Received January 10, 2018, accepted February 9, 2018, date of publication February 19, 2018, date of current version March 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2807761

An Activity Rule Based Approach to Simulate ADL Sequences

STEIN KRISTIANSEN, THOMAS P. PLAGEMANN, AND VERA GOEBEL
Department of Informatics, University of Oslo, 0316 Oslo, Norway

Corresponding author: Stein Kristiansen (steikr@ifi.uio.no)

This work was supported in part by the Regional Research Fund Oslofjord through the TRIO Project under Grant 217737, and in part by the Research Council of Norway through the CESAR Project under Grant 250239/O70.

ABSTRACT The concept of activities of daily living (ADL) has for many years successfully been used in a broad range of health and health care applications. Recent hardware and software developments suggest that the future use of ADL will not only benefit from the transition from manually created ADL logs to automatic sensor-based activity recognition and logging but also from the transition from manual inspection of ADL sequences to their automatic software-driven analysis. This ADL sequence analysis software will be core part in mission critical systems, like ambient assisted living, to detect for example changing health status. Therefore, proper testing and evaluation of this software is mandatory before its deployment. However, testing requires data sets that include normal ADL sequences, hazards, and various kinds of long term behavioral changes; which means it might require weeks or even months to monitor individuals to capture such ADL sequences. Thus, collecting such data sets is very costly, if feasible at all; and very few data sets are available on-line. Therefore, we present an approach to create the necessary data sets for testing through simulation. The simulation of ADL sequences is based on existing ADL sequences and uses probabilistic activity instigation and durations with a novel concept called activity rules to create data sets for proper testing. Activity rules are used to model how individuals resolve activity conflicts. We implemented these concepts as a discrete event simulator, called ADLSim. The evaluation of ADLsim shows that the simulated ADL sequences are realistic and able to capture the variability and non-predictable behavior found in the real world, and that activity rules can impact simulation results significantly.

INDEX TERMS Activities of daily living, activity rules, discrete event simulation, evaluation, modelling.

I. INTRODUCTION

The concept of Activities of Daily Living (ADL) plays a central role in many Health Care services. It refers to the basic activities required to be performed on a daily basis to live an independent life. It is particularly relevant in elderly care. By ranking the adequacy of performing basic tasks like feeding, bathing, toileting and transferring, one can obtain objective measures on the health status and independence of elderly people [14]. ADL are however also important outside of elderly care, e.g., when assessing the independence of injured [13], [22] and diseased people [24], or people with certain disabilities [23]. In many cases, it is desirable to monitor care receivers, for instance to assess their recovery process, capture indications of health deterioration or to prevent dangerous situations. With today's rapid increase in the elderly population, the human resources to perform the necessary monitoring and analysis is becoming ever scarcer. Information and Communication Technology (ICT) supported home care and elderly care, like Ambient Assisted Living (AAL), aim to alleviate this problem by automating the monitoring and the analysis of monitoring results. The purpose is to enhance the quality of life by improving the quality of care and enabling individuals an independent life.

There are three key technologies to achieve this goal (see Figure 1-a): (1) sensor technology [1], (2) activity recognition [5], [19], [26], [28], [29], and (3) software to analyze the recognized sequences of activities, because important developments of health and independence often show up only as long-term developments in ADL sequences. Thus, it is necessary to perform longitudinal analysis and mining of ADL sequences to unveil long-term trends [16], [27]. The resulting system can be seen as a mission critical system that can have severe impact on health and even life if they do not work properly. As such it is very important that all

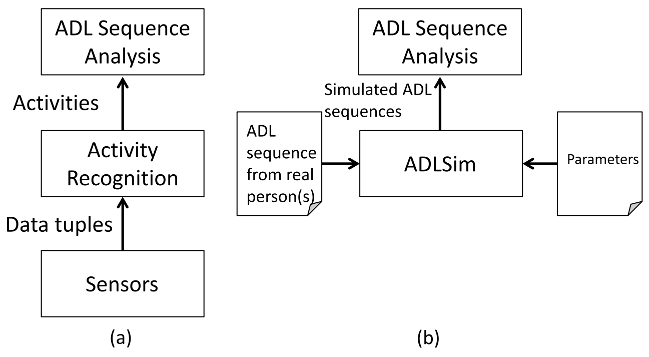


FIGURE 1. **(a) Key AAL enabling technologies. (b) The role of ADLSim, i.e., to test software for ADL sequence analysis.**

system components are tested thoroughly before they are deployed. Sensor testing and evaluation can be done in a controlled environment, and the range of input signals is known for the sensor types. However, it is not straightforward to evaluate the software for activity recognition and ADL sequence analysis. Both require data sets as input for testing, but larger data sets are required for ADL sequence analysis than for activity recognition. Instead of data from individual activities that are seconds or minutes long, ADL sequences from periods of weeks or months are needed for longitudinal studies. Thus, substantially more effort is required to create data sets to test ADL sequence analysis software. As such it is not surprising that there exist considerably more data sets for activity recognition [25]. In fact, there exist very few data sets with ADL sequences despite the long and prominent history of ADL in health and Health Care.

