An Interaction-Centric Dataset for Learning Automation Rules in Smart Homes

Kai Frederic Engelmann[1], Patrick Holthaus[1], Britta Wrede[1], Sebastian Wrede[1,2]
[1]Cluster of Excellence Cognitive Interaction Technology (CITEC), Bielefeld University, Germany
[2]Research Institute for Cognition and Robotics (CoR-Lab), Bielefeld University, Germany
kengelma/pholthau/bwrede/swrede@techfak.uni-bielefeld.de

Abstract

The term *smart home* refers to a living environment that by its connected sensors and actuators is capable of providing intelligent and contextualised support to its user. This may result in automated behaviors that blends into the user's daily life. However, currently most smart homes do not provide such intelligent support. A first step towards such intelligent capabilities lies in learning automation rules by observing the user's behavior. We present a new type of corpus for learning such rules from user behavior as observed from the events in a smart homes sensor and actuator network. The data contains information about intended tasks by the users and synchronized events from this network. It is derived from interactions of 59 users with the smart home in order to solve five tasks. The corpus contains recordings of more than 40 different types of data streams and has been segmented and pre-processed to increase signal quality. Overall, the data shows a high noise level on specific data types that can be filtered out by a simple smoothing approach. The resulting data provides insights into event patterns resulting from task specific user behavior and thus constitutes a basis for machine learning approaches to learn automation rules.

Keywords: Behavioral Corpus, Smart Homes, Internet of Things, Ambient Intelligence

1. Introduction

In a recent literature review, Mennicken et al. (2014) examine the current scientific attempts of realizing the concept of a *smart home*. They name the collaboration of environment and user – or shared autonomy – as one of the promising challenges in the field. A system that suggests automatic behaviors for the user is therefore suggested as a possible solution to accomplish satisfying home automation routines.

Consequently, a smart environment has to be able to offer users a set of functionalities based on their behavior. Ur et al. (2014) thereby revealed in a study that the functionalities can be enabled on the basis of simple trigger-action rules in more than 60% of the cases. In order to compute these kind of rules, many existing approaches apply techniques that learn user behavior. Aztiria et al. (2008), for example, use machine learning techniques are combined with a speech recognition system that allows for verbal programming of the environment. Such a system however still needs manual confirmations or adjustments to control devices like roll-shutters or light systems. Only a few approaches, like MavHome (Cook et al., 2003), are able to skip programming routines and learn the rules unsupervised instead. In an intelligent room, residents are observed and the system tries to predict user behavior.

We also claim that it is feasible to learn a certain set of rules, e.g. lowering temperatures if the window is open, solely from the observation of user behaviors. Those rules can then be assembled in a system that is aimed to support inhabitants of a smart home for example by suggesting a certain configuration preset to the user. In contrast to existing home automation technology, such a system is able to infer more general rules than remembering the status of a single device at a certain time, e.g. turn on the coffee machine at six o'clock. Instead, it provides the capability to relate multiple devices with a user preference.

In order to obtain such preferential rules, a data-set of user actions that lead to a given preferred configuration is necessary. Such a data-set can then be employed to train an algorithm that learns the appropriate relations. Unlike in other fields of machine learning, e.g. speech understanding or face detection there are no appropriate data-sets available to public use as far as we know. We therefore present a new kind of corpus that is explicitly designed to suit the requirements of learning ambient configurations from user behavior.

In this paper, we describe the characteristics and extent of the obtained corpus in greater detail. Section 2. elucidates on the experimental set-up in which the raw data has been recorded. More information on the corpus content is provided by Section 3.. A preliminary analysis that consists of a segmentation and a transformation step is presented in Section 4.. This Section also gives insights into first findings by comparing successful and non-successful segments. Gained results are then briefly discussed in Section 5. and the paper concludes in Section 6. with a short summary.

2. Experimental Set-up

In order to obtain reasonable data from situations that can be found in real-world scenarios, the presented corpus has been recorded in the *cognitive service robotics apartment* (CSRA)[1]. This smart environment consists of a fully operational kitchen, living room, and bathroom, equipped with a set of standard furniture (cf. Figure 1). In addition to standard home automation hardware like thermostats or motion sensors, more sophisticated devices, such as microphones, cameras, and even a mobile robot, are installed in the apartment. Consequently, the CSRA qualifies as the ideal surroundings for recording this data-set because it (a) offers a several opportunities for user manipulation and (b) allows for the acquisition of detailed information about the execution of these actions.

