.** User interface for configuring agents. With an interactive floor plan, users can select the area for agent activation. On the right-hand menu, all available agents and their activation states are shown.

## User Interface

The configuration of the agent system and the used agents should not require expert knowledge of the environment. Therefore, an intuitive user interface is introduced to provide a simple way to easily modify the smart home control. Users should be able to select which agent functions they want to integrate at a certain location of their environment. Also, they should be able to adjust the emphasis of an agent’s dimensions.

A new agent control section is integrated into the existing user interface<sup>7</sup> of our smart home platform, as seen in Figure 4. It consists of a dynamic visualization of the environment with all its devices, services and control functions for modification. The main aspect is to simply display the environment on an interactive map based on the floor plan of the smart home. A menu tab is designed for agent configuration. To assign an agent to a location, a user must select this location on the interactive map by clicking on it. As a feedback, it gets highlighted and the agent configuration tab is showing a list of agents that are available on this location. The user can choose from this list which of the agents should be activated. Additionally, clicking on the agent will extend the menu for configuration of the agent dimension emphasis. A metric that combines knowledge of all assigned agents and their emphases, can be later employed to give feedback to the user of the estimated overall emphasis of the agent system. As a result, the user gets an idea how modifications of a single agent’s properties also affect the overall emphasis.

## CONCLUSION

In this ongoing work, we present the acquisition, categorization, and implementation of a set of intelligent agents that control a smart home in a competitive fashion. The emerging complex behavior aims to increase the inhabitants’ living conditions while being also easily comprehensible and configurable with the help of a simple user interface.

## ACKNOWLEDGMENTS

This work was supported by the Cluster of Excellence Cognitive Interaction Technology ‘CITEC’ (EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG).

