Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

SIMULATING DAILY ACTIVITIES IN A SMART HOME FOR DATA GENERATION

Jennifer Renoux
Franziska Klügl

School of Science and Technology
Örebro University
Fakultetsgatan 1
Örebro, 70182, SWEDEN

ABSTRACT

Smart Homes are currently one of the hottest topics in the area of Internet of Things or Augmented Living. In order to provide high-level intelligent solutions, algorithms for identifying which activities the inhabitants intend to perform are necessary. Sensor data plays here an essential role, for testing, for learning underlying rules, for classifying and connecting sensor patterns and to inhabitant activities, etc. However, only few and limited data sets are currently available. We present concepts and solutions for generating high-quality data using a flexible agent-based simulation tool. The basic idea is to integrate the simulation of a sensorised apartment with human behaviour modelling based on constraint-based planning that produces a sequence of daily activities. The overall set-up is shown to generate data that exhibits the same relevant properties as data from a comparable real-world apartment.

1 INTRODUCTION

In this work, we describe a new approach of combining advanced constraint-based planning with agent-based simulation to produce datasets that can be used for developing intelligent Smart Homes.

During the last decade, Smart Homes have become one of the hottest topics in computer science. For creating useful applications beyond simple feedback loops such as regulating the lighting based on perceived human presence, algorithms are needed that adapt to the inhabitants' particular situation and activities. For developing, training, optimizing and testing such intelligent algorithms, data is needed. Such data, consisting of sensor readings and associated human status and activity information (the "ground truth") are very expensive to acquire.

Several datasets already exist and are publicly available, such as the MIT activity recognition (Tapia et al. 2004), the CASAS (Cook and Schmitter-Edgecombe 2009) or the ARAS (Alemdar et al. 2013) datasets. Yet, they are limited in the number of activities and occupants they consider, as well in the sensor usage. To test and train algorithms, it is important to have access to datasets for specific scenarios or specific types of person. Indeed, the typical day of a student, a healthy adult or a frail elderly will be very different. Current publicly available datasets are therefore not sufficient. However, collecting new ones is a very costly and difficult task. While sensor readings can be easily collected after installing a sensor system, sensor data are not enough when considering learning and evaluation of algorithms. Ground-truth, i.e. the list of all activities performed by a human inhabitant as well as their start and end time, is needed, as most algorithms use a kind of supervised learning approach. Therefore, the inhabitant has to document her daily routine, which can result in a large investment from the user and is highly error-prone, as such annotations are done manually. Also, in e-Health and Assisted Living application especially, one might want to generate data corresponding to a given event such as a fall or a health-related problem, which is very complicated or even impossible for a human participant to simulate. For all these reasons, the idea of generating data using simulation suggests itself.

The main contribution of this paper is a general approach that can be used to generate data from any environment and sensor layout. Our approach simulates a Smart Home with all its sensors and inhabitant(s). A simulated human inhabitant is modelled by an agent living in a simulated, sensorised apartment. The simulated sensors record information as their real counterparts would record in the real world. In addition, the simulated inhabitant agent also tracks its activities. As it would be done in a real-world set-up, all data is stored in a database using exactly the same set-up as in a corresponding real-world data collection campaign.

Our approach combines a constraint-based planner generating an activity plan for a full day that is then executed by the simulated inhabitants within a spatially explicit agent-based sensor network simulation. That means the activity generation is independent of the sensor network simulation. Specific behaviour features are modelled without reference to the sensor readings; the simulated sensor set-up can be changed without regarding the simulated human planning and activity scheduling. This provides the flexibility to use very different sensor configurations as well as modelling individual behaviour traits.

2 RELATED WORK

Several Smart Home simulators have been developed for different purposes. The focus of those simulators may be either on data collection, as we consider in this contribution, or on visualization.

Synnott et al. (2015) distinguish two approaches for simulation: model-based and interactive. A third approach combining model-based and interactive simulation has risen recently, called the hybrid approach.

Model-based simulators, such as those described by Bouchard et al. (2010); Helal et al. (2012) use pre-defined models of the agent behaviour, describing different possible events and activities with their probability of occurrence, their order and their duration. They can be real-time or not. They enable the generation of data for extended periods of activities. Of course, the quality of the data relies heavily on the quality of the modelling. These simulators usually consider big granularities of event and activities, as it is complicated to model fine-grained events and activities. Finally, these approaches are usually entirely scripted. If the user wish to simulate several days (or more) of data, she needs to script each day independently. This is a major drawback for the use of these simulators in e-health applications as several weeks of data are usually necessary in order to detect long-term patterns. The use of agent-based simulation could reduce the burden of generating several days of data by modelling the rules and reasoning process an agent follows to decide about its daily activities instead of scripting each activity independently. This is the approach we propose here. Recently, Kamara-Esteban et al. (2017) presented a model-based simulator using an agent-based approach similar to our work. There are two major differences: first, the planning and execution are performed at the same time, making it difficult to create overall coherent plans that for example ensure that all mandatory tasks are performed during the simulated day; second, their system shows the same drawback as many other existing model-based simulators: the responsibility of creating days that are different enough to be realistic, falls upon the modeller providing the list of activities for each day.