[1] `http://cit-ec.de/en/content/cognitive-service-robotics-apartment-ambient-host`

Figure 1: Overview shot of the cognitive service robotics apartment. In the foreground, a study participant is opening one of the living room's windows. In the background, there are the hallway and kitchen area. On the central wall in front of our mobile robot there is the control panel for manipulation of the temperature and radio.

2.1. Participants

In total, N=59 subjects participated in the study and were recorded successfully. All of them filled in a questionnaire. There were 30 male and 29 female participants of ages between 18 and 64. 54% of them (32 participants) are between the ages of 22 and 27. Almost two thirds (63%, or 37 participants) of our participants indicated that they did not have any experience in smart home environments or with programmable appliances.

2.2. Instructions

The investigator and the participant entered the apartment together. After a short introduction the investigator hands over a list of tasks to the participant and leaves the room. We asked them to solve each task on their own. Most participants followed exactly our instructions, i.e. they went through the tasks in the given order (40 runs, 66%). Nevertheless, a large part (up to 21 runs on task 5, 36%) significantly deviated from the experimental script resulting in more complexly structured data (see below, section 4.3.). The tasks that were given to the participants were:

1. Air the room by adjusting the thermostat and opening the window.

2. Use the electric kettle in the kitchen and make a tea while listening to the radio.

3. Close the window and adjust the temperature back to room temperature.

4. Relax in the living room. Choose a suitable light setting and watch TV after turning off the radio.

5. Leave the room after shutdown of all devices and regulating the temperature.

While task 1 and 3 contain quite clear instructions, tasks 2 and 5 allow for more variation. The participants were free to turn on the light in the kitchen or to turn off the radio if it did not fit their preferences. Task 4 is the most complex one. While the target temperature is only one value, choosing a light setting depends on four variables (power state, hue, saturation, value) for every lamp in the room.

2.3. Questionnaire

After each participant left the apartment, they filled in the questionnaire. Whose results reveal that 54 participants (92%) want to have at least some supportive automation in their own homes. However, the vast majority (46, which corresponds to 78%) wants to remain in control and wants that each automated process – without exception – should be cancelable. This shows the urgent need for control, which entails transparent and manageable interfaces, for the inhabitants of an intelligent environment.

3. Corpus Content

For data acquisition, we essentially use the same setup as Holthaus et al. (2016), i. e. all devices inside the apartment communicate via the same middleware (Wienke and Wrede, 2011). User actions (e. g. the manipulation of a light switch) and sensory input (e. g. the current temperature) are thereby recorded to hard-disk together with their temporal annotations (Moringen et al., 2013). As a result, our corpus provides timely synchronized information on all user-triggered events and the resulting configurations inside the apartment, which is a prerequisite to learn temporal and causal relationships between them. The resulting files are convertible between JSON[2] or XML[3] format.

Furthermore, videos from four different perspectives have been recorded for further evaluation. However, note that due to privacy issues audio and video are not part of the corpus. Only anonymous sensory data and system events will be publicly available. The data-set covers more than 250,000 events from all 59 persons solving the five different tasks inside our ambient living apartment. There are up to 9190 events in a single run. Each run lasts 12 minutes on average. In total, there are more than 40 different types of events in the corpus, of which Table 1 gives a few of the most prominent examples.

The data types from Table 1 can be characterized in more detail as follows. The window handle sensor gives information on the current state of the two windows in the living room. It can take the values *open*, *tilted*, and *closed*. In total, there are three different value expressions and two different streams. The stream publishes as soon as the state changes, i. e. the angle of a handle is altered by the user.

The reed switch sensor provides an *open* or *closed* state for the two windows in the living room, two cupboards in the kitchen, and all four doors of the apartment.