We explain the lack of such data sets as follows: First, ICT assisted home and elderly care is a relatively new research area, and has not yet been widely adopted in Health Care. Like any other new technology, it consequently suffers from an initial lack of adequate real-world data sets. First, there has been efforts to create facilities where such data can be obtained [10], [11], [27], but there are currently only very few open data sets with recordings of ADL sequences available on the Internet, like [29] and [30]. Second, capturing real-world ADL sequences is costly, because the required monitoring needs to be conducted over long time periods, and it is not easy to recruit human test subjects. Third, an ADL sequence captured in the real-world only reflects what has happened and represents only one realistic instance of possible ADL sequences. Therefore, it is impossible that all possible representative ADL sequences of any potential caretaker in any possible situations are captured in real-world data sets. For example, hazards are - as seen from the care takers point if view - hopefully not captured. Yet, software to assist caretakers must be able to properly detect hazards and general trends combinations of these. In other words, real-world data sets are (if available) a good choice to test particular cases, but not sufficient to perform testing in such a way that there is a rather high probability that the software detects all important events and provides the correct analysis results. Producing hand-crafted ADL sequences could be used to create data sets for stress testing the software, but again these can only include selected cases since it is cumbersome to create the ADL sequences and there is the danger of bias when data sets are crafted by hand to determine if the software analyses the ADL sequence correctly. The problem of insufficient data sets is worsened if the ADL sequence analysis software is based on deep learning technology. Deep learning approaches are very promising, but their performance strongly depends on the training data in terms of quantity and quality.

We aim to support development, testing, and evaluation of ADL sequence analysis software by presenting in this paper an approach and its implementation as discrete event simulator (called ADLSim). Figure 1-b illustrates the role of ADLSim, namely to produce simulated ADL sequences suitable to test ADL sequence analysis software. During deployment, the ADL sequence analysis software receives the input from activity recognition software. The simulated sequences describe the activities performed in an apartment by a single person, i.e., the simulation of multiple persons is reserved for future work. The approach is nevertheless already very useful for a large class of AAL software, e.g., to assess the independence of elderly, diseased or handicapped people that live alone. Like most other simulators, ADLSim is a set of tools to create models, and an execution environment to execute these models. As such, ADLSim is agnostic to the ADL that are required for testing. To create appropriate data sets for testing and evaluation, the developer of ADL sequence analysis software needs to select a suitable set of ADL to model long-term behaviour of the care taker, and to determine the parametrisation of the resulting models. The goal is to produce realistic ADL sequences with enough random behaviour such that a sufficiently large set of different ADL sequences can be used and that there is no bias between the tester and the developer (which is often the same person). To address the challenge of realism we model the behaviour of caretakers using probability distributions to determine when activities are instigated and for how long they last. We base the simulator on discrete event simulation to facilitate flexible simulation, i.e., of real behaviour learned from a real-world data set, purely hypothetical behaviour, or a combination of both.

A key contribution in this work is the novel concept called activity rules to accurately simulate the process of deciding which activity to perform first when faced with multiple candidate activities (called activity conflict). This is a well-known situation from real life, and how conflicts among different pairs of activities are resolved strongly depends on the individuals. Therefore, we argue that activity rules are important for the generation of realistic ADL data sets. Furthermore, the combination of adjustable activity rules and distribution parameters enable model individualization, i.e., they can be adjusted to reflect differences among individuals in terms of key characteristics of behaviour.

The reminder of this paper is structured as follows. We first summarize related work in Section II. Section III presents the principles we follow to create models, Section IV describes how these models are used to perform ADL simulation, and Section V how we parametrise models using an ADL sequence captured from a real person. Section VI presents an evaluation of our simulator, and Section VII concludes the paper and presents future work.

II. RELATED WORK

There exist already several simulators that in some way support the evaluation of ADL analysis software. Some of these simulators use existing ADL sequences as input to simulate movement [6] or sensor readings over time [7]. The simulators rely on pre-existing data sets, either with ADL sequences recorded from a real person available on-line or by creating the traces manually. Manual creation of data sets implies a certain amount of work, and recordings of ADL sequences from the real world contain only behaviour that has occurred in real life. Such simulators are therefore not suitable to evaluate ADL analysis software designed to detect long-term changes in behaviour, i.e., across months or years, and/or behaviour that occurs seldom in the real world and might therefore not be captured in data sets recorded from a real person. In addition, it is not possible in these simulators to introduce behaviour for specific hypothetical hazards or anomalies. ADLSim supports effective, automatic generation of data sets with long-term ADL sequences to enable the evaluation of such software.