Interactive approaches, such as proposed by Synnott et al. (2014); Ariani et al. (2013) assume a human controlling an avatar while it performs activities in the simulated environment. These approaches allow more coherent and realistic behaviour as there is a real human involved. However, they require near-real time interaction between the controlling human and the simulated environment. Therefore producing large amounts of data quickly is hardly possible. Thus, their usage is more restricted to the generation of data during a short periods, focusing on specific activities or situations.

Some recent work focused on bringing together the best of both worlds to create more complex activity patterns with less work for human modellers or controllers. This new approach is usually referred to as Hybrid Approach. Lundström et al. (2015) combine an interactive approach with probabilistic modelling of passive infra-red sensors. The interactive approach is used to generate data resulting from the interaction of the avatar with smart objects, while the model-based approach is used to generate data resulting from passive monitoring such as motion sensors in a room. Alshammari et al. (2017) combine interactive generation of data for small periods with model-based approach to aggregate these periods into full days.

3 SIMULATING A SMART HOME ENVIRONMENT

When generating a data set, an essential requirement is that the simulated sensor readings make sense in relation to each other. The result of the simulation must be a coherent set of sensor measurements from a simulated Smart Home. Therefore, the starting point is that sensor measurements are triggered by activities of simulated inhabitants, when the agent performs them moving through the apartment and interacting with furniture. Therefore, the coherence of the measurements originates in the coherence of the simulated activities.

3.1 Overview

We use spatially explicit agent-based simulation:

1. The sensor network is modelled as a simple, reactive multi-agent system. Sensors are agents that monitor connected objects. As discussed in Section 3.2, we abstract the actual measurement process to interaction with their environment.
2. Simulated inhabitants are agents that use high-level reasoning capabilities to determine a full activity plan for the day. This sequence of activities is then executed in the sensorised simulated apartment. Activities take place at particular locations in the apartment, movement between those locations is part of the execution.

Sensor agents can be placed anywhere on the map or can be connected to any object that can be used in an activity. When adding a new sensor in the simulation, the modeller simply needs to place it on the map and connect it to a room or an object for the sensor agent to be usable. Spaces such as rooms do not just have geometric extensions, but form a kind of navigation network, so that movement between activity locations is complete and coherent also triggering sensors that are associated with intermediate rooms.

The behaviour of the inhabitant agent is determined by the execution of an activity plan. Such a plan consists of a sequence of activities that are characterized by which inhabitant takes part in the activity, where the activity shall happen and what objects are used during that activity. While executing its activities, the inhabitant agent is in the corresponding room, uses associated objects and is monitored by object sensors or rooms sensors. We do not actually simulate the effect of an activity on the agents' environment beyond object usage. This makes the overall approach easily extendible with new types of activities. Figure 1 gives an overview of the basic model concepts and how they relate to each other. Figure 4 in Section 4.1 shows an example of a simulated apartment.

In the remainder of this section we will describe the different ingredients of our specific agent-based simulation. We want to achieve:

- Appropriate sensor models, including noise leading to erroneous measurements
- Sufficiently valid inhabitant model with respect to activities and activities' duration
- Sufficiently accurate overall (spatial) representation of the Smart Home setup

3.2 Simulating Sensors

A basic question is about the level of abstraction on which the simulated sensors are handled. In principle, one can identify two types of sensors: sensors associated with particular objects which change values when a user is interacting with it (e.g. pressure sensors on a chair); and sensors associated with a particular space which change value when the user is moves through the monitored space (e.g motion sensors).

One can identify three possible abstraction levels for modelling and simulating sensors:

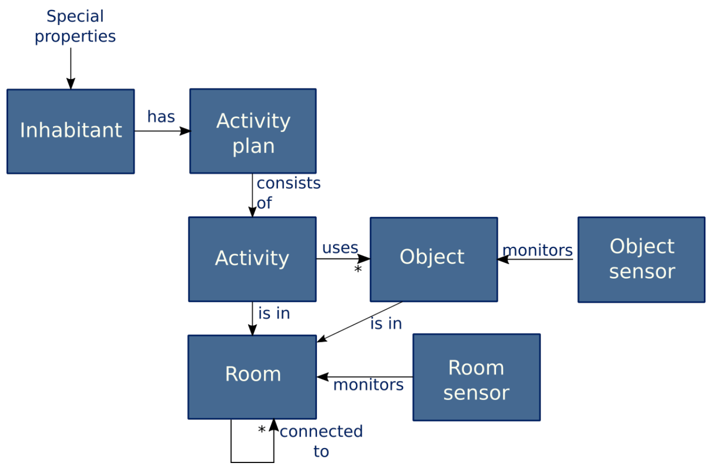
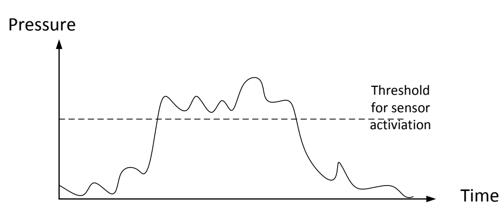
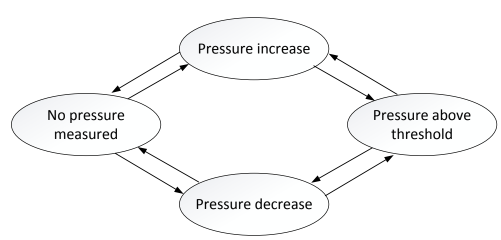