The current temperature is measured with a single sensor in the center of the apartment and is available in *degree Celsius* with one decimal place. Furthermore, a desired temperature can be set at the console in front of the robot

[2] `http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf`

[3] `https://www.w3.org/TR/2008/REC-xml-20081126/`

Event	# Expressions	# Streams
Window handle sensor	3	2
Reed switch	2	8
Target temperature	>100	1
Actual temperature	>100	1
Power consumption	>100	27
Radio	2	1
Lamp (Power state)	2	38
Lamp (Color)	>100	38
Motion ensor	2	16

Table 1: Exemplary excerpt of events from the data-set. For each type, the number of possible expressions as well as the number of streams (data sources) is given.

(cf. Figure 1). An event is published if any of the two change. User modification of the target temperature needs a button click for each change of 0.1 degrees. Thus, a correction of two degrees results in a total of 20 events.
A further sensor gives integer information on the current power consumption in *watt* at 27 of the power outlets in our apartment. In the first 17 trials, an event only occurs if the value changes by at least 23 watts. After these participants, we changed the resolution of the power consumption sensors as we realized that the initial resolution was too low to for example perceive activations of the lamps which have a demand of five watts. In the remainder, the resolution is set to one watt so that we are able to register these kind of changes.
Also at the central console, users can press a button to turn on the radio (cf. task 2). Accordingly, the data-set contains information whether the radio is currently *playing* or *silent*.
In the apartment, there are 38 light bulbs that can be configured to a preferred color. All configuration changes are available as events in the data-set, i. e. the current color is given as three integers for *hue*, *saturation*, and *value* (HSV color space). In addition, the power state is given as *on* or *off* because the lamps might also be switched by the users.
A rough estimate on user position is given via 16 of the apartment's motion sensors. For each of them, an event is triggered if the state changes from *no movement* to *some movement* or the other way round.

4. Data Analysis

This section is an overview of a first analysis of the corpus. We describe some characteristics of the data and show that this corpus is a useful tool to explore relations of behavioral data.

4.1. Segmentation

We surveyed the 59 runs and segmented all data according to the five given tasks, resulting in 59 participants times 5 tasks = 295 segments. This was based on manual annotation of the video data. We tried to compose segments with little noise and the main events of each task. We were looking for moments where we see the participant reading their instruction paper. It turned out that some of the light switch and handle sensor events, we see on the video, are not contained in the data. In order to achieve a better overview of the data, we decided to initially consider only the seven events listed in Table 2 for a precise analysis. These streams are robust publishers with no data loss in our study.
Figure 2 shows a boxplot of the duration of the segments of each tasks. One can recognize the improvement on task 3 in comparison with task 1. Furthermore task 4 is the one with the highest mean because we did not demand a specific time they watch TV.

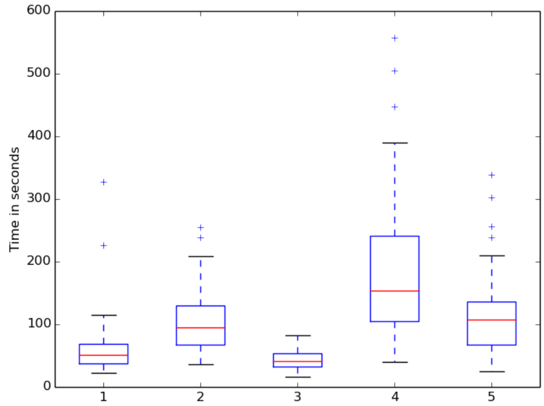

Figure 2: Box plot of the duration of all five tasks. For each task, the median is given as a red line. 50% of all durations are contained within the box, all other data (except outliers) are bound by the black horizontal dashes.

Stream	Tasks	Binary
Reed switch window 1	1,3	yes
Reed switch window 2	1,3	yes
Reed switch door	5	yes
Radio	2,4	yes
Target Temperature	1,3,5	no
Power consumption kettle	2	no
Power consumption TV	4,5	no

Table 2: Selected streams for our data analysis. For each stream, the relevant task is listed and it is given whether it contains binary data or not.