Long-running ADL sequences are most commonly generated using Markov models [2], [21], [31]. This is motivated by the observation that human behaviour is probabilistic such that activities are faithfully modelled as states in a Markov model, which is supported by the successful application of Markov models in works on activity recognition, e.g., in [29]. The work in [31] and [32] furthermore leverages the circadian and habitual rhythm in human behaviour to create accurate models. The main drawback of these approaches is that the models are relatively difficult to extend, e.g., to explore various types of purely hypothetical behaviour.

A more flexible approach is to simulate the onset of activities either based on (1) the need to perform the activity, e.g., physiological needs like hunger or thirst [3], [8], [15], or (2) statistics obtained from data sets recorded from real people [9], [18]. These approaches facilitate the exploration of hypothetical activities and activity sequences through the modification, addition or removal of individual statistical distributions or need models. Such approaches however introduce a new problem: since activities are not instantaneous, i.e., have a duration of more than zero time units, conflicts can occur where more than one candidate activity is eligible for execution at any given point in time. Such conflicts are typically resolved by either interrupting or postponing activities according to some general scheme, e.g., priorities, that applies to all simulated activities in the same way. We argue that resolving all activity conflicts the same way is often not realistic. In real life, the way activity conflicts are resolved depend on the activities involved in the conflict, as well as the concrete individual carrying out these activities.

To the best of our knowledge, our approach and its implementation in ADLSim are advancing the state-of-the-art in three aspects: (1) we provide an extensible set of activity rules that allow to realistically discriminate among activity conflicts when determining how to resolve them, (2) we facilitate the seamless introduction of purely hypothetical behaviour into otherwise normal behaviour that can be learned from a data set with a recording of a real-world ADL sequence, and (3) we integrate ADLSim with the popular CEP system Esper to facilitate direct evaluation of state-of-the-art data analysis software, which is demonstrated in our previous work [16].

III. REQUIREMENTS AND ADL MODELLING PRINCIPLES

This work addresses the challenge of creating a model of the behaviour of a single human (expressed as ADL sequences) that enables the simulation of normal behaviour, with the unpredictable day-to-day variability found in real daily living, as well as hypothetical behaviour involving anomalies and hazards. We first look at three fundamental requirements that must be satisfied for ADL simulation to be useful, and then look at the principles we follow to satisfy these requirements.

A. REQUIREMENTS FOR ADL MODELLING

For an ADL simulator to be as useful as possible, it should satisfy the following three fundamental requirements: (1) it should produce realistic ADL sequences, (2) it should provide the possibility to simulate purely hypothetical behaviour, and (3) it should be activity agnostic. We explain each requirement below.

Requirement 1 - Realism: To perform reliable evaluation of software for ADL sequence analysis, it is important that the evaluation is based on realistic ADL sequences, i.e., that essential characteristics of the simulated sequences correspond to behaviour that can be found in the real world. This includes realistic day-to-day variations in behaviour which may yield non-deterministic results that cannot be predicted before simulation. Since individuals differ in personality, life style, age, gender, etc., realistic ADL sequences are those that capture the behaviour that is characteristic for a given real or imagined individual. This furthermore implies that it should be feasible to parametrise the models to reflect what is found in an ADL sequence captured from a real person.

Requirement 2 - Ability to Simulate Hypothetical Behaviour: ADL sequence analysis software in AAL systems is commonly designed to analyse long-term trends as well as to detect high-risk events, e.g., particular hazards or anomalies. These events might occur rather seldom and might not exist in available data sets with recordings of real-world ADL sequences. This problem is exacerbated by the fact that a large number of events are required to be confident that the ADL sequence analysis software can in fact detect the event. To enable the evaluation of such systems, it is therefore necessary that our simulator can simulate purely hypothetical behaviour, i.e., that is not present in any recorded real-world

ADL sequence. In the absence of a suitable data set, it might even be necessary to construct complete, purely hypothetical personalities, which was necessary in our previous work [16]. It should furthermore be possible to seamlessly mix hypothetical behaviour into normal behaviour that is, e.g., learned from a real-world ADL sequence. The reason is that ADL sequence analysis software is typically evaluated in terms of how accurately they identify interesting behaviour that may not be present in any real-world ADL sequence. It should also be possible to vary the simulated duration as necessary to, e.g., enable longitudinal studies.

Requirement 3 - Activity Agnosticism: The traditional use of ADL is focused on a small set of basic and instrumental activities, which are not necessarily sufficient for future AAL systems and other applications that use ADL. As such, ADLSim should not be bound to a particular set of activities and it must be very easy for developers to introduce new activities in ADLSim.