Figure 1: Overall conceptual framework describing the basic idea of model structure.

- **The simulated sensor reproduces how the real sensor works**, as detailed as possible, which means that it synthesizes the measurement process. The simulated sensor explicitly handles the sensed (numeric) values that another process needs to make sense of. Using this lowest level of abstraction results in the most precise and realistic sensor model. Figure 2a illustrates the value that the simulated sensor can provide. Another element interprets the sensor values e.g. based on a threshold. Simulated sensor errors and noise can be easily integrated.
- **The sensor value progression is represented by qualitative values** in a kind of state chart. The behaviour of the sensor is still explicitly modelled, yet now qualitatively by an abstracted version of the detailed value function above. Figure 2b gives an example for a pressure sensor. With this level of abstraction, there is no explicit simulation of the numeric pressure value, yet some abstract dynamics of the measured value can be expressed. Errors and delays in measurements can be introduced by manipulating the transitions.
- **The sensor sets its value based on interaction with the sensed entity instead of explicitly modelling the measuring process**. Information about when the sensor is triggered is taken directly from its environment: the pressure sensor would return "on", if the state of the object that it is attached to changes to being used.
  Instead of actually measuring, the sensor is represented as an agent that accesses the state of the associated object(s). Boolean trigger values can be also replaced by static numeric values. Instead of sending a boolean value, the sensor could send a predefined temperature, water consumption value, etc.



(a) Detailed sensor simulation



(b) Qualitative sensor value dynamics

Figure 2: Different granularity for sensor representation.

For the current version of the agent-based Smart Home simulation, we decided for the simplest, interaction-based concept. Sensors are modelled by a simple agent type that is associated with a room or with a particular environmental object such as a couch or an oven. The simulated human modifies the state of the objects during its activities: when entering the room, it registers its presence in the room, by sitting down at a couch, it modifies the state of the object, "using" it. The sensor agent accesses its associated environmental element and changes its output value accordingly. In that way, we modelled a variety of simple, un-intrusive sensors with which a Smart Home can be equipped that return boolean values (Light on, pressure measured...). Similarly, we model temperature sensors: the room manages a room temperature value that adapts when the human is for example using the room with specific activities, such as having a shower. The temperature sensor simply accesses this room property. In that way, we replace the actual simulation of a synthetic measurement process by interaction between sensor and environment object.

3.3 Activity Generation

3.3.1 Behaviour Modelling in General

As the behaviour of the simulated inhabitant shall mirror human behaviour in a believable way, the behaviour needs to be based on a high-level description of what happens during the simulated time within the Smart Home. In principle, three types of approaches for modelling agent behaviour can be identified (Klugl 2009):

1. **Behaviour describing approaches**: The behaviour model is described using scripts or finite state automata. The modelled agent does not reason, it simply follows the behaviour description given by the user. Clearly, the script may contain branching or reactions to particular perceptions. For example, if the agent perceives an obstacle, it turns into a random direction and tries to continue moving. Adaptation beyond direct reactions are difficult as they affect the overall (individual) script. Also, only limited complexity is possible, as every decision in reaction to some perception or internal state property of the agent needs to be explicitly included as such in the behaviour description.
2. **Behaviour configuring approaches**: The behaviour is generated based on a behaviour representation similar to a skeletal plan or goal tree. The agent possesses one of more goals and a set of conditionally applicable pre-defined plans for achieving those goals. When the agent has decided to achieve a particular goal, it may refine it to sub-goals or may select a plan schema that fits to the currently perceived situation. The selected plan is configured according to the situation, e.g. in terms of which objects to interact with.
3. **Behaviour generating approaches**: The agent uses first principle planning for generating the behaviour from using knowledge about "how the world works". That means the agent actually reasons about which effects a potential action has and under which circumstances an action can be taken for generating a plan of actions or activities. This architecture gives the most flexibility in conceiving behaviour flexibility and individualism. The modeller does not script the agent behaviour but gives more general information on how the world works and what preconditions need to be fulfilled, etc.

