Reconstruction of Everyday Life Behaviour based on Noisy Sensor Data

Max Schröder, Sebastian Bader, Frank Krüger and Thomas Kirste
Mobile Multimedia Information Systems Group, Institute of Computer Science, University of Rostock, 18051 Rostock, Germany

Keywords: Activity Recognition, Human Behaviour Analysis, Intention Recognition, Smart Home System.

Abstract: The reconstruction of human activities is an important prerequisite to provide assistance. In this paper, we present an activity and plan recognition approach which is based on causal models of human activities. We show, that it is possible to estimate current activities, the underlying goal of the user, and context information about the state of the environment from noisy sensor data. Therefore we use real world data obtained from a smart home system while observing unrestricted activities of daily living in an inhabited flat. We evaluate the accuracy of the recognition for simulated data of different granularity and data obtained from the smart home system. We furthermore show that performance measures solely based on action sequences are not sufficient to evaluate a recognition system.

1 INTRODUCTION

The reconstruction of human behaviour based on sensory inputs is a challenging research problem with several applications. In this paper, we focus on the reconstruction of human behaviour within a living environment instrumented with a simple off-the-shelf smart home system. We show how the noisy sensor data obtained from the smart home system can be interpreted and analysed to reconstruct the behaviour of a person. We employ a causal model of potential activities to disambiguate the sensory inputs.

The development of technical devices enables new features and applications as well as smaller device sizes and cheaper production costs. Additionally, the integration of networking technologies in various kinds of technical devices makes it easy to access internet services. This provides the basis of the Internet of Things (McEwen and Cassimally, 2014). Based on the ubiquitous availability of sensors and computational resources, the integration of assistance technologies into every day life became feasible. However, to provide a pro-active assistance beyond simple if-then-rules we need (a) to analyse the human behaviour and (b) to infer likely future goals. The detection of the goals underlying the protagonist's actions enables an assistive system to execute appropriate supporting actions.

To evaluate our approach for the reconstruction of the human behaviour, we: 1. instrumented an inhabited flat with a smart home system to collect real-world data about the resident, 2. analysed the behaviour to identify exemplary scenarios (activity sequences, e.g. morning routine), 3. collected data covering 40 days in total, and 4. collected 37 sequences for the identified scenarios including a fine-grained annotation of activities. We evaluated the performance of our activity reconstruction model by comparing the real actions (from the annotations) with the estimated ones. A huge amount of the action sequences could be reconstructed successfully, i.e. only small deviations occurred between real and estimated actions.

The contribution of our investigation is a logical modelling approach for human behaviour that enables plan recognition by not only using observation data, but also context and time information to infer the most likely action sequence. Additionally these models can be re-used in other settings. We further show how this can be implemented in a real world setting. The extended evaluation shows the importance of an evaluation with real data compared to simulated data.

This paper is structured as follows: Next, we present our specific setup. In Section 2, after describing some related work with respect to activity recognition, we show how to reconstruct the human behaviour from noisy sensors based on causal models. Therefore, we present our general approach of *Computational Causal Behaviour Models*, and describe our model in detail. In Section 3, we present results of a first evaluation based on simulated as well as real-world data. Finally we draw some conclusions and

Schröder, M., Bader, S., Krüger, F. and Kirste, T.
Reconstruction of Everyday Life Behaviour based on Noisy Sensor Data.
DOI: 10.5220/0005756804300437
In *Proceedings of the 8th International Conference on Agents and Artificial Intelligence (ICAART 2016) - Volume 2*, pages 430-437
ISBN: 978-989-758-172-4

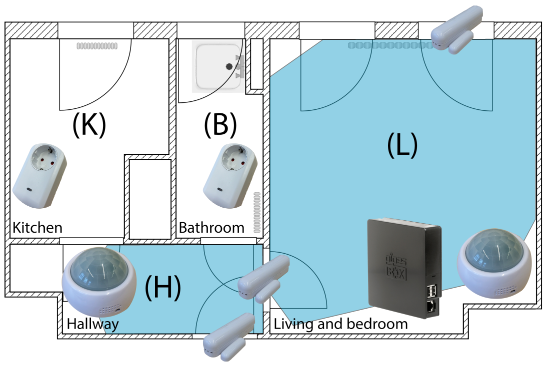

Figure 1: The flat that has been equipped with smart home system components. Blue areas indicate the estimated observation area of the motion function (multi sensor).

discuss possible future work.

