Automatic Daily Activity Schedule Planning for Simulating Smart House with Elderly People Living Alone

Can Jiang[1(✉)] and Akira Mita[2]

[1] Graduate School of Science and Technology, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa, Japan
canjiang@keio.jp

[2] Department of System Design Engineering, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa, Japan
Mita@keio.jp

Abstract. A simulation tool that supports developers to build scenarios automatically in multiple simulation platforms is proposed. As an essential part of this simulator, this study proposed an activity schedule generator to mimic the daily life of elderly people living alone. This generator outperforms existing methods of activity schedule planning in three aspects: 1) it is adaptive to the layout of a simulated smart house; 2) there is no unspecified time in the timeline of generated schedules; and 3) it generates stable, but not tedious schedules for a number of days. A real-time location data generator is proposed to convert generated schedules to simulated real-time location data of the resident, and a proposed interface converts these simulated location data to simulated records of virtual passive infrared (PIR) sensors, which can be used to optimize placement of PIR sensors in a smart house.

Keywords: Elderly people living alone · Smart home simulator · Activity of daily living · Motivation · Automatic scenario generation

1 Introduction

The elderly population is increasing worldwide. An estimated 617.1 million people are aged 65 and over in 2015, and this number is projected to increase to 1 billion in 2030, and 1.6 billion in 2050 [1]. More than 20% of men and 40% of women aged 65 and older chose an independent lifestyle in many countries [2]. Pimouguet et al. [3] indicated living alone shortened life expectancy by 0.6 years for elderly people. Elderly individuals living alone would benefit from specialized care, but a shortage in the global workforce of aged-care workers [4] has made this difficult.

Under these conditions, smart houses with a sensor network and domestic robots have been built to address the aged-care worker shortage. The sensor networks provide real-time health monitoring [5] and a means of detecting emergencies [6], while mobile domestic robots provide location-based support [7] and services [8] for residents. To ensure the effectiveness of the sensor networks and robots, real test beds were built to conduct experiments for collecting data. However, building a test bed is expensive, and

M. Jmaiel et al. (Eds.): ICOST 2020, LNCS 12157, pp. 171–183, 2020.
https://doi.org/10.1007/978-3-030-51517-1_14

simulations are necessary for smart house developers to test and verify their ideas before building a real one.

Developers typically conduct simulations using the following three steps. (1) Manually create a simulation scenario by first building a house and resident body models and defining the activity schedules and movement routes of the virtual resident or controlling the virtual resident manually. (2) Place virtual sensors, devices, or robots to record data and/or operation performances. (3) Analyze recorded data or operation performances and evaluate simulation design. As a typical simulation constructed in step (1) requires a lot of time, developers can only prepare a limited number of scenarios. Moreover, the developers may use multiple simulators for different purposes, e.g., using CST Microwave Studio to test the communication of a wireless sensor network, OpenSHS [9] to collect virtual sensor records for sensor arrangement optimization, and Stage [10] to plan the operation policies of mobile robots. When the developers use another simulator, they must repeat steps (1) even if they use the same simulation scenario.

We propose a simulation tool that provides diverse simulation scenarios and can support smart house developers to complete step (1) automatically in multiple simulation platforms [11]. This simulator consists of generators and interfaces as show in Fig. 1. The proposed generators produce diverse information such as indoor spatial attributes and resident travel patterns. This information is used to create a scenario that can run on different simulation platforms through various interfaces. We proposed a spatial attribute generator [12] and travel pattern generator [11], and used two interfaces [11] to transfer the data generated by them to models and virtual sensor records of the simulators.

As an essential part of our simulator, we propose an activity schedule generator. With generated travel patterns, these schedules are converted to simulated real-time location data, which can be used in simulations with interfaces. The rest of this paper is organized as follows. In Sect. 2, we review related work of daily activity schedule generation. Section 3 describes the methodology to generate activity schedules. Section 4 details the performance of this generator. Section 5 introduces how the generated data can be used in simulations.

2 Related Works

A number of scholars generated daily activity schedules as intermediate results to generate sensor records in a virtual smart house, which are essential for simulations.

Renoux et al. [13] generated activity schedules with a constraint-based planning method. The constraints include that the start time and duration of each activity are over reasonable intervals, and a number of activities need to be performed within certain time intervals before their corresponding activities, e.g., preparing lunch for 0 to 5 min before having lunch.