4.2. Transformation from Event- to State-Space

As you can see in Table 2 some events do not have binary data and most events are just publishing on demand (as a light switch). But it is important to know the current state of each device (on/off or open/close) to determine the overall environment configuration. A smart environment can open a window or turn on the TV but it is not able to correct the angle of a handle or the power consumption. So a main step is to quantize those event streams to binary states.
The two types need a different strategy to be transformed into state space. The binary events from the radio button (on/off) and the reed switch (open/closed) are taken as transitions from one state to another. More complicated is the transformation of processes like power consumption

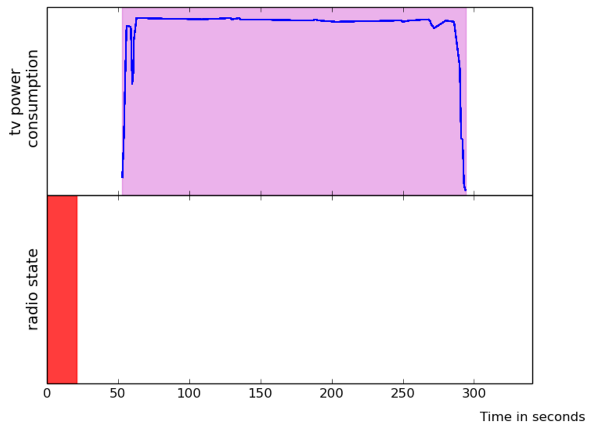

Figure 3: Visualization of the data of an exemplary execution of task 4 by a single user. At around 20 seconds, the radio is switched from on (red box, bottom) to off. 30 seconds later, the TV is switched on (purple box, top), which can be deducted from the power consumption of the respective power outlet (blue line).

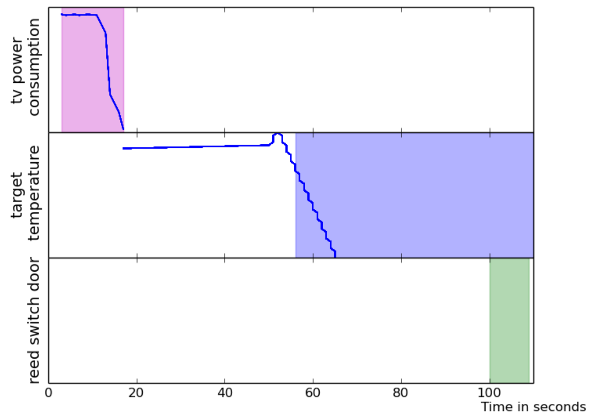

Figure 4: Visualization of the data of an exemplary execution of task 5 by a single user. After switching off the TV (purple box, top), the participant lowers the desired temperature of the apartment (blue box, central). About 50 seconds later, the front door gets opened (green box, bottom).

changes. Small changes occur continuously in many appliances like the fridge which automatically regulates its temperature. And after implementing a lower threshold, the number of events increases rapidly. So it is not possible to interpret every rise and fall as different states. Moreover some streams does not appear until the device turned on. For the purpose of finding proper transformations, we analysed a base line of all power consumption sensors and calculated the mean and the standard deviation from a one minute record of our apartment without inhabitants. We interpret a power state as ”on” if the current consumption is above its mean + standard deviation base line. For the temperature we set the base line to 19 degrees Celsius. A temperature below 18.5 degrees Celsius is interpreted as a cool state.

Examples of a visualization of the states can be seen in Figure 3 for task 4 and in Figure 4 for task 5.

The first third of Figure 4 displays a graph of the current power consumption of the television. The region where we classify the power state as ”on” is colored purple. The second third does the same for the target temperature configuration. We colored the region blue where we assume a cool state. Events which only publish a new current temperature are ignored for this classification. The most bottom third shows the states of the entrance door. In the data are events of the installed reed switch that indicates opening or closing movements. The colored region is the time where the door is opened.

4.3. Complete vs Incomplete Segments

A first analysis of the event data revealed that some segments did not contain all of the streams we assume to be relevant (as can be seen in Table 2). These are no technical mistakes but resulting from different user behavior, i.e. users forgot to carry out a subtask. As this can be interpreted as ”erroneous” data, we assigned the label ”complete” to all segments containing all data in all expected streams and ”incomplete” to all segments where one or more expected stream was missing. Thus, variations within the ”complete” segments can be considered as naturally occurring variants that need to be captured by a learning model. In general, the variations pertain to small changes in sequence, e.g. the radio is turned on before (or after) the user goes to the kitchen. The incomplete segments contain data from task executions where participants skipped some tasks, left the room to get some help or tried to solve tasks in an uncommon way (e.g. starting the TV to turn on a radio). In this kind of data there are thus additional interactions and events without a corresponding task. Table 3 shows how many segments of the different tasks have been classified as complete vs incomplete.