For our experiments we instrumented an inhabited ”one-room flat” (combined living and bedroom) with a hallway, kitchen and bathroom (see Fig. 1) with the following devices: 1. *Multi sensor:* Motion, temperature, humidity and luminosity (H,L) 2. *Power switch:* On/off button push, voltage, current, power factor, power wattage and power consumption (K,B) 3. *Door/window sensor:* Open/close change (H,L).

In addition to the raw sensor data, the ground truth has been recorded through a mobile application that allows the on-line annotation of human behaviour. The performed activities are simply selected by the annotator and recorded together with a time stamp.

After an analysis of the daily routines of the resident, four exemplary scenarios have been identified. All four scenarios start and end in the living room.

S1) Fetch and Read Mail: The person walks into the hallway to put on shoes and leaves the flat to fetch mail. After a short time he returns and reads mail as well as newspaper. (10-30 min., every day, partial sensor coverage)

S2) Grocery Shopping: After putting on jacket and shoes, the person leaves the flat to return after approx. 30 minutes. Afterwards the goods are stored in the kitchen. (20-45 min., several times a week, partial sensor coverage)

S3) Go to Work: After putting on jacket and shoes, the person leaves the flat for 4-10 hours. The person then returns to the living room. (5-10 hours, several times a week, partial sensor coverage)

S4) Morning Routine: After waking up, the person opens sun blinds and window, and walks into the bathroom. After taking a shower, the person dresses and has breakfast in the kitchen. (30-90 min., every day, full sensor coverage)

Scenarios S1 to S3 are more strict than S4 in the following sense. The necessary actions to reach the goal have a stricter order. Please note, that this does not influence the action duration. The fixed order of the single actions follows directly from the causal dependencies. E.g., to leave the flat, the shoes have to be put on. This can be done in the hallway only. Therefore the person has to walk to the hall first.

2 BEHAVIOUR RECONSTRUCTION

The literature usually distinguishes between activity recognition from low level sensors and high level plan recognition (Sukthankar et al., 2014). The first uses methods of machine learning such as support vector machines or decision trees to estimate the activity of a human subject. Here, activities often comprise basic operations such as sitting, standing, or walking. Prominent examples are (Bao and Intille, 2004) or (Lee and Mase, 2002). Additionally, to incorporate temporal knowledge, temporal classifier such as Hidden Markov Models (HMM) or Hidden Semi Markov Models (HSMM) are applied. While these approaches usually reach very high recognition rates, they are inherently unsuited to infer high level connections of the current activity, the state of the environment and/or future action or the final goal. Plan recognition, in contrast, deals with estimating the sequence of future activities, including the final goal, from an observed sequence of actions. However, typical plan recognition approaches are not able to deal with uncertainties as known from sensors.

Several researchers strive at combining these fields of research by using Bayesian reasoning to conclude high level knowledge about the plan and goal from low level sensors. (Bui et al., 2002), for instance, introduced the Abstract HMM, a hierarchical representation of plans and goals. The lowest level consists of basic actions that, when combined, result in sub plans of the upper level. Each level thereby combines sub plans from the lower level to create higher level plans. The topmost level consists of high level plans, including the final goal. Action selection probabilities are specified manually. They showed this approach to be viable by inferring the final goal of a human subject from location data. (Liao et al., 2007) introduced a system for assistance in urban environments based on a graphical model that incorporates different behaviours and different goals. They provide several extensions to (Bui et al., 2002), ranging from allowing the user to follow a sequence of goals to learning the action selection probabilities from training data. Additionally, the approach allowed for detecting novel behaviour. The authors use location data (from GPS) to demonstrate that their system is working.