Bouchard et al. [14] generated activity schedules using behavior trees (BTs) as intermediate results to generate the simulated evolution of signal strength between RFID readers and tags. However, designing BTs is complicate, and the authors only showed an example of generating the schedule for making coffee or tea.

Alshammari et al. [9] replicated and modified schedules originally designed by humans. The methods of modification include combining two samples of original schedules and changing the start and end time of activities. The activity schedules correspond to virtual binary sensor records, thus, a large number of records are generated simultaneously. This method is simple, but generated schedules may have high similarity.

Mshali et al. [15] generated long-term activity schedules using a Markov model, and five transition matrices associated to different periods of a day were designed. The authors also proposed an adaptive and context-aware algorithm for monitoring the daily activities of elderly and dependent persons, and generated schedules were used to test the algorithm in simulations.

Lee et al. [16] generated activity schedules with a motivation-driven method. A motivation value (MV) represents the desire of a virtual agent to perform a class of activities with the agent performing an activity when its corresponding MV reaches its threshold. Motivations are classified by levels; if two MVs reach their thresholds at the same time, the agent will perform the activity that corresponds to the higher-level motivation. This method has sufficient potential for improvement if the mechanism of evolution of the MVs is designed carefully.

3 Activity Schedule Generation

3.1 Problem Statement

To build our activity schedule generator module, we need to improve upon the methods mentioned in Sect. 2 by addressing the following issues. 1) The list of activities that can be performed by a virtual resident is determined by the layout of simulated house, e.g., the resident can only watch TV if a TV is in the house. The above methods are for a determined layout with a fixed activity list. As our simulator contributes to provide diverse simulation scenarios by producing diverse layouts, we need a method that can process dynamic activity lists. 2) The above methods generate schedules whose timelines include unspecified times between the end time of an activity and the start time of the next one. Where the resident has been and what he/she has done during the unspecified time are undetermined, thus, generated sensors records did not cover entire days. 3) Most of the above methods generated schedules for one day or less, but long-term activity schedules are required for our simulation.

We developed a motivation-driven method on the basis of that presented in the reviewed study [16] to build our activity schedule generator. An MV represent a resident's desire to perform an activity sequence (AS). While performing the activity is dependent on its MV reaching its threshold in [16], in our method, the MVs are used to determine the probability distribution ($\boldsymbol{P}$) of sampling the next AS. The evolution of the MVs is adaptive to the input indoor spatial data and resident's profile. The input data represent a layout that determines what AS can be performed, thus, this adaptive evolution mechanism addresses issue 1). The profile represents a resident's tendencies to activities, which is quantified by durations ($\boldsymbol{D}$), periods ($\boldsymbol{T}$) and frequencies ($\boldsymbol{f}$) of an AS. We need to design an evolution mechanism and initialize the MVs carefully to

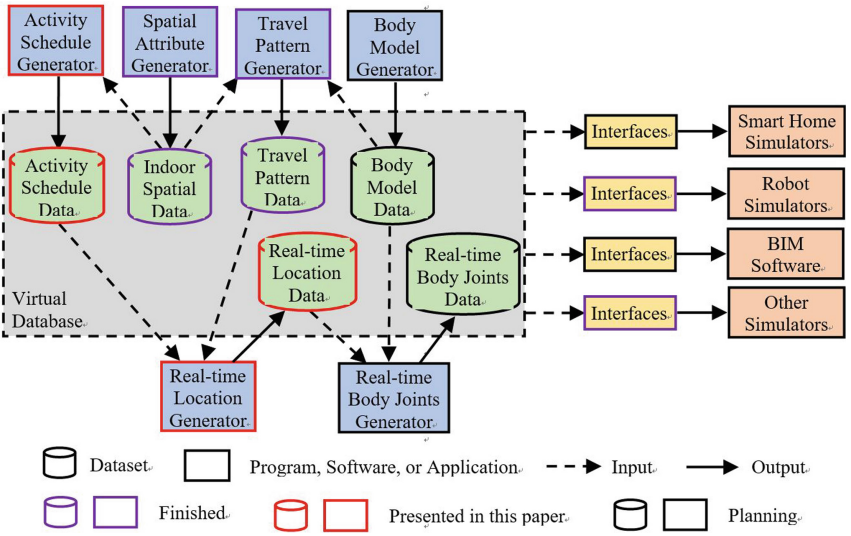

Fig. 1. General framework for building the simulation tool [11].

generate stable, but not tedious, activity schedules in the long term that will address issue 3). We can address issue 2) by taking into account more activities. The studies mentioned in Sect. 2 took into account a limited number of activities, implying that these activities occupy the entire timeline, which is unrealistic.