B. PRINCIPLES OF MODELLING ADL SEQUENCES

We define activity as a tuple $a = \langle id, name \rangle$, where id is a unique identifier and $name$ is a descriptive string. Activities are typically performed multiple times over a period of time. This is captured by the concept Activity Instance (AI), which is a tuple $ai = \langle a, st, et \rangle$, where a is an activity and st and et its start and end times. The overall goal of ADLSim is to simulate a realistic sequence of such AI over time, called Activity Instance Sequence (AIS). AIS is formally defined as an ordered sequence $as = \{ai_0, \ldots, ai_{n-1}\}$ of n AI where $\forall x, y : x \leq y \Rightarrow ai_x.st \leq ai_y.st$, i.e., where AI are ordered in increasing order of start time. Note that this definition allows input and output AIS to contain overlapping AI, e.g., watching TV and answering the phone.

In order to generate within ADLSim useful AIS, we capture the characteristics of activities and how individuals manage them in so-called Activity Models (AM). The fundamental design principles for realistic AM are based on the insight that a central task of the simulator is to determine the start time $ai_x.st$ and end time $ai_x.et$ of each AI ai_x, i.e., the durations $ai_x.et - ai_x.st$. Therefore, AM are based on the following assumptions:

1) Within ADLSim only one AI is selected as the current AI at any point in virtual time, but it is possible to switch from one AI to another at any time. Since human multitasking is in most cases rapid task switching we consider AI switching as a good approximation for concurrent activities.
2) The time at which activities are performed is based on factors that motivate their execution. Examples include physiological or psychological needs like thirst, hunger, and the need to socialise and sleep, daily routines like eating dinner at a given Time-of-Day (ToD) and executing certain activities in sequence, e.g., exercising and showering.
3) Routines, needs, and the circadian rhythm causes AIS to exhibit statistical regularities. This can differ from person to person, and implies probabilistic day-to-day variations in AIS.

The naive approach where start times $ai_x.st$ and durations $ai_x.et - ai_x.st$ of AI ai_x are determined only based on probability distributions can lead to conflicts. The reason is that AI have a non-zero duration, i.e., $ai_x.et - ai_x.st > 0$, which can lead to a situation where two AI ai_a and ai_b are both potential instances for the current activity, i.e., where $ai_a.st \leq ai_b.st$ and $ai_b.st \leq ai_a.et$. This corresponds to a situation in which a person desires to perform multiple activities simultaneously. Since performing several activities in parallel violates Assumption 1, we refer to this situation as an Activity Conflict. We need mechanisms to serialise the conflicting activities in a realistic way, i.e., that corresponds to the real-life process of deciding which activity to perform. This may in turn lead to other activities being postponed, interrupted or omitted. We handle these challenges using three core concepts: (1) instigation time, i.e., when a person wishes to perform an activity, (2) start time, i.e., when the activity is actually started, and (3) Activity Rules (AR) which define how to serialise overlapping activities.

AR are invoked upon activity conflicts and guide their serialization. Examples include terminating or suspending an ongoing AI ai_{cur} to execute a newly instigated activity ai_{new}, postponing ai_{new} until the completion of ai_{cur}, or omitting ai_{new} and re-scheduling it for later instigation. The appropriate AR for any given activity conflict depends mainly on the involved activities and the personality of the simulated individual. AR can for example be based on interviews or semantics of activity names. The chosen set of AR should faithfully reflect an important part of daily behaviour that differs between individuals. Note that such serialization might result in a discrepancy between when a person wishes to perform an activity, i.e., the instigation time, and when it is actually performed, i.e., the start time, which causes non-deterministic variations during simulation. In the presence of multiple postponed or suspended activities, a given AI may be subjected to a sequence of conflict resolutions per instigation before it is started, and it may be interrupted multiple times during execution. Note that the probability and frequency of activity conflicts increases with the activity density, which is determined by the duration and frequency of AI. The number of activity conflicts also tends to increase with the variance of the chosen distributions, since collections of highly dispersed probability distributions generally have larger overlapping regions than those with highly concentrated distributions.

There are two basic approaches to determine instigation time: (1) use motivation factors and trigger activation based on threshold values, like in [3], [8], and [15], or (2) use statistical distributions of instigation time and duration, potentially obtained from an AIS from a real person, like in [9] and [18]. Approach 1 is well suited to study how changes in motivation affect human behaviour. However, one important task in home care is to detect and signal about abnormal or hazardous situations. Deviations from the normal, and similarities between the current situation and particular hazards are typically detected based on statistical analysis. It is thus important that the simulated behaviour exhibits the statistical properties that characterise the event of interest. We argue that need-based simulation is not well suited for this, since it is not possible to predict a priori how the development of needs affect the statistical properties of the final behaviour. Therefore, we adopt Approach 2 that (1) facilitates direct specification of the statistical properties of the behaviour, (2) allows for parametrisation based on AIS from the real world, and (3) permits the direct statistical comparison of simulated AIS with AIS from the real world.