While all of the above mentioned approaches successfully recognise the plan of users from low level sensors, they apply methods of machine learning to establish the action selection (transition model). This requires training data and effectively prevents the transition models from being reused for similar settings. (Baker et al., 2009) used a textual description of the scenario in terms of precondition and effects to reason about the final goal of a human participant. A similar approach was introduced by (Ramírez and Geffner, 2011) and (Hiatt et al., 2011). A computational description is used to generate generative models with sparse transition matrices. Actions are usually described in terms of precondition and effects. (Krüger et al., 2014) introduced the term *Computational State Space Models (CSSM)* to summarise such approaches. CSSMs use compact descriptions that allow a reuse for similar settings.

The application scenario targeted within this paper is the recognition of everyday behaviour in instrumented homes. Several researches focused on this setting, resulting in wide variety of approaches in the literature. (Wilson and Atkeson, 2005) instrumented a complete house with anonymous binary sensors like motion detectors, light barriers, and pressure mats to recognise the activities of the residents. Person specific behaviour models learned from training data and were later used for inference. They showed that the recognition accuracy decreases with increasing number of residents, due to the inability of identification of anonymous sensors. Similarly, (van Kasteren, 2011) applied different temporal models such as HSMM or Hierarchical HMMs to recognise the activities of elderlies in an instrumented home. They also used sensors like motion detectors, reed switches and pressure mats as source of observation. The models were created from training data, but it has been shown that transfer learning can be used to apply a trained model to a similar scenario. However, there is no work on applying plan recognition based approaches to reconstruct the behaviour within instrumented homes.

2.1 CCBM Toolbox

Our investigation uses so-called Computational Causal Behaviour Models (CCBM), an implementation of Computational State Space Models. The main objective of CCBM is to estimate the state sequence of a dynamic system from noisy and ambiguous sensor data. Besides this reconstruction, it is also possible to simulate and validate state sequences of such a system. In this paper, we consider the resident of the flat together with the state of the world as the dynamic system.

CCBM uses an action language, similar to STRIPS (Fikes and Nilsson, 1971) or PDDL (Mcdermott et al., 1998) to describe actions of the system by means of preconditions and effects. The system's state is thereby described as a combination of environment properties (e.g., position of the dining table). Actions model how this state might evolve over time. More formally, consider a set of states $\mathcal{S}$, each given by combinations of propositions, a set $\mathcal{A}$ of action labels, and a ternary relation $\rightarrow \subseteq \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ representing the labelled transitions. If the triple $(s,a,s') \in \rightarrow$, where s and s' are states and a is an action, we say a is applicable in s. The state s' is the result of executing action a in state s. An initial state is a special state, describing the condition of the environment at starting time. A goal is a set of states. The state space is constructed by combining all propositions from the description of the environment.

A behaviour model consists of the following parts: (1) the type-hierarchy, to group elements; (2) predicates and functions, to describe allowed element properties; (3) actions, to specify the system's dynamic; (4) elements in the application scenario, to describe the application specifics; (5) initial state of the environment; and (6) the goal states. (1)-(3) are described in the *domain model*, while (4)-(6) form the *problem description*. See e.g. (Krüger et al., 2014) for a more detailed description of the underlying ideas.

CCBM applies Bayesian Filtering methods to estimate the state sequence of a dynamic system from sensor data. Therefore, a Dynamic Bayesian Network is constructed from the behaviour model. The Bayesian filtering framework requires two submodels to be specified in order to be applied: the observation model, providing the probability $p(y \mid x)$ that the sensor data y is result of the system being in state x and the transition model $p(x_{t+1} \mid x_t)$, which describes the probability that the system's state changes from one state at time t to another at time $t+1$. In addition, CCBM allows to specify a duration model for each action and different action selection heuristics to select an action while being in a given state.

Due to the large state space and the unrestricted duration model, exact inference is intractable. In the CCBM Toolbox we use marginal filtering (Nyolt et al., 2015), which implements an efficient approximation in discrete state spaces.