3.2 Model of AS, *MV*, Resident Profile, and *P*

Mapping the Relationship Between AS and *MV*. A resident performs activities on the basis of motivation, in which MVs quantify the degree of motivation. When an MV is high, the resident may perform a sequence of activities to satisfy the motivation, e.g., if the value of hunger motivation is high, he/she will cook and eat.
We determined that a resident has 13 motivations at most, which correspond to 13 MV_i-AS_i pairs: MV_1: wash and brush teeth => sleep (at night) => wash and brush teeth, MV_2: sleep (at noon), MV_3: take food => cook => take tableware => eat, MV_4: take a bath => get dressed => put clothes in wash machine, MV_5: get dressed => go out => get dressed, MV_6: go to toilet (short duration), MV_7: go to toilet (long duration), MV_8: watch TV, MV_9: read, MV_{10}: clean, MV_{11}: take clothes out of washing machine, MV_{12}: wander, and MV_{13}: relax. An AS consists of one or more activities, and activities in the AS are performed in order without lag to satisfy the corresponding motivation and decrease the corresponding MV_i. MV_i determines the possibility of performing ith AS, P_i.

AS and *MV* Are Adaptive to the Layout. The actual composition of an AS is also adaptive to the layout like the evolution of MVs. An activity in an AS will be omitted if its corresponding places are not in the layouts The mapping relationship of all activities

and places is shown in Table 1, e.g., if the kitchen stove, refrigerator, and cupboard are not in the house, AS_3 will omit the procedure take food => cook => take tableware, which implies the resident eats food prepared by someone out of the house in this case. If all activities of an AS can not be performed because of the layout, its MV_i will always be 0.

Resident's Personal Profile. To keep generated schedules diverse and reasonable, parameters corresponding to the evolution of MVs should depend on the resident's profile. The profile represents a resident's tendencies to satisfy different motivations, which is determined by sampling $\boldsymbol{D}$, $\boldsymbol{T}$, and $\boldsymbol{f}$ of an AS performed over reasonable intervals. *Di* means the duration of the resident performing the *i*th AS in a period (T_i), e.g., D_8 means how long the resident performed the activity "watch TV" per day on average if $T_8 = 1$ day. T_i and f_i mean the period and frequency of performing the *i*th AS, respectively, e.g., T_{12} means how many days between two instances of the resident's wandering and f_6 means how many times the resident performed the activity "go to toilet (short duration)" per day on average, where $T_i \times f_i = 1$.

Mechanism of MVs Evolution. MVs quantify the motivations to perform activities, MV_i usually decreases when the *i*th AS is performed, increases when other ASs are performed, and remains unchanged in special cases. The values of $\boldsymbol{D}$, $\boldsymbol{T}$, and $\boldsymbol{f}$ determine the speed of the increase and decrease of MVs. The rules below show the evolution of *MV*s, where *Rules* 1.1) and 1.2) indicate the situations when MV_i remains unchanged, 2.1) to 2.4) show the mechanism of MVs increasing, and 3.1) to 3.4) show the mechanism of them decreasing.

Rule 1.1) MV_i is always 0 if *i*th AS can not be performed.
Rule 1.2) MV_{13} = 1.02Norm_MV in any case, where Norm_MV is a constant.

Note: MV_{13} corresponds to relax. This rule means the resident is performing the activity "relax" when all other MVs are low. Setting MV_{13} as a constant keeps MVs stable in the long term.

Rule 2.1) Ways of MVs increasing include linear and step functional increasing.
Rule 2.2) If MV_i is not fixed and $i \neq 7$ or 11, MV_i increases linearly when the *i*th AS is not performed. The increment is determined by Eq. (1),

$$MV_i(t + \Delta t) = \begin{cases} MV_i(t) + \omega_i \Delta t + \varepsilon, & \text{if the resident is not sleeping.} \\ MV_i(t) + 0.1\omega_i \Delta t + \varepsilon, & \text{if the resident is sleeping, } \textit{and } i = 3 \text{ or } 6. \\ MV_i(t) + \varepsilon, & \text{if the resident is sleeping, and } i \neq 3 \text{ and } 6. \end{cases} \tag{1}$$

where ω_i is an increasing rate and ε is random noise.