Another important aspect of Requirement 1 (realism) is to capture the type of unpredictable behaviour found in reality and to create AIS that cannot be foreseen by developers. This is commonly realised with stochastic, discrete event simulation. Non-predictable output is produced by combining two elements: (1) sufficiently complex models that change over time according to deterministic rules, and (2) pseudo-random numbers that are used as input to these models. While the distributions of the random numbers are known a priori, the output from the models is often non-predictable and can only be known after executing the simulation. As an example, consider the case of computer network simulation. These simulators use models of interconnected nodes with protocols that follow deterministic rules, and pseudo-random numbers, e.g., for bit errors, back-off times, and node movements. Both the protocol rules and the parameters of the random distributions are known a priori, but the results of the simulation, such as end-to-end delay, throughput, jitter, and packet loss, are only known after executing the simulation. The same is true for instigation times and start times in ADLSim. While instigation times are driven by a priori known statistical distributions, start times are unknown before simulation and can vary significantly and non-deterministically between runs with different random seeds.

IV. SIMULATING AIS

To validate our ADL modelling approach, we implement ADLSim as a discrete event simulator using an extensible set of AM based on parametric probability distributions to determine the instigation and durations of activities, and AR to resolve activity conflicts among AI. By combining discrete event simulation with parametric distributions and easily modifiable AR, it is possible to meet Requirement 1 (realism) by choosing distribution parameters and AR that reflect the essential aspects of a given real or personality. Requirement 2 (the ability to introduce hypothetical behaviour) is satisfied because (1) discrete events can be scheduled to introduce arbitrary changes to the models at any given point in time, and (2) parametric distributions and AR can easily be modified by these events to introduce arbitrary modifications in behaviour. Finally, the AM and AR are designed from the beginning to satisfy Requirement 3 (activity agnosticism), by composing AM from only highly generic components like probability distributions that describe activity onset and duration. We thereby make only minimal assumptions about the type of activities we can model with AM and AR. They are for instance not restricted to model the activities found in any particular real-world AIS, and can, as we demonstrate in Section VI, be parametrised to model entirely hypothetical activities that are not found in any given real-world AIS. We perform further evaluation of ADLSim in the context of these requirements via the experiments and results analysis in Section VI. Below, we describe the simulation loop in Figure 2, with emphasis on how the different types of parameters (highlighted in red) affect simulations.

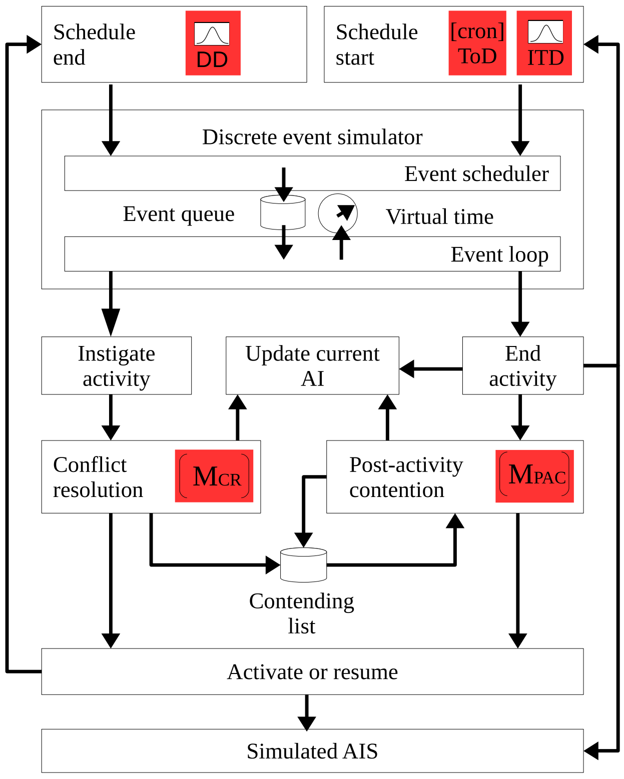


FIGURE 2. Simulator design. The main parameters are outlined in red.

A. ACTIVITY MODELS

The instigation times and durations of AI are determined based on probability distributions and their parametrisations. Instigation times are determined using an instigation time distribution (ITD) defined as a tuple $ITD = \langle PDF : f(x, dp), dp : \{p_1 \in \mathbb{R}, p_2 \in \mathbb{R}, \ldots\}\rangle$, where PDF is the probability density function, and p_i is parameter number i. Similarly, durations are determined using duration distribution (DD) defined as the tuple $DD = \langle PDF : f(x, dp), dp : \{p_1 \in \mathbb{R}, p_2 \in \mathbb{R}, \ldots\}\rangle$.