In principle, all three approaches could be used to model the behaviour of inhabitant agents. We selected the third one for several reasons: as we wanted to simulate complex daily plans for an extended period, scripting and configuring approaches would come with a too high effort to make the simulated behaviour coherent over a several days on the level of granularity that is necessary for triggering the sensors. In addition, the high variations that happen in humans' daily life between days would be very difficult and costly to handle by a modeller.

3.3.2 Activity Generation Concept

The daily activity schedule an agent is generated automatically through constraint based planning. The schedule is made up of activity instances, each of them including six attributes: activity name, start time, duration, participant, location, list of objects used.

All these attributes are generated by the planner using *a priori* knowledge provided by the modeller. This explicit knowledge contains information such as the Earliest Start Time (EST) and Latest Start Time (LST) as well as the minimum and maximum duration, the possible locations in which this activity could take place as well as all the possible set of objects that can be used while performing this activity. In addition to these, other constraints such as temporal or resource constraints are formulated that basically represent dependencies between activities. For instance, a temporal constraint `before(preparingFood eating [0 300])` can be used to describe that the activity `preparingFood` needs to be performed from 0 to 300 seconds (= 5 minutes) before the activity `eating`. A resource constraint would explicit the fact that the same chair cannot be used by two agents simultaneously or that one agent can not be in two different locations at the same time. Constraints are in principle different from rules, they just represent conditions that a viable solutions needs to fulfil, there is no direct consequence from such a condition.

Activities can be mandatory or optional. Mandatory activities are performed at least once per day while optional activities might not been performed at all. An arity can be defined for each activity to determine the minimum and maximum number of times that this activity should be performed during a day. For instance, activities such as `Preparing Breakfast` or `Preparing Lunch` are optional but can occur only once a day.

The overall strategy of the planner is to first select optional activities and organize mandatory and optional activities within a day satisfying all active constraints. Then, empty spots in the day are filled by activities called `in,<location>`, meaning that the occupant is in a specific location and may be using an object, but is not performing any specific activity. This step is important to improve the realism of the generated dataset and ensure that there are no wholes, yet still provide a reasonable, coherent activity schedule.

The choice of constraint-based planning with explicit modelling of the constraint network is motivated by the flexibility it offers. It is possible to formulate different scenarios and types of inhabitants, including inhabitants with health condition such as frailty or disease. It also makes easy to consider several occupants in a Smart Home by adding constraints to express the fact that an object cannot be used by two different agents at the same time or the fact that certain activities (e.g. eating) cannot be performed alone. It is also possible to create daily plans depending on dynamic parameters such as the day of the week, the current season, etc.

3.4 Bringing It Together

Figure 3 presents the overall framework (also indicating the actual systems that have been used for implementation) illustrating the communication between the simulator and the planner. At the beginning of each simulated day, the simulator requests a new plan from the planner through the interface, providing any useful parameter (such as the current day, the current season or for which particular agent). The interface updates the planner's knowledge database with the information provided by the simulator and requests an activity schedule from the planner. Finally, it transmits this plan to the simulator.

To plan the agent's daily activities, we used *SpiderPlan* (K[Pleaseinsertintopreamble]ckemann 2016), which is a constraint-based planner that allows the inclusion of different types of knowledge such as goals, temporal constraints, reusable resources and interaction constraints (Köckemann et al. 2015). We formulated a knowledge base suitable to generate activity plans with more than 80 potentially relevant constraints describing "how a day works". For generating this initial constraint base, we did not use data that we collected as mentioned in Section 4.1, but grounded them on common sense. In later versions of this knowledge base, data will be used to ground the assumptions done in a better way.

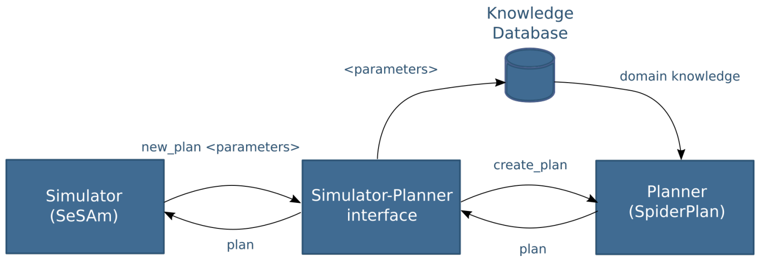


Figure 3: Communication between the different systems involved: simulator and planner.

The simulator was implemented using SeSAm (www.simsesam.org), a fast prototyping platform in which agent behaviour is formulated using a kind of activity diagrams that can also easily handle explicitly spatial simulations.