2.2 Causal Model

The causal model describes the evolution of states from a logical point of view. It consists of two compo-

```
(:action open_window
  :agent resident
  :parameters (?p - person ?l - location)
  :duration (lognormal
    (open_window_duration) (open_window_sd) )
  :precondition (and
    (location ?p ?l)
    (has_windows ?l)
    (not (windows_open ?l)) )
  :effect (and
    (windows_open ?l)
    (ventilated ?p ?l) )
  :observation ( setLocation ?l )
)
```

Listing 1: Example of the action `open_window` in CCBM.

nents: Actions and predicates as well as object types are defined in the *domain model*. The *problem model* contains a description of the initial and goal state as well as the declaration of available objects.

Predicates describe the environment, e.g. the predicate `(has_windows living_room)` indicates that the location `living_room` has windows. Actions are described by precondition-effect-rules containing:

agent. Restrict the possibility to execute the action to a specific agent.

parameters. Action parameters and their type that may be used within the action. Possible parameter combinations will be applied to the action schema; the result are so-called grounded actions.

duration. A probability density function that specifies the duration of this action.

precondition. Conditions that must hold in the current state before action execution. They restrict the number of states where the action is applicable. Preconditions are specified as first order formulae.

effect. A list of changes to the current state that will be applied after action execution.

observation. Observations, e.g. sensor data, will be handled in the observation model (Section 2.4). The observation element of an action description is used to describe effects of the sensor data.

An example action for opening a window is shown in Listing 1. Actions are designed to be applicable in different experimental settings, e.g. multi user environments or other flats. This is implemented by predicates that specify environmental parameters and the use of objects for both residents and locations.

To capture the change in environmental conditions, we used a multi-agent-modelling approach. In the action description in Listing 1, the predicate `(ventilated ?p ?l)` will not be removed automatically if it once is set. This does not sufficiently imitate environmental influences, e.g. air consumption.

```
(define (problem read_mail)
  (:domain everydaylife)
  (:objects
    johndoe - person
    hall bathroom ... outside - location )
  (:init
    (location johndoe living_room)
    (wears_clothes johndoe)
    (has_mail) )
  (:goal (and
    (location johndoe living_room)
    (read_mail johndoe) )
)
```

Listing 2: A snippet of the ”Fetch and read mail” scenario description in CCBM.

Additional actions (for a second agent) capture these influences. Please note that this will not influence the action execution of the resident.

Our setting describes four different scenarios within the same domain to enable comparison of the reconstruction with respect to the same observation data, i.e. a single domain model has been implemented with four additional problem models. An example snippet of the ”Fetch and read mail” scenario is shown in Listing 2.

Our causal model consists of 28 action schemas that result in 172 grounded actions. Four of these action schemas were used to simulate environmental influences such as air consumption. Additionally, 35 predicate schemas of which 8 are for the definition of the flat were used. The corresponding state space sizes are shown in Table 1. The maximum state space size results from the combination of all predicates, which is much higher than the real state space size that only counts reachable states. Again here is a difference between the scenarios S1 to S3 whose actions have a strict order and scenario S4: the number of states is much smaller for the first three.

2.3 Modelling Action Durations

The actions within the causal model require a duration to be defined by a probability density function. The timestamped annotations in our setting enable the calculation of these durations. A set of probability density functions have been evaluated using the Akaike Information Criterion (AIC). The log normal distribution is the one that best fits our durations across all scenarios. Additionally, it must be determined what

Table 1: The size of the state space for the four scenarios.

Scenario ID	S1	S2	S3	S4
Max. size	2^{25}	2^{24}	2^{23}	2^{28}
Real size	744	992	496	11904

one time step in the model represents in reality, i.e. how much time will elapse in reality if a time step in the model is over. We used one second as a time step duration.

The identified time model will be applied by specifying the `:duration` element within each action and the declaration of the corresponding predicates in the problem domain. Because our causal model consists of four problem models, it is possible to use different time models for each of them. This enables further a-priori knowledge about different probability density functions in different situations.