The way the ITD is interpreted depends on the type of the activity. Type 1 activities are typically need-based, e.g., using the toilet and grocery shopping. The instigation times of Type 1 activities are best approximated with a given frequency. Thus, their ITD describe the durations between subsequent instigations for that activity. Type 2 activities are typically routine-based and are performed at approximately the same points in time every day, e.g., having lunch, eating dinner, and taking medicine. Thus, instigation times of Type 2 AI are best approximated using (1) a set $ToD = \{C_0, \ldots, C_{n-1}\}$, where each C_i is a time of the day defined in the crontab format,[1] and (2) one ITD per ToD to add probabilistic deviations from the ToD. Thus, while the instigation times for Type 1 AI are based directly on a sample from their ITD, those for Type 2 AI are determined by adding a sample from their ITD to the corresponding ToD. Activities may furthermore be assigned a non-zero omission probability $OP \in [0, 1]$ to model the occasional omission of activities upon instigation.

An AM for an activity a is a tuple $\langle a : A, type \in \{1, 2\}, DDList, ITDList, ToDList, OPList\rangle$, where a is the activity to model, *type* defines the activity type, and *DDList*, *ITDList*, *ToDList* and *OPList* are lists of *DD*, *ITD*, *ToD*, and *OP*, respectively. One element is selected per list to determine an instigation time. Each list is iterated sequentially, wrapping back to the first element upon encountering the end of the list. Using such lists adds flexibility by, e.g., allowing an activity to be performed multiple times per day with differing *DD*, *ITD* and *OP*.

Note that the AM above do not explicitly encode any information about the ordering of AI. The reason follows from Assumption 2, implying that many activities are motivated mainly by physiological needs and the time of the day rather than by the activities performed before and after them. In that sense, our approach differs fundamentally from related works based on Markov chains. Instead, AI ordering emerges indirectly, and non-deterministically, during simulation as result of a series of probabilistic instigations of individual activities, coupled with the resolution of activity conflicts according to AR. As mentioned in Assumption 2, a subset of the activities may nevertheless commonly be performed in a given sequence as part of a routine. Examples include going to the bathroom before and after sleep, followed soon after by eating breakfast. To improve realism, we therefore complement the above AM with the possibility to specify such AI sequences, called AI nexi (plural of nexus), such that each activity is instigated immediately or some time after the completion of the preceding one according to its ITD. AI in such nexi are called dependent variants of the activity they regard, while the other AI are called independent variants. A given activity may have AI of both variants. Such nexi can, like individual activities, be of Type 1 or Type 2. Since an AI nexi is instigated at the time its first activity is instigated, the type of a nexus is inherited by its first activity.

B. ACTIVITY RULES

Activity rules for n activities are contained in n-by-n matrices of the form

$$M = \begin{bmatrix} AR_{A_0 \to A_0} & \ldots & AR_{A_0 \to A_{n-1}} \\ \vdots & \ddots & \vdots \\ AR_{A_{m-1} \to A_0} & \ldots & AR_{A_{n-1} \to A_{n-1}} \end{bmatrix}$$

[1]Explained in https://en.wikipedia.org/wiki/Cron/

where each element $AR_{A_{new} \to A_{cur}}$ is an activity rule denoting how to resolve the conflict between two AI, namely for a new activity A_{new} and a current activity A_{cur}. The elements of these are chosen from an extensible set of AR, with associated semantics and conflict resolution procedures.

We have two versions of the matrix M. The first, M_{CR}, is used for conflict resolution, i.e., when A_{new} is instigated during the execution of A_{cur}. Examples include terminating A_{cur} (called terminate), suspending A_{new} (called suspend), postponing A_{new} (called postpone) or omitting A_{new} (called omit). In addition, more complex custom rules and procedures can be created and invoked before and/or after consulting M_{CR} for particular activity pairs, e.g., "omit activity LeaveHouse when activity GoToBed is scheduled for instigation less than 30 minutes into the future".

As a result of conflict resolution, multiple AI might have been postponed or suspended during the execution of an ai_x. These are stored in the contending list $CL_{ai_x} = \{Cai_0, \ldots Cai_{n-1}\}$ belonging to ai_x together with any dependent AI following ai_x in a nexus. Upon the completion of ai_x, all AI in CL_{ai_x} are subjected to a post-activity contention to determine which AI ai_{x+1} should follow ai_x in the simulated AIS. The matrix M_{PAC} holds the AR that guides the post-activity contention via the following procedure: A candidate AI ai_{cand} for ai_{x+1} is initially set to $ai_{cand} = Cai_0$. For subsequent Cai_i in the list, the AR $AR_{C_i \to A_{cand}}$ in M_{PAC}, where $A_{cand} = ai_{cand}.a$ and $C_i = Cai_i.a$, is consulted to determine whether or not Cai_i should replace the current AI ai_{cand}. If so, the AI are swapped, i.e., the current AI ai_{cand} is inserted into the contending list of Cai_i, and Cai_i is chosen as the current candidate by setting $ai_{cand} = Cai_i$. If, on the other hand, Cai_i should not replace the current ai_{cand}, Cai_i is inserted into in the contending-list of ai_{cand}. When all Cai_i have been evaluated as potential current candidate, the final ai_{cand} is chosen as the AI ai_{x+1} that follows ai_x in the simulated AIS. The exception is if ai_{x+1} is omitted due to a non-zero omission probability, upon which the process is repeated with the contending list of the final ai_{cand}.