Note: When the resident is sleeping, the increasing rates of the MVs of "eat" and "go to toilet (short duration)" decrease to 10%, rates of other MVs decrease to 0.

Table 1. Mapping relationship between activities and places.

Activity	Place
Sleep (at night) Sleep (at noon) Relax	Bed
Wash and brush teeth Take a bath	Bathroom
Take food	Refrigerator
Take tableware	Cupboard
Take food Take tableware Cook	Kitchen stove
Eat	Dining table-chair set
Put clothes in washing machine Take clothes out of washing machine	Washing machine
Go to toilet (short duration) Got to toilet (long duration)	Toilet
Watch TV Relax	Sofa-TV set
Relax Read	Writing desk-chair set
Get dressed	Wardrobe
Eat Go out	Entrance
Clean	Trash bin
Wander	None

Rule 2.3) Following the principle that MV_i should be generally unchanged after one period in an ideal case, ω_i can be determined by $\boldsymbol{D}$, $\boldsymbol{T}$, and $\boldsymbol{f}$.

Note: e.g., for the 8th AS, watching TV, assuming that the resident watches TV for 4 h (D_8), and sleeps 8 h ($D_1 + D_2$) per day (T_8 = 24 h), ω_8 is determined by Eq. (2),

$$\omega_8 = \frac{\text{Norm\_MV}}{[T_8 - (D_1 + D_2) - D_8]}. \tag{2}$$

which means ω_8 should increase by Norm_MV in the remaining 12 h, while it decreases by Norm_MV during the 4 h (D_8).

Rule 2.4) MV_i increases step by step if i = 7 or 11. For the 7th AS, "going to toilet (long duration)", MV_7 increases by Norm_MV/(3 × T_7) 2.5 h after the resident starts eating. For the 11th AS, "taking clothes out of the washing machine", MV_{11} increases by Norm_MV 1 h after the resident puts their clothe into the washing machine.

Rule 3.1) MV_i will decrease if the ith AS has been performed except if i = 13.

Rule 3.2) The decrease of MV_i depends on the T_i and actual duration, AT. It is defined by Eq. (3) except if the AS is eating breakfast, the decrease is ($2AT/3T_i$) × Norm_MV.

$$MV_i(t+AT) = MV_i(t) - \frac{AT}{T_i}\text{Norm\_MV}. \tag{3}$$

Rule 3.3) AT is related to T_i, AT is sampled from [$0.97T_i$, $1.03T_i$] for i = 1, from [$0.4T_i$, $0.9T_i$] for i = 5, from [$0.3T_i$, $0.7T_i$] for i = 8 or 9, and from [$0.95T_i$, $1.05T_i$] for other cases.

Rule 3.4) The resident may perform the activities "eat" and "go to toilet" outside. When he/she is going out, if MV_3, MV_6 or MV_7 reach Norm_MV, and there is sufficient time to perform the corresponding AS, this MV decreases as the AS is performed.

Initialization of MVs. MVs should be initialized before evolution, which can be achieved by determining when each AS will be performed for first time. The time when the ith AS is first performed is approximately equal to the time when MV_i first reaches Norm_MV. We sample the initial time from 9:30 PM of one day to 1:00 AM of the next day, and the resident is going to sleep. MV_0 thus is Norm_MV, as D_i and ω_i is known, other initial MVs can be calculated with Eq. (1), e.g., assuming that D_1 = 8 h, and the resident will eat breakfast 1 h after waking up, initial MV_3 is calculated by Eq. (4).

$$MV_3 = \text{Norm\_MV} - 0.1\omega_3 \times 8\,\text{h} - \omega_3 \times 1\,\text{h}. \tag{4}$$

To keep the generated schedule stable in the long term, we need to avoid two MVs whose ASs require long durations to reach Norm_MV at the same time.