After the plan is received by the agent in the simulator, the agent starts to execute the first activity in the simulated home. For that aim, it determines, whether it is in the room in which the activity shall take place. If not, it moves to that room. Navigation is based on a simple navigation network generated from rooms that are either adjacent or connected via a door. When an agent enters a room, it "checks-in" into that room which triggers a state change of the room and consequently enables a sensor connected to that room to change its measured and communicated value. When the agent arrives in the room in which the activity shall take place, it "tells" the appropriate objects there that it uses them with the corresponding consequences for connected sensors. We use a time-stepped simulator, that means simulated time explicitly passes. When the activity duration is exceeded, the agent checks whether there is a further activity in its schedule and continues with that.

4 EVALUATION

As the coherence of data produced by heterogeneous simulated sensors over time is achieved by simulating realistic human behaviour, we focus our evaluation on the believability of the generated daily activity plans. We used questionnaires to be filled by humans indicating in how far the generated activity plans appear to be realistic.

4.1 Evaluation Scenario

The basic scenario corresponds to a real apartment inhabited by one person in which we also collected data. This data-collection has been done within the E-care@Home project. All software and datasets, including the simulated ones, created during this project will be made available (http://www.ecareathome.se/open-source-software/). Figure 4 presents the layout of the apartment including sensors. Object sensors are mostly pressure sensors on chairs, sofa and sensors that detect whether the TV or the oven are on. Motion sensors can be found in living room, the bed room and kitchen. The sensor in the bathroom is a temperature sensor which just gives information whether there is person in the bathroom or not. The only differences in sensor set-up between the simulated and real apartment. are that the latter has no sensors in bedroom and bathroom.

Data from the real apartment has been gathered prior to this evaluation. Two days (from 00:00 to 20:00, the 4 remaining hours being needed to recharge the sensors) of data were available for evaluating the simulated data set and activity plans against real data. The inhabitant of the apartment annotated the sensor measurements recording start and end time of high-level activities.

4.2 Planner Configuration

The planner has been configured with activities usually performed by the real Smart Home inhabitant, when staying at home most of the day.

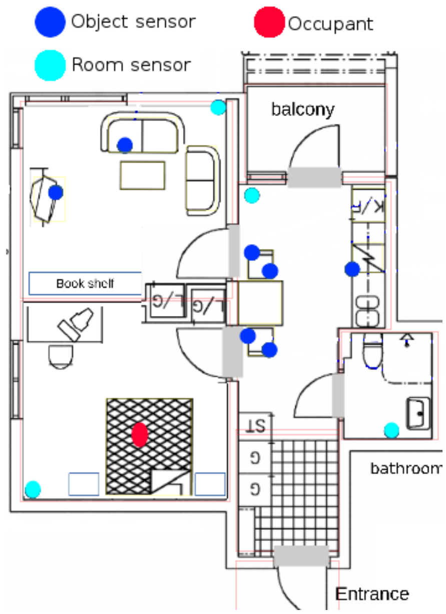


Figure 4: The simulated apartment. The sensor layout is copying the real installation.

We computed plans with a total of 8 mandatory activities – governed by up to 32 different active constraints – and between 5 and 12 optional activities – each optional activity fulfilling 3 or 4 additional constraints. The list of mandatory and optional activities and relevant temporal constraints is given in Table 1. Resource constraints were used to specify that the user cannot perform two different activities at the same time but are not presented in this table because of space limitation.

4.3 Analysis of the Generated Data

Evaluation needs to be based on two different aspects: the plausibility of the generated plans in general and the correctness of the generated data with regards to the generated plans and the real dataset.

4.3.1 Analysis of the Generated Daily Plans

To determine the plausibility of our generated activity schedules, we used face validation in which subjects were asked to systematically judge the plausibility of full day activity schedules. A total of 19 participants answered to an online questionnaire. This questionnaire provided 8 examples of daily activity plan, of which 4 were generated by our simulator and 2 were created by the real inhabitant of the real apartment, 2 additional plans were taken from the ARAS data set (Alemdar et al. 2013). Example of timelines as shown in our survey are given in Figure 5.

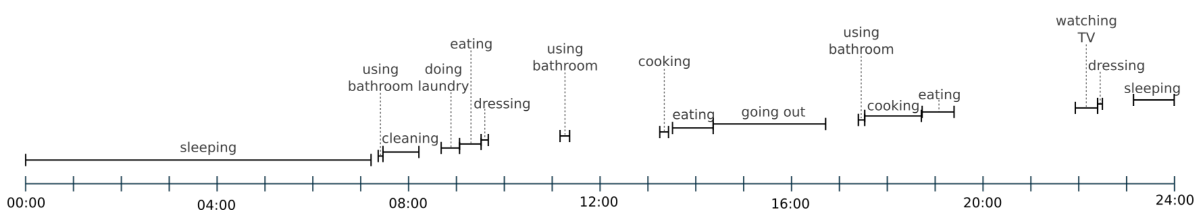
The 8 timelines were presented in a random order to the participant, who was then asked to answer two questions for each of them: 1) the participant had to grade the plausibility of the timeline on a scale from 1 to 7 (1 being not plausible at all and 7 being very plausible) and 2) the participant had to indicate whether to her opinion the plan has been generated by a simulator or a real person.