For a detailed example of M_{CR} and M_{PAC}, consult Section B where the content of the M_{CR} and M_{PAC} used in one of the experiments in our evaluation, HypoSim, is presented and explained.

V. MODEL INSTANTIATION USING REAL-WORLD AIS

In order to create realistic AIS it is important to be able to parametrise the models to reflect the distinct characteristics of the behaviour of real individuals. This section presents an approach to instantiate such models based on an AIS of a real person, hereafter referred to as a real-world AIS. We base our approach on the iterative parametrisation process depicted in Figure 3. We consider four potential sources of information:

1) **Statistics from the real-world AIS** can be used to estimate ITD and DD parameters and nexi of activities.

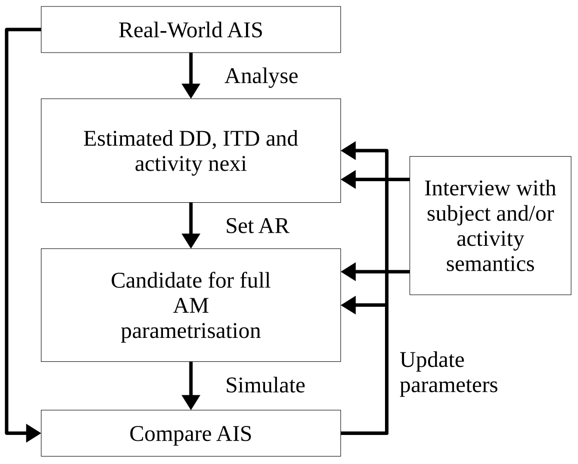

Real-World AIS
Analyse
Estimated DD, ITD and activity nexi
Set AR
Candidate for full AM parametrisation
Simulate
Compare AIS
Interview with subject and/or activity semantics
Update parameters

FIGURE 3. **Parametrising ADLSim using a real-world AIS.**

2) **Semantics implied in the activity names and/or descriptions** can help determine AR and nexi of activities.
3) **Interviews with the monitored person** (if available) can help determine AR and nexi of activities, since these parameters reflect how the modelled individual resolves activity conflicts.
4) **Simulated AIS** can be compared with the input AIS from the real person to investigate how representative they are, and may suggest how to improve the parametrisation. We provide one example in Section VI where results from a simulation Sim1 help to improve parametrisation in a second simulation Sim2 via, e.g., outlier removal.

Parameters of DD can be estimated from the durations of the AI in the real-world AIS, e.g., using maximum likelihood estimation. We currently assume that AR are selected based only on activity semantics or interviews. Estimating AR from the real-world AIS is left for future work. We explain below how to estimate ITD and activity nexi from real-world AIS.

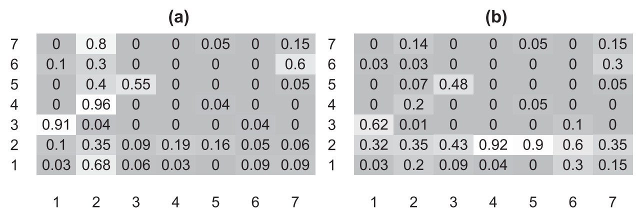
A. ESTIMATING ITD PARAMETERS

The real-world AIS denotes only which activities are performed when, i.e., their start time st and end time et, and lacks explicit information about instigation times. Therefore, the best we can do is to estimate instigation times based on the available start times. This is based on the assumption that AI are started as soon as possible after instigation.

To estimate ITD, we must first determine the type of the activity. We do this by studying histograms of the AI start times. Histograms for Type 1 activities exhibit distinct modes around one or many ToD, while those for Type 2 activities lack such modes. This is obvious since Type 1 activities are per definition those that are started around ToD. It should thus be clear from visual inspection of the histograms, or via established methods in circular statistic to identify modes, that Type 1 activities exhibit such modes. In this work, we achieve good results by visual inspection combined with the semantics of the activity names, and defer the study of potentially useful statistical tools for future work. As an example, consider the top row of histograms in Figure 5 for the seven activities from the real-world AIS used in our evaluation. The x-axes denote hour of the day, and the y-axes the frequency of start times within each hour (a mathematical definition of this metric is found in Section VI-B). We see that the histograms for Activity 3 (Take Shower), Activity 4 (Go to Bed), Activity 5 (Prepare Breakfast), and Activity 6 (Prepare Dinner) have distinct modes around ToD 10:15h, 9:24h, 19:22h, and 23:54h,[2] respectively, and are therefore likely of Type 2. The same is not true for Activity 2 (Use Toilet), which is more likely of Type 1. These conclusions are supported by the semantics of the activity names, i.e., the string "Go to Bed" denotes an activity that is typically started around the same ToD every day (Type 2 activity).