Relationship Between MV and P. The possibility of performing the ith AS depends on the motivation value, MV_i, as shown in Eq. (5)

$$P_i = \frac{\exp[\max(0, MV_i - 0.98\text{Norm\_MV})]}{\sum_{j=1}^{13}\exp[\max(0, MV_j - 0.98\text{Norm\_MV})]}. \tag{5}$$

3.3 Implementation

We wrote a Python3 program to achieve activity schedule generation. A sample of the indoor spatial attribute data (***Spatial_data***) and total generation duration (Total_Dur) were input into the program, and it returns a resident's daily activity schedule during the Total_Dur. The pseudocode of the program is shown below, where constants, variables, and variable vectors are in regular, italic, and bold italic styles, respectively.

```
program Schedule_Generation(Spatial_data, Total_Dur):
1 AS_canbe_perform := Process_Input(Spatial_data)
2 T, D, f := Generate_Resident_Profile()
3 ω := Calculate_Increse_rate(T, D, f)
4 MV, Init_time := Initialize_MVs&time(T, D, f)
5 Current_ASnum, time := 1, Init_time
6 ASnum_list, time_list := [], []
7 while time < Init_time + Total_Dur do:
8   AT := Determine_actual_duration(Current_ASnum, T)
9   MV := Update_MV(Current_ASnum, time, MV, ω, T, AT)
10   time := time + AT
11   Next_ASnum = Sample_next_ActSeq(MV)
12   If Next_ASnum != Current_ASnum do:
13     ASnum_list.append(Next_ASnum)
14     time_list.append(time)
15     Current_ASnum := Next_ASnum
16 Activity_Schedule = Post_Process(ASnum_list, time_list, Spatial_
data)
17 return Activity_Schedule
```

The program first processes the input spatial attribute data, analyzes the layout, and determines what ASs can be performed in the house in Line 1. The resident's profile is determined by sampling $\boldsymbol{D}$, $\boldsymbol{T}$, and $\boldsymbol{f}$ in Line 2. $\boldsymbol{\omega}$ is calculated in Line 3 in accordance with *Rule 2.3)*. The original $\boldsymbol{MV}$ and the start time of the schedule generation are determined in Line 4. In Line 5, we assume the resident performs AS_1 at the beginning of the generation, and the variable *Time* records the current time. Two lists are created in Line 6, ***ASnum_list*** and ***time_list***, which will record the number of all performed ASs and their start times chronologically, respectively. From Lines 7 to 15, the program determines the *AT* of performing each AS with *Rule 3.3)*, updates $\boldsymbol{MV}$ using the other rules, samples the next performed AS with Eq. (5), and stores the number of performed ASs and their start times in ***ASnum_list*** and ***time_list***, respectively. The program converts these two lists into an activity schedule in Line 17. The schedule indicates the start times of all activities performed.

4 Performance of the Generator

We input indoor spatial data generated by the spatial attribute generator into the activity schedule generator, which then produces diverse activity schedule data. For example, a sample of spatial data whose layout is shown in Fig. 2 is input into the generator. As the places "desk" and "washing machine" do not exist in the house, AS_9 (read) and AS_{11} (take clothes) can not be performed. The activity schedule generator then determines the resident's profiles and generates their corresponding schedules. Two example schedules are shown in Fig. 3. Figure 3a) shows a schedule for a resident who sleeps around noon, goes out, watches TV, and takes a bath every day, while Fig. 3b)

shows a schedule for who does not sleep around noon, watches TV, and takes a bath every day, but only goes out every four days.

We also tested the performance of our generator on PC with an Intel(R) core(TM) i7-8550U @1.80-GHz CPU. The generator ran 100 times in 3.987 s.

Additional generated schedules are available via this website [17].

5 Using Generated Activity Schedules for Simulation

Smart house are often equipped with passive infrared (PIR) sensors. When residents are in the detection range of one, it turns on, otherwise, it remains off. Each PIR sensor has a unique ID number which can be recorded when the sensor turns on or off. By placing several PIR sensors in the house and analyzing their records, a resident's movement trajectories can be acquired, which can be used to determine whether they contain wandering travel patterns associated with dementia [18].

In the simulation, the PIR sensor records were generated from simulated real-time location data. We built a generator that could convert an activity schedule, a sample of indoor spatial data, and several samples of travel pattern data into a sample of real-time location data. We developed an interface to convert the real-time location data into virtual PIR sensor records, which can be used to optimize the placement of the PIR sensors in a smart house.