2.4 Observation Model

An observation model in the form of external C++ code needs to be implemented in addition to the causal model. This handles the consecutively fetch of the observation data and calculates the probability that the observation data is the result of the system being in the current state. The system state can be delivered by the causal model using the `:observation` element of an action. E.g. the `open_window` action in Listing 1 calls the C++ function `setLocation` of the corresponding observation model.

3 EVALUATION

To evaluate the performance of our behaviour model and its reconstruction abilities we performed four investigations on different modelling levels. First, we excluded influences from observation model and annotated smart home data by using synthesized action sequences as observation data. Afterwards, simulated sensor data was used for evaluation. Finally, the annotated data was used to investigate both the reconstruction of plans as well as the recognition of the most likely scenario.

3.1 Evaluating the Causal Model

For the first evaluation we used an inbuilt feature of the CCBM Toolbox that reads a list of actions as observation data. This enables the evaluation of the performance of the causal model and the reconstruction of the action sequence while excluding effects from an observation model or real observation data. This is a common method to evaluate the performance of plan recognition systems. Altogether four steps have been done:

1. Five causally correct plans, each achieving the goal, were randomly generated for every scenario.

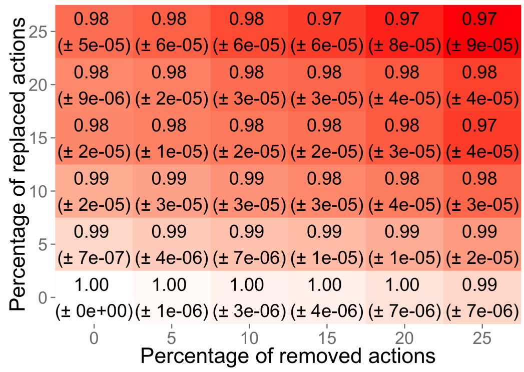

Figure 2: Accuracy of the reconstruction for scenario 1 with noisy action sequences as observations.

2. All plans were randomly modified five times in the following sense: For every time step of the plan the corresponding action was removed with a probability of up to 25 %. In addition, an action was replaced by another action of the same causal model with a probability of up to 25 %.
3. An action sequence was estimated by marginal filtering the noisy plans.
4. Finally, the results of the filter were compared with the initial plans. We use the accuracy to measure the performance, i.e. we counted the amount of correctly estimated grounded actions.

Our expectations were, that higher noise levels result in lower accuracy of the reconstructed action sequence. The evaluation results for the first scenario "Fetch and read mail" are shown in Fig. 2. The heat map displays the accuracy of the reconstruction with respect to the noise levels; the colour indicates the rate of the current accuracy relative to all other accuracies; the numbers inside show the mean accuracy across all noisy plans at the corresponding level.

The accuracy is very high at all noise levels: Even when 25 % of the actions were removed and 25 % of them were replaced, the accuracy is 0.97 ($\pm 9e^{-5}$). That means, 97 % of the action sequences had been correctly estimated if up to the half of the time steps were changed. The impact of replaced actions on the accuracy is bigger than the one of removed actions. The results of the other scenarios were almost identical with similar accuracies. Hence, the causal model can very reliably reconstruct action sequences from noisy action sequences which indicates that a strict model has been defined.

3.2 Evaluating Simulated Sensor Data

The evaluation with simulated sensor data has been done to exclude effects from the annotated observation data while using an observation model. To simulate sensor data, we generated a sequence of locations from a number of random plans. This mimics sensory inputs from e.g. PIR sensors (motion function of the multi sensor). The following steps have been performed:

1. The location of the resident was extracted from five causally correct random action sequences, as they were used in the evaluation of the causal model (Section 3.1), i.e. the resulting file contains the correct location for the resident at every time step. These locations were converted into tuples of boolean values having one value for each location. True values indicate the presence of the resident.
2. The list of location tuples, was modified: True values of correct locations became false with up to 25 % probability and true values were inserted for wrong locations with up to 25 % probability.
3. An observation model was implemented, that parses the location tuples and calculates the corresponding probability.
4. Every location tuple set was used as observation data to estimates the most likely action sequence. The results were compared with the original plans. Again we used the accuracy as measurement of performance.