Figure 6 summarizes the results of the first question. The average of the grades for the simulated days is 4.7 while the average for the real days is 3.79. This indicates a tendency that the users found the simulated days more plausible than the real days. However, the realism of one real day was specifically rated much lower than the other. If we exclude this day from the comparison, the average becomes 4.31, which is comparable to the simulated days. When looking into the details, one can see that longer and more structured activities during the day are rated as more plausible. There are two potential reasons for the low believability of real timelines: 1) the participants have biased expectation of what a normal day should look like and don't realize how often users can change activities ; or 2) annotating activities is an extremely challenging task and the resulting timelines might contain errors and imprecisions.

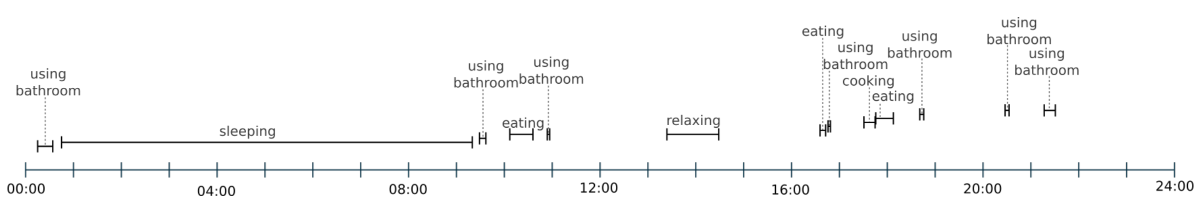
Table 1: List of mandatory (marked with *) and optional activities and list of additional temporal constraints.

Activity	**[EST-LST]**	**Duration**
sleepingMorning*	[00:00-00:00]	[6-7] hours
sleepingEvening*	[21:00-23:00]	[1-3] hours
eatingBreakfast*	[07:00-11:00]	[15-30] min.
eatingLunch*	[11:30-14:00]	[20-60] min.
eatingDinner*	[18:00-21:00]	[20-60] min.
inBathroom*	[06:00-23:00]	[3-30] min.
dressingMorning*	[06:00-12:00]	[2-10] min.
dressingEvening*	[19:00-23:00]	[2-10] min.
prep.Breakfast	[06:00-11:00]	[5-30] min.
prep.Lunch	[06:00-14:00]	[5-120] min.
prep.Dinner	[06:00-21:00]	[5-120] min.
laundry	[06:00-23:00]	[15-30] min.
watchingTV	[06:00-23:00]	[20-120] min.
goingOut	[06:00-23:00]	[30-240] min.
relaxing	[06:00-23:00]	[60-240] min.
cleaning	[06:00-23:00]	[30-200] min.

Constraint
`before(preparingX, eatingX, [1, 300])`
`before(eatingBreakfast, eatingLunch, [14400, inf])`
`before(eatingLunch, eatingDinner, [14400, inf])`
`before(dressingMorning, goingOut, [1, inf])`
`before(goingOut, dressingEvening, [1, inf])`
`before(eatingDinner, dressingEvening, [1, inf])`



(a) Timeline of activities generated by the simulator



(b) Timeline of activities as captured by a human user

Figure 5: Daily activities as presented in our questionnaire-based plausibility check.

The first case would indicate that it is important to consider more fragmented activities when simulating data. The second case would highlight even more the use for simulated data as they are more reliable than self-annotated ones. However, more experimentation needs to be done to make reliable statements.

Figure 7 shows the percentage of participants who believed that the timeline was created by a simulator, a real person or were unable to choose.

The results are coherent with the trend identified previously, as the 46% of the participants declared that the real days were produced by simulators against 32% for the simulated days. Once again, we can see that the real day 4 is an outlier as only 16% of the participants declared that it has been created by a real person.

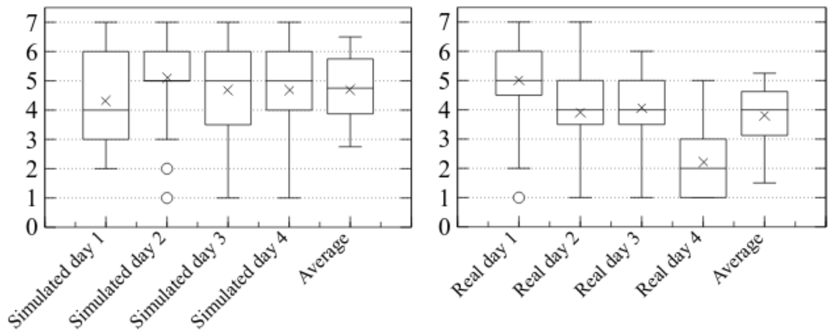


Figure 6: Grading of the activity timelines.