Examples of the performance of the real-time location data generator and the interface are shown in figures and tables. Figure 2 shows a sample of spatial data. Table 2 shows part of an activity schedule. The travel pattern data are shown in Fig. 4. The above data are input into the generator to produce the real-time location data. Figure 4 also shows the positions of the five PIR sensors located in the virtual house. Their coordinates are [100, 0], [300, −150], [500, −200], [550, 50] and [850, 0]. The interface converts the real-time location data into the records of PIR sensors, which is shown in Table 3.

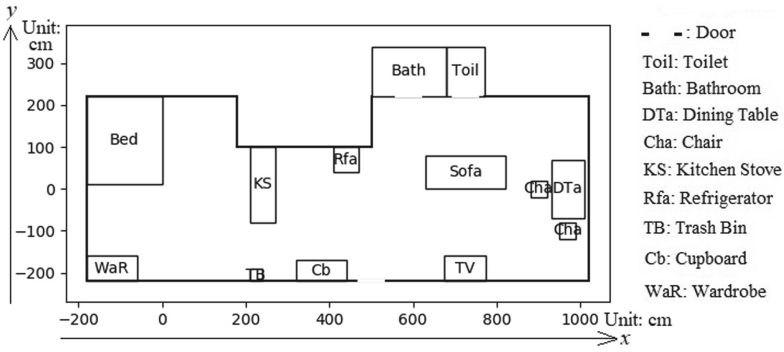

Fig. 2. Layout of input indoor spatial data.

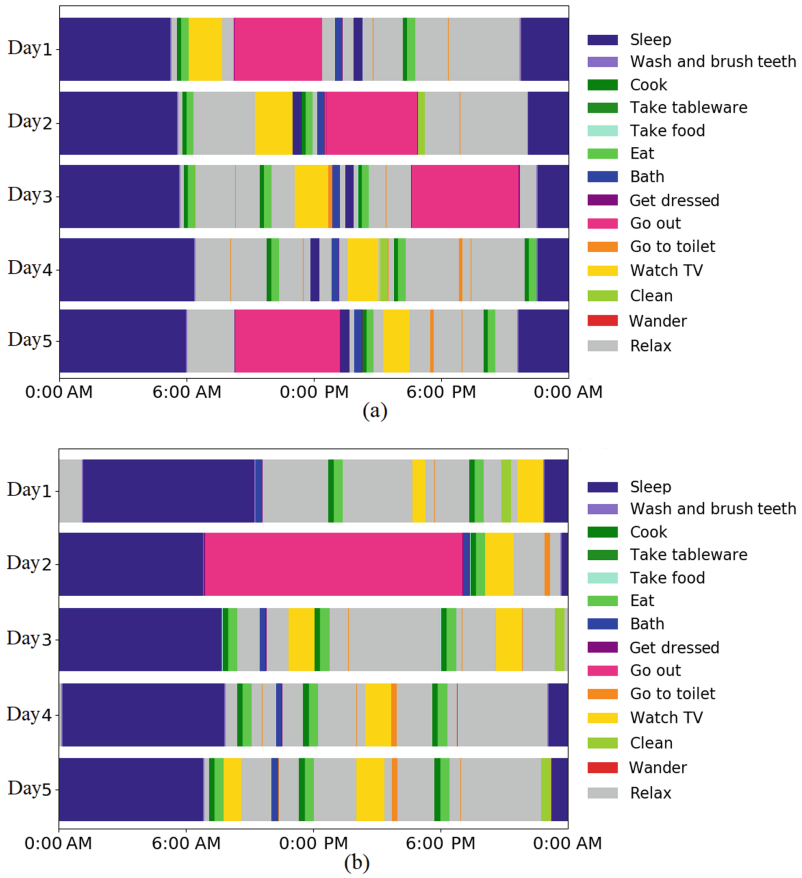

Fig. 3. Generated activity schedules.

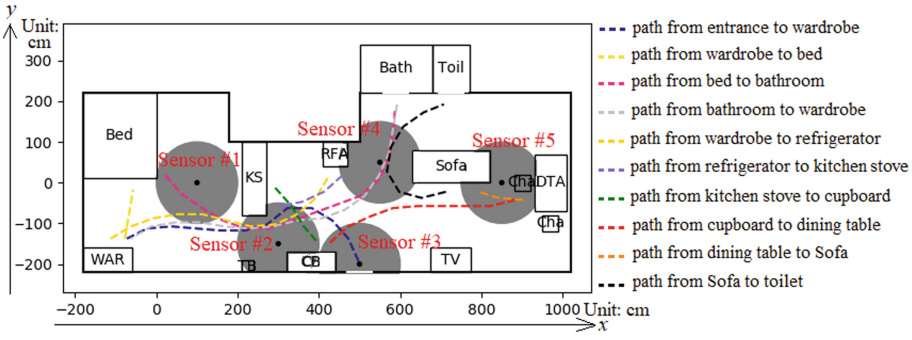

Fig. 4. Layout of input indoor spatial data with PIR sensors and travel pattern data.