<sup>7</sup><http://bcozy.org>

## REFERENCES

1. Bessam Abdulrazak, Sylvain Giroux, Bruno Bouchard, H el ene Pigot, and Mounir Mokhtari (Eds.). 2011. Toward Useful Services for Elderly and People with Disabilities. 6719 (2011). DOI: <http://dx.doi.org/10.1007/978-3-642-21535-3>
2. Bessam Abdulrazak, Sylvain Giroux, Bruno Bouchard, H el ene Pigot, and Mounir Mokhtari (Eds.). 2011. *Toward Useful Services for Elderly and People with Disabilities*. Lecture Notes in Computer Science, Vol. 6719. Springer Berlin Heidelberg, Berlin, Heidelberg. DOI: <http://dx.doi.org/10.1007/978-3-642-21535-3>
3. Simin Ahmadi-Karvigh, Ali Ghahramani, Burcin Becerik-Gerber, and Lucio Soibelman. 2017. One size does not fit all: Understanding user preferences for building automation systems. *Energy and Buildings* 145 (2017), 163–173. DOI: <http://dx.doi.org/10.1016/j.enbuild.2017.04.015>
4. Rezaul Begg and Rafiul Hassan. 2006. Artificial Neural Networks in Smart Homes. In *Designing Smart Homes: The Role of Artificial Intelligence*. Vol. 4008. 146–164. DOI: [http://dx.doi.org/10.1007/11788485\\_9](http://dx.doi.org/10.1007/11788485_9)
5. Dario Bonino and Fulvio Corno. 2010. Rule-based intelligence for domotic environments. *Automation in Construction* 19, 2 (mar 2010), 183–196. DOI: <http://dx.doi.org/10.1016/j.autcon.2009.10.008>
6. Raymond J. Cole and Zosia Brown. 2009. Reconciling human and automated intelligence in the provision of occupant comfort. *Intelligent Buildings International* 1, 1 (jan 2009), 39–55. DOI: <http://dx.doi.org/10.3763/inbi.2009.0007>
7. Diane J. Cook. 2009. Multi-agent smart environments. *Journal of Ambient Intelligence and Smart Environments* 1, 1 (2009), 51–55. DOI: <http://dx.doi.org/10.3233/AIS-2009-0007>
8. Diane J. Cook, Michael Youngblood, and Sajal K Das. 2006. A multi-agent approach to controlling a smart environment. In *Designing Smart Homes: The Role of Artificial Intelligence*. Vol. 4008. Springer, 165–182. DOI: [http://dx.doi.org/10.1007/11788485\\_10](http://dx.doi.org/10.1007/11788485_10)
9. Diane J. Cook, Michael Youngblood, Edwin O. Heierman, Karthik Gopalratnam, Sira Rao, Andrey Litvin, and Farhan Khawaja. 2014. MavHome: an agent-based smart home. *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003. (PerCom 2003)*. October (2014), 521–524. DOI: <http://dx.doi.org/10.1109/PERCOM.2003.1192783>
10. B De Carolis, G Cozzolongo, S Pizzutilo, and V. L. Plantamura. 2005. Agent-Based Home Simulation and Control. In *Foundations of Intelligent Systems*. 404–412. DOI: [http://dx.doi.org/10.1007/11425274\\_42](http://dx.doi.org/10.1007/11425274_42)
11. Patrick Holthaus, Christian Leichsenring, and Jasmin et al. Bernotat. 2016. How to adress smart homes with a social robot? A multi-modal corpus of user interactions with an intelligent environment. In *Proceedings of the 10th Language Resources and Evaluation Conference*. European Language Resources Association (ELRA), 3440–3446.
12. Chui Yew Leong, A. R. Ramli, and Thinagaran Perumal. 2009. A rule-based framework for heterogeneous subsystems management in smart home environment. *IEEE Transactions on Consumer Electronics* 55, 3 (2009), 1208–1213. DOI: <http://dx.doi.org/10.1109/TCE.2009.5277977>
13. Z. Meng and J. Lu. 2016. A Rule-based Service Customization Strategy for Smart Home Context-Aware Automation. *IEEE Transactions on Mobile Computing* 15, 3 (2016), 558–571. DOI: <http://dx.doi.org/10.1109/TMC.2015.2424427>
14. Meg E. Morris, Brooke Adair, Kimberly Miller, Elizabeth Ozanne, Ralph Hansen, Alan J. Pearce, Nick Santamaria, Luan Viega, Maureen Long, and Catherine M. Said. 2013. Smart-Home Technologies to Assist Older People to Live Well at Home. *Journal of aging science* 1, 1 (2013), 1–9. DOI: <http://dx.doi.org/10.4172/2329-8847.1000101>
15. M. S agi, D. Mijic, D. Milinkov, and B. Bogovac. 2012. Smart home automation. In *2012 20th Telecommunications Forum (TELFOR)*. 1512–1515. DOI: <http://dx.doi.org/10.1109/TELFOR.2012.6419507>
16. Blase Ur, Elyse McManus, Melwyn Pak Yong Ho, and Michael L. Littman. 2014. Practical Trigger-action Programming in the Smart Home. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 803–812. DOI: <http://dx.doi.org/10.1145/2556288.2557420>
17. Heleen Vandaele, Jelle Nelis, Tim Verbelen, and Chris Develder. 2016. Remote management of a large set of heterogeneous devices using existing IoT interoperability platforms. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST 170* (2016), 450–461. DOI: [http://dx.doi.org/10.1007/978-3-319-47075-7\\_49](http://dx.doi.org/10.1007/978-3-319-47075-7_49)
18. Diedrich Wolter and Alexandra Kirsch. 2017. Smart Environments: What is it and Why Should We Care? *KI - K unstliche Intelligenz* 0, 0 (2017), 0. DOI: <http://dx.doi.org/10.1007/s13218-017-0498-4>
19. Sebastian Wrede, Christian Leichsenring, Patrick Holthaus, Thomas Hermann, and Sven Wachsmuth. 2017. The Cognitive Service Robotics Apartment: A Versatile Environment for Human-Machine Interaction Research. *KI - K unstliche Intelligenz* 31 (2017), 299–304. Issue 3. DOI: <http://dx.doi.org/10.1007/s13218-017-0492-x>
20. G.M. Youngblood, L.B. Holder, and D.J. Cook. 2005. Managing Adaptive Versatile Environments. In *Third IEEE International Conference on Pervasive Computing and Communications*, Vol. 2005. IEEE, 351–360. DOI: <http://dx.doi.org/10.1109/PERCOM.2005.23>