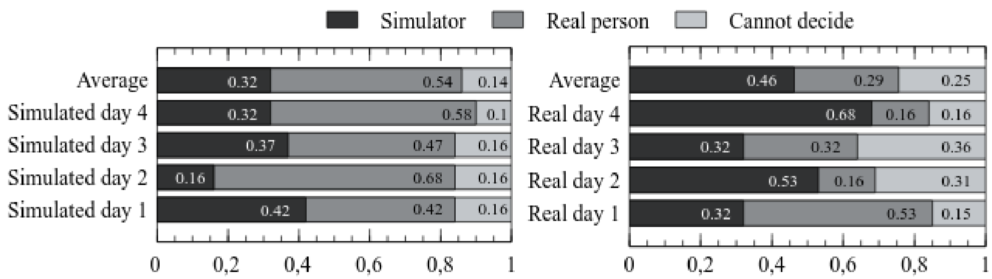


Figure 7: Percentages of participants selecting "Simulator", "Real person" or "Cannot decide".

This evaluation highlights two interesting points. First, our planner seems to be able to produce believable activity timelines. Second, there seems to be a discrepancy between the expectation of a real day looks like when looking at a complete day and the actually recorded timeline of a real day, whether this difference is due to real activity fragmentation or to errors during activities annotation is unclear.

4.3.2 Analysis of the Generated Sensor Data

We generated 4 days of data with our simulator and computed the time during which each sensor is activated (Figure 8) and compared it to the reference dataset (produced by a real human user). Figure 8 shows the result of this comparison.

We can first note that the motion sensors are activated much more often in the simulator than in the real home. This can be explained by the fact that our simulation does not allow yet the sporadic activation of motion sensors as it is usually the case with real sensors. In our case, a motion sensor will be activated for as long as the agent is in the room, while in a real Smart Home, the motion sensors can be inert when the agent is sitting on the couch or sleeping. The important difference we observe on the chair pressure sensors can be explained by two facts: the activity `relaxing` can take place in the kitchen in the simulation (as stated possible by the real inhabitant during the modelling) while it mostly took place in the bedroom or in the living room in the real environment; the activity `cooking` can expect the user to sit on a chair while cooking (as stated possible once more by the real inhabitant) though the occupant of the real home was standing while cooking during these two days of recording. These observations however are rather anecdotal, for a profound analysis more simulation runs need to be statistically compared to more real data sets to make statements about comparable distributions, etc.

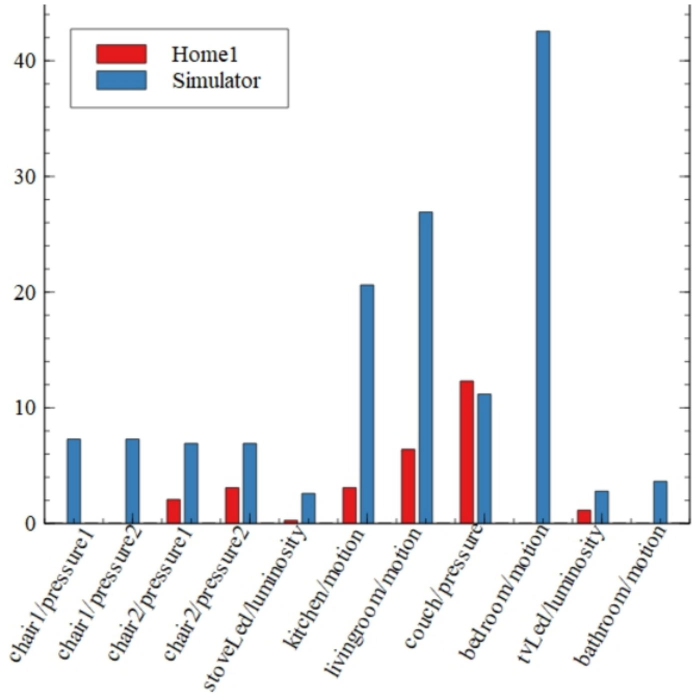


Figure 8: The average time during which is sensor is activated (in % of the day).

5 DISCUSSION AND CONCLUSION

In this paper, we presented a framework to generate sensor data from a simulated Smart Home, using a flexible agent-based simulation tool and constraint-based planning. The data generated can be used in order to test or train algorithms that are then directly usable in real-world applications. Our evaluation showed that the activity plans generated by the simulator show some plausibility. However, the comparison of these plans also revealed problems with real data sets.

An interesting aspect relates to reactive behaviour. As we separate activity planning from execution in simulation, the activity planning may not have foreseen the situation in which the agent encounters itself completely and the agent would need to spontaneously adapt the plan for the rest of the day. In the current version, the simulated agent would need to call the constraint planner with the new situation and wait for a new plan. Handling the plan changes in the simulator directly would require a tighter coupling between planning and simulation – loosing the flexibility and clear software architecture.