Table 2. Part of the activity schedule shown in Fig. 3(a).

Activity	Start time	Activity	Start time
Go out	5d AM 8 h 17 m 7 s	Cook	5d PM 2 h 17 m 56 s
Get dressed	5d PM 1 h 13 m 59 s	Take tableware	5d PM 2 h 28 m 44 s
Sleep	5d PM 1 h 16 m 12 s	Eat	5d PM 2 h 29 m 46 s
Relax	5d PM 1 h 40 m 0 s	Relax	5d PM 2 h 49 m 20 s
Bath	5d PM 1 h 54 m 36 s	Watch TV	5d PM 3 h 15 m 1 s
Get dressed	5d PM 2 h 15 m 10 s	Go to toilet	5d PM 4 h 30 m 33 s
Take food	5d PM 2 h 17 m 18 s		

Table 3. Simulated records of virtual PIR sensors.

Time		Time	
5d PM 1 h 13 m 45 s 9	#3 ON	5d PM 2 h 17 m 7 s 5	#1 ON
5d PM 1 h 13 m 47 s 8	#3 OFF	5d PM 2 h 17 m 9 s 7	#1 OFF
5d PM 1 h 13 m 50 s 2	#2 ON	5d PM 2 h 17 m 10 s 9	#2 ON
5d PM 1 h 13 m 53 s 4	#2 OFF	5d PM 2 h 17 m 14 s 0	#2 OFF
5d PM 1 h 54 m 19 s 5	#1 ON	5d PM 2 h 17 m 50 s 5	#4 ON
5d PM 1 h 54 m 22 s 9	#1 OFF	5d PM 2 h 17 m 50 s 6	#4 OFF
5d PM 1 h 54 m 24 s 2	#2 ON	5d PM 2 h 17 m 53 s 6	#2 ON
5d PM 1 h 54 m 27 s 5	#2 OFF	5d PM 2 h 28 m 38 s 5	#2 OFF
5d PM 1 h 54 m 30 s 2	#4 ON	5d PM 2 h 28 m 39 s 7	#2 ON
5d PM 1 h 54 m 34 s 4	#4 OFF	5d PM 2 h 29 m 34 s 5	#2 OFF
5d PM 2 h 14 m 52 s 3	#4 ON	5d PM 2 h 29 m 34 s 5	#3 ON
5d PM 2 h 14 m 56 s 4	#4 OFF	5d PM 2 h 29 m 36 s 1	#3 OFF
5d PM 2 h 14 m 59 s 1	#2 ON	5d PM 2 h 29 m 41 s 9	#5 ON
5d PM 2 h 15 m 2 s 5	#2 OFF	5d PM 3 h 30 m 21 s 7	#5 OFF
5d PM 2 h 15 m 4 s 4	#1 ON	5d PM 3 h 30 m 23 s 8	#4 ON
5d PM 2 h 15 m 5 s 1	#1 OFF	5d PM 3 h 30 m 27 s 7	#4 OFF

6 Conclusion

Smart houses with a sensor network and domestic robots were built to take care elderly people living alone. Many simulation tools have been proposed to help smart house developers test and verify their designs, but it takes time and effort to build a simulation scenario, and developers need to repeat scenario-generation procedures if they want to use multiple simulators. To address these issues, we proposed a simulation tool that provides diverse simulation scenarios and enables developers to build scenarios automatically in multiple simulation platforms [11].

In this paper, we proposed an activity schedule generator that is an essential part of our simulator. With an improved motivation-driven method, the generator produces diverse daily activity schedules to mimic the daily lives of residents living alone. It outperforms existing generators in three aspects: 1) it is adaptive to the layout of a

simulated smart house; 2) there is no unspecified time in the timeline of generated schedules; and 3) it generates stable, but not tedious schedules for a number of days.