Fig. 3 displays a heat map of the reconstruction accuracies for scenario S1. Similar to the evaluation of noisy action sequences we expected that higher noise levels result in lower accuracies. However, the results are different. The heat map is split in two parts: If wrong locations are always false (leftmost column), the accuracies are identical for all values of the probability for correct locations. The other part of the heat map indicates the opposite to our expectation, i.e. the accuracy increases if also the probability for wrong locations increases. The impact of the probability of correct locations still coincides with our expectation. If the probability for wrong locations is small, the uncertainty about true values is small, too. However, this results very likely in wrong locations to be considered as correct if their boolean value is true. If the probability of wrong locations increases the uncertainty about true values also increases which will then be compensated by the causal model.

The results for the other scenarios are similar with accuracies not lower than 60 % of correctly estimated actions, i.e. using the causal model, the process can reconstruct at least 60 % of the actions based on noisy location information as the only observation data.

After these first evaluation steps that use computer generated observation data, we investigated the reconstruction with annotated real sensor data.

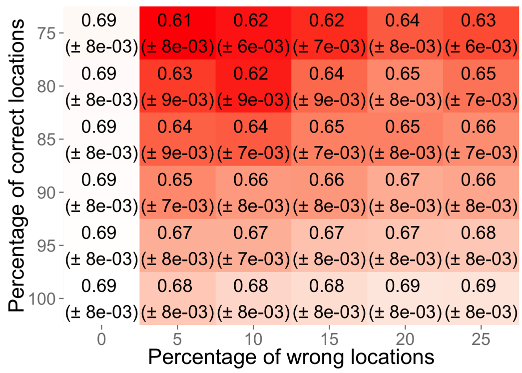

Figure 3: Accuracy of the reconstructions for scenario 1 with simulated location observations.

3.3 Evaluating Annotated Data

In the third evaluation we used real sensor and actuator measurements together with annotations that indicate the currently executed action of the resident. A valid annotation is a list containing all actions from the initial state to the goal state for the corresponding scenario. During the annotation process several incidents may occur, e.g. an action has not been annotated by the resident or an action has been annotated with the wrong label. Here, we used only valid sequences. This can be ensured by converting the annotation list into the corresponding action sequence that has been validated with the causal model.

We restricted the evaluation to data sets which cover a full scenario and are accompanied by causally correct annotations. The filtered results were compared to the plans produced from the annotations. The steps in detail were:

1. Sensor and actuator measurements were combined with valid annotation sets in the way that tuples of features were created for every time step. The features include all directly measured sensor values. Additionally a counter for the number of time steps since the last value update was added. Intermediate values for temperature, humidity, light and instant power wattage were calculated by linear interpolation.
2. Two observation models were implemented: (OM1) a location based approach (as in Sec 3.2) that uses events (motion, power switch and door state changes) and instant power wattage only to recognise appearance of the resident; (OM2) precalculated action class probabilities from the predictions of a decision tree that has been build using the annotated data sets.
3. The observation data sets, i.e. the feature tuples or pre-calculated probability tuples, were marginally filtered using OM1 and OM2.

Table 2: Accuracies for observation model OM1 and OM2. Column # shows the number of valid data sets.

ID	#	Accuracy OM1	Accuracy OM2
S1	5	0.41 ($\pm 7.97e^{-3}$)	0.41 ($\pm 1.23e^{-1}$)
S2	4	0.80 ($\pm 6.59e^{-3}$)	0.58 ($\pm 1.59e^{-1}$)
S3	8	0.99 ($\pm 1.93e^{-5}$)	0.83 ($\pm 1.15e^{-1}$)
S4	9	0.22 ($\pm 2.85e^{-2}$)	0.38 ($\pm 3.24e^{-2}$)

4. The accuracy of the filter results were calculated using plans that have been generated from the annotations.

Table 2 shows the accuracies. The observation model OM1 performs best for the scenarios S2 and S3. The second observation model achieves lower accuracies for the scenarios S1, S2 and S3 compared to OM1, but outperformed it in S4. The variance of the accuracies is much bigger for the observation model OM2, i.e. there were very good and very bad estimations. A reason for this are actions that produce similar sensor and actuator measurements and therefore cannot be distinguished.