In its current version, our approach cannot handle parallel and interleaved activities. A simulated inhabitant is required to be at one location to start an activity and to stay there for the entire duration of this activity. This limitation needs to be removed in future versions by refining our conceptual model. Additional future work will consider the use of probabilistic reasoning to decide of the daily plan.

ACKNOWLEDGEMENT

This work is supported by the distributed environment Ecare@Home funded by the Swedish Knowledge Foundation, 2015-2019.

REFERENCES

Alemdar, H., H. Ertan, O. D. Incel, and C. Ersoy. 2013. "ARAS Human Activity Datasets In Multiple Homes With Multiple Residents". In *Proceedings Of The 7th International Conference On Pervasive Computing Technologies For Healthcare*, Pervasivehealth '13, 232–235. ICST, Brussels, Belgium.

Alshammari, N., T. Alshammari, M. Sedky, J. Champion, and C. Bauer. 2017. "Openshs: Open Smart Home Simulator". *Sensors* 17(5):1003.

Ariani, A., S. J. Redmond, D. Chang, and N. H. Lovell. 2013. "Simulation Of A Smart Home Environment". In *3rd International Conference On Instrumentation, Communications, Information Technology, And Biomedical Engineering (ICICI-BME)*, 27–32. IEEE.

Bouchard, K., A. Ajroud, B. Bouchard, and A. Bouzouane. 2010. "SIMACT: A 3D Open Source Smart Home Simulator For Activity Recognition". In *Advances In Computer Science And Information Technology*, 524–533. Miyazaki, Japan.

Cook, D. J., and M. Schmitter-Edgecombe. 2009. "Assessing The Quality Of Activities In A Smart Environment". *Methods Of Information In Medicine* 48(5):480–485.

Helal, A., K. Cho, W. Lee, Y. Sung, J. Lee, and E. Kim. 2012. "3D Modeling And Simulation Of Human Activities In Smart Spaces". In *2012 9th International Conference On Ubiquitous Intelligence & Computing And 9th International Conference On Autonomic & Trusted Computing (UIC/ATC)*, 112–119. IEEE.

Kamara-Esteban, O., G. Azkune, A. Pijoan, C. E. Borges, A. Alonso-Vicario, and D. LÓPez-De-IpiÑA. 2017. "MASSHA: An Agent-Based Approach For Human Activity Simulation In Intelligent Environments". *Pervasive And Mobile Computing* 40:279–300.

Klugl, F. 2009. "Agent-Based Simulation Engineering". Habilitation Thesis, University Of WÜRzburg.

Köckemann, U. 2016. *Constraint-Based Methods For Human-Aware Planning*. Ph. D. thesis, Örebro University.

Köckemann, U., F. Pecora, and L. Karlsson. 2015. "Inferring Context And Goals For Online Human-Aware Planning". In *Proceedings Of The 2015 IEEE 27th International Conference On Tools With Artificial Intelligence (ICTAI)*, ICTAI '15, 550–557. Washington, DC, USA: IEEE Computer Society.

Lundström, J., J. Synnott, E. Järpe, and C. D. Nugent. 2015. "Smart Home Simulation Using Avatar Control And Probabilistic Sampling". In *2015 IEEE International Conference On Pervasive Computing And Communication Workshops (Percom Workshops)*, 336–341. IEEE.

Synnott, J., L. Chen, C. Nugent, and G. Moore. 2014. "The Creation Of Simulated Activity Datasets Using A Graphical Intelligent Environment Simulation Tool". In *36th Annual International Conference On Engineering In Medicine And Biology Society (EMBC)*, 4143–4146. IEEE.

Synnott, J., C. Nugent, and P. Jeffers. 2015. "Simulation Of Smart Home Activity Datasets". *Sensors* 15(6):14162–14179.

Tapia, E. M., S. S. Intille, and K. Larson. 2004. "Activity Recognition In The Home Using Simple And Ubiquitous Sensors". In *Pervasive Computing*, edited by A. Ferscha and F. Mattern, 158–175. Berlin, Heidelberg: Springer Berlin Heidelberg.

AUTHORS BIOGRAPHIES

JENNIFER RENOUX is a post-doctoral researcher in the Center for Applied Autonomous Sensor Systems at the Örebro University, Sweden since 2016. Sheholds a Ph.D. in Artificial Intelligence and Applications from the University of Caen Normandy, France. Her research interests include planning and reasoning under uncertainty, especially epistemic planning. Her email address is jennifer.renoux@oru.se.

FRANZISKA KLÜGL is a Full Professor in Information Technology at the Örebro University (Sweden) since 2011. She holds an Habilitation and PhD in Computer Science from the University of Würzburg, Germany. She is interested in issues related to systematic development of agent-based simulation models including specific approaches to specification, implementation, documentation and validation. Her email address is fkluegl@acm.org.