The classification is useful to identify some characteristics in the data. Figure 5 shows all 59 segments of task 5 in the same style as Figure 4. In all of the 38 segments of this task that we classified as complete the environment is in a ”television off” and a ”temperature cool” state when the entrance door opens. The 21 runs that are classified as incomplete can be recognized by missing television or temperature events. This could be a first step to an automation rule.

Task	Complete	Incomplete
1. air room / thermostat	45	14
2. kettle / radio	49	10
3. close window / thermostat	44	15
4. TV / ambient light	44	15
5. leave rooom	38	21

Table 3: Classification of data-set by the presence of all streams per task. For each of the five tasks, the number of complete and incomplete segments is given.

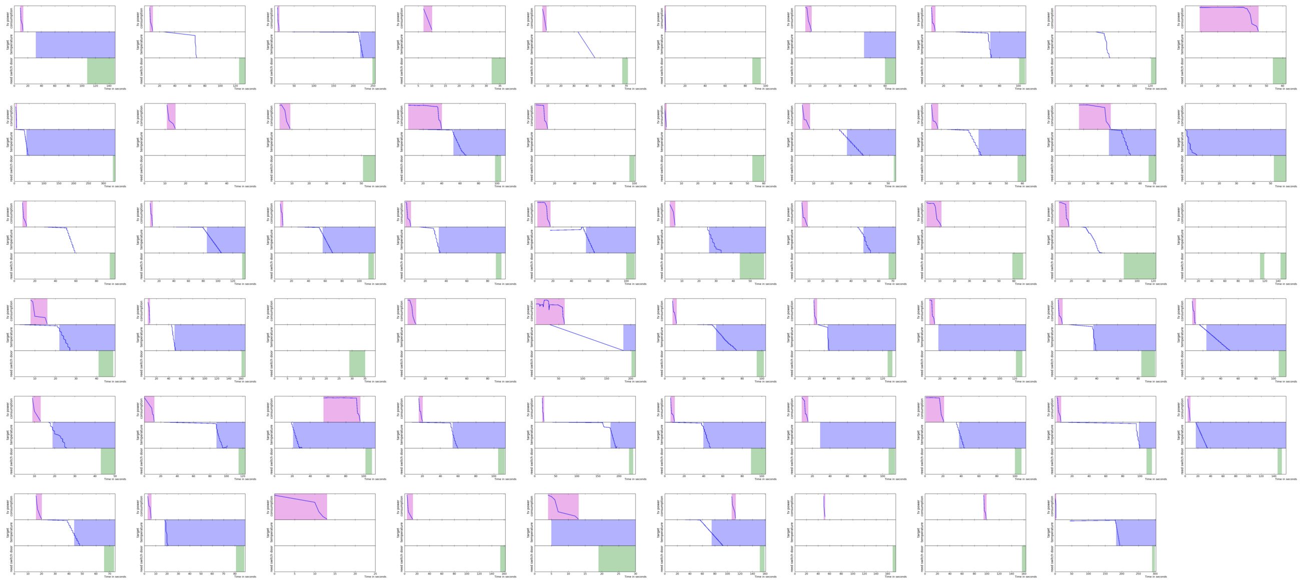

Figure 5: An overview of all 59 runs of task 5. 38 of those runs can be classified as complete according to the presence of the three events ”turn off the tv” (purple), ”lowering the temperature” (blue) and ”open the entrance door” (green)

5. Discussion

A learning approach will need to be based on a prior unsupervised learning phase where the system learns to associate events (triggers such as switches) with their effects (sensory information such as light, temperature or changes in power consumption). Furthermore, most recordings include the opening of the entrance door for arriving and for leaving (46 runs 75%). This raises the question whether a more structured data representation is possible, e.g. by combining events such as door opening and motion detector, in order to distinguish whether a person has left the room or another person has entered the room. Another big challenge is the chronological relation between the events. Some people need less time between two tasks and the corresponding controls during one task.