Even if the accuracies of the scenarios differ a lot, the mean accuracy of all scenarios sums up to 58 % for the observation model OM1 and 55 % for OM2, i.e. more than half of the actions have been correctly reconstructed from the sensor data.

3.4 Evaluating Goal Recognition

Finally, we compared the likelihood of the marginal filter results for all scenarios using every annotated data set to estimate the pursued goal. Again, the feature tuple and pre-calculated probabilities were used as observation data with their corresponding observation models OM1 and OM2.

Table 3 shows an overview of the scenario recognition performance. As in (Blaylock and Allen, 2014), precision is defined as the relative number of time steps in which the correct scenario is the most likely one. Convergence indicates whether the correct scenario will be identified at all. The convergence point is defined as the earliest point at which the correct scenario is and stays the most likely one. Negative values indicate that the observation data set did not converge.

The results show that here is a difference between the results of scenario S1 to S3 and scenario S4 when using observation model OM1. The first three scenarios have always been correctly identified whereas the last one never converges. Though the convergence point is very late in the scenarios S1 and S2 their precision is rather high. This indicates that there have been multiple trend changes in the probability distribution of the likelihoods. Other than with the first observation model, all of the data sets from scenario S4 were correctly identified with the observation model OM2. Also the precision of 0.922 is very high. However, scenario S1 has never been correctly identified by OM2.

Even if both observation models do not recognise all scenarios equally they could successfully recognise 65 % (OM1) and 70 % (OM2) of the scenarios. Also the average precision is rather high with 44 % (OM1) and 66 % (OM2). These values indicate the general performance of the scenario recognition in our setup.

4 CONCLUSION

In this paper we investigated the reconstruction of human behaviour from smart home system data using a Computational Causal Behaviour Model. The model consists of action descriptions in the form of precondition-effect-rules and a description of initial and goal states. It has been converted into a probabilistic model that was used to filter observation data. The data sets are based on both computer generated random data and real world data which has been recorded in an inhabited flat. The reconstruction is evaluated on four scenarios that have been identified from the behaviour of the resident.

Two kinds of computer generated random observation data have been used. Noisy action sequences could be reconstructed with very high accuracies of at least 97 % of the original actions even when up to half of the observations have been modified. This reveals the robustness of our causal model. The second kind of random observation data were noisy location information. Using them, a reconstruction accuracy of at least 60 % could be achieved, i.e. more than half of the action sequence was reconstructed correctly, even when up to 25 % of the location information were changed.

The smart home system data has been evaluated using two observation models for both reconstruction of the action sequence and recognition of the pursued goal. One of the observation models uses the idea of the simulated location information in the form of considering sensor events that indicate the appearance of the resident. The second observation model uses pre-calculated action class probabilities. Those have been calculated using a leave-one-out cross-validation for the prediction of a decision tree.

The results of the goal recognition revealed that the first observation model correctly identifies scenarios that are strictly ordered and partially outside the flat whereas the second observation model is less de-

Table 3: Comparison of precision, convergence and convergence point for both observation models.

ID	Scenario	Mean Length	Mean Prec.		% Conv.		Mean Conv. Point	
			OM 1	OM 2	OM 1	OM 2	OM 1	OM 2
S1	Fetch and read mail	420	0.383	0.047	1.00	0.00	0.824	-0.002
S2	Go to grocery shopping	2064	0.340	0.728	1.00	0.75	0.910	0.022
S3	Go to work	33602	0.974	0.718	1.00	0.75	0.029	0.032
S4	Morning routine	2869	0.048	0.922	0.00	1.00	-0.0004	0.082

pendent on those. Many of the annotated observation data sets could be successfully identified, but not all scenarios could be recognised equally well.