A generated schedule includes a list of activities and their start time. The list of activities determines all starts and ends of indoor walking paths with spatial attributes of a virtual house, the travel pattern generator then generates all paths. The generated paths determine simulated real-time locations of a resident with the list of start time.

The real-time locations can be converted to records of virtual sensors with interfaces, and these records can be used to optimize designs of smart house. For example, we convert the real-time locations to records of virtual PIR sensors, and the records are useful for optimizing placement of these sensors.

Acknowledgments. This research was partially supported by a grant from the Japan Society for the Promotion of Science (JSPS KAKENHI 18H00968) and a scholarship of Mizuho International Foundation.

References

1. He, W., Goodkind, D., Smith, P.K.: An Aging World: 2015: International Population Reports, U.S. Census Bureau (2016)
2. Reher, D., Requena, M.: Living alone in later life: a global perspective. Popul. Dev. Rev. **44**(3), 427–454 (2018)
3. Pimouguet, C., et al.: Impact of living alone on institutionalization and mortality: a population-based longitudinal study. Eur. J. Public Health **26**(1), 182–187 (2016)
4. Simoens, S., Villeneuve, M., Hurst, J.: Tackling nurse shortages in OECD countries: Technology Report, Organisation for Economic Co-operation and Development (2005)
5. Vuong, N.K., Chan, S., Lau, C.T., Chan, S.Y., Yap, P.L., Chen, A.S.: Preliminary results of using inertial sensors to detect dementia-related wandering patterns. In: Proceedings of 37th International Conference on Engineering in Medicine and Biology Society (EMBC2015), Milan, pp. 3703–3706 (2015)
6. Das, B., Cook, D.J., Krishnan, N.C., Schmitter-Edgecombe, M.: One-class classification-based real-time activity error detection in smart homes. IEEE. JSTSP **10**(5), 914–923 (2016)
7. Do, H.M., Pham, M., Sheng, W., Yang, D., Liu, M.: RiSH: a robot-integrated smart home for elderly care. Rob. Auton. Syst. **101**, 74–92 (2018)
8. Fischinger, D., et al.: Hobbit, a care robot supporting independent living at home: first prototype and lessons learned. Rob. Auton. Syst. **75**(A), 60–78 (2016)
9. Alshammari, N., Alshammari, T., Sedky, M., Champion, J., Bauer, C.: OpenSHS: open smart home simulator. Sensors **17**, 1003 (2017)
10. Vaughan, R.: Massively multi-robot simulation in stage. Swarm Intell. **2**, 189–208 (2008)
11. Jiang, C., Mita, A.: Automatic spatial attribute and travel pattern generation for simulating living spaces for elderly individuals living alone. Build. Environ. **176**, 106776 (2020). https://doi.org/10.1016/j.buildenv.2020.106776
12. Jiang, C., Mita, A.: Automatic floorplan generation of living space for simulating a life of an elderly resident supported by a mobile robot. In: Proceedings of 36th International Symposium on Automation and Robotics in Construction (ISARC 2019), Banff, pp. 688–695 (2019)

13. Renoux, J., Klügl, F.: Simulating daily activities in a smart home for data generation. In: Proceedings of Conference on Winter Simulation (WSC2018), Gothenburg, pp. 798–809 (2018)
14. Bouchard, B., Gaboury, S., Bouchard, K., Francillette, Y.: Modeling human activities using behaviour trees in smart homes. In: Proceedings of 11th Conference on PErvasive Technologies Related to Assistive Environments (PETRA), Corfu, pp. 67–74 (2018)
15. Mshali, H., Lemlouma, T., Magoni, D.: Context-aware adaptive framework for e-health monitoring. In: IEEE International Conference on Data Science and Data Intensive Systems, Sydney (2015)
16. Lee, W., et al.: Automatic agent generation for IoT-based smart house simulator. Neurocomputing **209**, 14–24 (2016)
17. https://github.com/Idontwan/Activity-Schedule-Generator/tree/master/Schedules
18. Gochoo, M., Tan, T., Velusamy, V., Liu, S., Bayanduuren, D., Huang, S.: Device-free non-privacy invasive classification of elderly travel patterns in a smart house using PIR sensors and DCNN. IEEE Sens. J. **18**(1), 390–400 (2018). https://doi.org/10.1109/JSEN.2017.2771287