One way to approach this challenge is by the introduction of (possibly manually encoded) logical or semantic information about the data streams. One important characteristic of some of the events in our corpus is their action character. For example, the event “light on” actually refers to an action the system can carry out, whereas “door open” does not, as it refers to a sensor information. For this kind of data one could add the tag “action”. On the other hand, there are events that refer to sensory information as well as other state information (e.g. “window open” or “door open”) as derived from some events. This kind of information can be interpreted as containing information about (potential) desired end states. All of these types of information can serve as triggers. For example, the power consumption of a certain plug or the state “TV on” can trigger an alarm when it coincides with “apartment empty”, a state that can be derived from certain patterns of movement sensor information and door opening events.

With this kind of additional information the corpus can be much easier used to learn different situations like making tea or watching TV, as well as to learn rules of the form ‘If this trigger then that action to achieve this desired end state’. Even more, this would allow to apply reasoning on the learned rules in order to derive new rules. For example, the system could have learned that the desired end state “light” can be achieved by turning on the lights, but also by opening the blinds. In case one of these actions fails to achieve the desired end state (e.g. because of a broken light or because it is dark outside) it could apply the other rule.

Thus, by adding logical or semantic information to our database learning could be enhanced.

6. Conclusion

We recorded a corpus containing human behavior data as well as system data while solving every day tasks in a smart environment. This is to our knowledge the first corpus containing such rich data. This corpus will help to provide insights into human behavior that are fundamental for developing interactive learning schemes for smart home environments. With this corpus we provide a first data-set allowing to learn relationships between sensor and actuator data, thus allowing to learn simple rules. Further research needs to be carried out in order to show if this data contains sufficient information for predicting the user’s intention or if additional semantic information about the data streams are needed. Overall, our corpus shows that at least simple every day tasks can be observed as specific patterns in the system’s event data.

Acknowledgements

This work was funded by the KogniHome project and the German Federal Ministry of Education and Research (BMBF). It 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). The authors would like to thank Marian Pohling for providing a software connection between RSB and OpenHAB.

Bibliographical References

Aztiria, A., Augusto, J. C., and Izaguirre, A. (2008). Autonomous learning of user's preferences improved through user feedback. In *Proceedings of the 2nd Workshop on Behaviour Monitoring and Interpretation, BMI'08, Kaiserslautern, Germany, September 23, 2008*, pages 72–86.

Cook, D. J., Huber, M., Gopalratnam, K., and Youngblood, M. (2003). Learning to control a smart home environment. *In Innovative Applications of Artificial Intelligence*, 2003.

Holthaus, P., Leichsenring, C., Bernotat, J., Richter, V., Pohling, M., Carlmeyer, B., Köster, N., Meyer zu Borgsen, S., Zorn, R., Schiffhauer, B., Engelmann, K. F., Lier, F., Schulz, S., Cimiano, P., Eyssel, F., Kummert, F., Herrmann, T., Schlangen, D., Wachsmuth, S., Wagner, P., Wrede, B., and Wrede, S. (2016). How to Address Smart Homes with a Social Robot? A Multi-modal Corpus of User Interactions with an Intelligent Environment (in press). In *International Conference on Language Resources and Evaluation*. European Language Resources Association (ELRA).

Mennicken, S., Vermeulen, J., and Huang, E. M. (2014). From today's augmented houses to tomorrow's smart homes: New directions for home automation research. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, pages 105–115, New York, NY, USA. ACM.

Moringen, J., Nordmann, A., and Wrede, S. (2013). A cross-platform data acquisition and transformation approach for whole-systems experimentation – status and challenges.

Ur, B., McManus, E., Pak Yong Ho, M., and Littman, M. L. (2014). Practical trigger-action programming in the smart home. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 803–812, New York, NY, USA. ACM.

Wienke, J. and Wrede, S. (2011). A Middleware for Collaborative Research in Experimental Robotics. In *IEEE/SICE International Symposium on System Integration (SII2011)*, Kyoto, Japan. IEEE.