We have shown that it is not sufficient to evaluate the performance of human behaviour reconstruction solely based on action sequences. Furthermore, Computational Causal Behaviour Models can easily be used together with smart home environments.

In the future our approach will be further evaluated using other environments, e.g. with multiple residents and in other flats. Additionally the set of actions and scenarios will be extended to cover additional scenarios in the flat such as cooking or sleeping. The observation model can also be extended by additional context information, e.g. the personal calendar, which might influence the reconstruction especially if the resident is outside of the flat. Another investigation will be the use of other time and observation models, e.g. a minute based time model and an observation model that considers delay times. Finally, we will combine our setting with the ideas presented in (Yordanova and Kirste, 2016) to learn the necessary models directly from a natural language text.

REFERENCES

Baker, C. L., Saxe, R., and Tenenbaum, J. B. (2009). Action understanding as inverse planning. *Cognition*, 113(3):329–349.

Bao, L. and Intille, S. (2004). Activity recognition from user-annotated acceleration data. In Ferscha, A. and Mattern, F., editors, *Pervasive Computing*, volume 3001 of *LNCS*, pages 1–17. Springer.

Blaylock, N. and Allen, J. (2014). Hierarchical goal recognition. In Sukthankar, G., Goldman, R. P., Geib, C., Pynadath, D. V., and Bui, H. H., editors, *Plan, activity, and intent recognition*, pages 3–32. Elsevier, A'dam.

Bui, H. H., Venkatesh, S., and West, G. A. W. (2002). Policy Recognition in the Abstract Hidden Markov Model. *J. of Artificial Intelligence Research*, 17:451–499.

Fikes, R. E. and Nilsson, N. J. (1971). Strips: A new approach to the application of theorem proving to problem solving. In *Proc. of the second Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 608–620, San Francisco. Morgan Kaufmann.

Hiatt, L., Harrison, A., and Trafton, G. (2011). Accommodating human variability in human-robot teams through theory of mind. In *Proc. of the 22nd Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 2077–2071, Barcelona, Spain.

Krüger, F., Nyolt, M., Yordanova, K., Hein, A., and Kirste, T. (2014). Computational State Space Models for Activity and Intention Recognition. A Feasibility Study. *PLOS ONE*, 9(11):e109381.

Lee, S.-W. and Mase, K. (2002). Activity and location recognition using wearable sensors. *Pervasive Computing, IEEE*, 1(3):24–32.

Liao, L., Patterson, D. J., Fox, D., and Kautz, H. (2007). Learning and inferring transportation routines. *AI*, 171(5-6):311–331.

Mcdermott, D., Ghallab, M., Howe, A., Knoblock, C., Ram, A., Veloso, M., Weld, D., and Wilkins, D. (1998). PDDL - the planning domain definition language. Technical Report TR-98-003, Yale Center for Computational Vision and Control,.

McEwen, A. and Cassimally, H. (2014). *Designing the Internet of Things*. John Wiley and Sons, Ltd.

Nyolt, M., Krüger, F., Yordanova, K., Hein, A., and Kirste, T. (2015). Marginal filtering in large state spaces. *Int. J. of Approximate Reasoning*.

Ramírez, M. and Geffner, H. (2011). Goal recognition over POMDPs: inferring the intention of a POMDP agent. In *Proc. of the 22nd IJCAI*, pages 2009–2014. AAAI.

Sukthankar, G., Goldman, R. P., Geib, C., Pynadath, D. V., and Bui, H. H. (2014). *Plan, activity, and intent recognition*. Elsevier, A'dam.

van Kasteren, T. L. M. (2011). *Activity Recognition for Health Monitoring Elderly using Temporal Probabilistic Models*. PhD thesis, Universiteit van A'dam.

Wilson, D. H. and Atkeson, C. (2005). Simultaneous Tracking and Activity Recognition (STAR) Using Many Anonymous, Binary Sensors. In *Pervasive Computing*, volume 3468, pages 62–79. Springer.

Yordanova, K. and Kirste, T. (2016). Learning Models of Human Behaviour from Textual Instructions. In *Proc. of ICAART 2016*, Rome, Italy.