Once the activity type is decided, their ITD parameters are estimated. For Type 2 activities, this is done in the same way as with DD, i.e., by fitting the ITD to the empirical distribution of the durations between subsequent start times of the same activity (called inter-activity duration, IAD), e.g., with maximum likelihood estimation. For Type 1 activities, a ToD is estimated first, then the parameters of the ITD are estimated based on how start times deviate from this ToD. To estimate the correct ToD, e.g., 10:15h for Activity 3 (Take Shower) in Figure 5, we must take into account that the 24-hour period is cyclic. Simply calculating the average of the timestamps after performing a modulo operation on the 24-hour period can yield highly inaccurate results, especially for activities whose distribution of start times cross the 24-hour boundary. A good example is Activity 4 (Go to bed) with a significant amount of activations right before and right after midnight. The aforementioned linear average would result in an average ToD at 13:15h, which is clearly far away from the correct mode at 23:54h. ToD estimation requires instead the use of circular statistics [12] as follows. For each AI $ai_i \in \{ai_x \in ais : ai_x.a = act\}$ consisting of the AI in AIS ais belonging to Activity act, calculate $m_{ai_i} = ai_i.st \bmod P$, where P is the number of time units in a 24 hour period. All m_{ai_i} are translated into coordinates (x_{ai_i}, y_{ai_i}) on a unit circle where one revolution corresponds to 24 hours, i.e., $x_{ai_i} = cos(\frac{2m_{ai_i}\pi}{P})$ and $y_{ai_i} = sin(\frac{2m_{ai_i}\pi}{P})$. The arithmetic mean of these coordinates denotes a vector $(\overline{x_{ai_x}}, \overline{y_{ai_x}})$ whose angle $\theta_{ToD_{act}} = atan2(sin(\overline{y_{ai_x}}), cos(\overline{x_{ai_x}}))$ can be used to calculate the final ToD estimate $ToD_{act} = \frac{\theta_{ToD_{act}}}{2\pi P}$. ToD_{act} represents the time-of-day from which the average distance to all nearby $ai_i.st$ is as low as possible. Here, the term nearby is defined by taking into account the fact that, in linear time, the time-of-day represented by ToD_{act} repeats once every 24 hours. The instance of ToD_{act} that is nearby is thus defined to be the one closest to ai_i in linear time, and is denoted $ToD_{act,i}$. The parameters $ITD_{act}.dp$ for an ITD of

[2]Since the activity *Go to bed* is usually started around midnight, and the 24-hour period is presented in a linear way, the mode around midnight appears as two modes.



(a)

7	0	0.8	0	0	0.05	0	0.15
6	0.1	0.3	0	0	0	0	0.6
5	0	0.4	0.55	0	0	0	0.05
4	0	0.96	0	0	0.04	0	0
3	0.91	0.04	0	0	0	0.04	0
2	0.1	0.35	0.09	0.19	0.16	0.05	0.06
1	0.03	0.68	0.06	0.03	0	0.09	0.09
	1	2	3	4	5	6	7

(b)

7	0	0.14	0	0	0.05	0	0.15
6	0.03	0.03	0	0	0	0	0.3
5	0	0.07	0.48	0	0	0	0.05
4	0	0.2	0	0	0.05	0	0
3	0.62	0.01	0	0	0	0.1	0
2	0.32	0.35	0.43	0.92	0.9	0.6	0.35
1	0.03	0.2	0.09	0.04	0	0.3	0.15
	1	2	3	4	5	6	7

FIGURE 4. Probabilities of nexi of activities, coloured from grey (P = 0.0) to white (P = 1.0). (a) P(A->B | A). (b) P(A->B | B).

Activity *act* is then selected to best predict the distances $|ai_i.st - ToD_{act,i}|$, e.g., using maximum likelihood estimation. During simulation, ITD_{act} can now be sampled and added to ToD_{act} to produce the final instigation times. ITD_{act} and ToD_{act} are finally inserted into the lists $AM_{act}.ITDList$ and $AM_{act}.ToDList$, respectively, that belong to the final AM AM_{act} used to model activity *act*.

B. ESTIMATING AI NEXI

Activities with a large fraction of dependent AI tend to occur in frequently occurring sub-sequences of AI in an AIS. To find dependent AI, we therefore search the real-world AIS for common sequences. These are found by studying the frequency of AI pairs. Figure 4 presents the outcome of this process for the AIS used in our evaluation. Each entry in row A and column B contains the number of occurrences an AI ai_x pre-ceding an AI ai_{x+1}, such that $ai_x.id = A$ and $ai_{x+1}.id = B$ (denoted $A \rightarrow B$), divided by the total number of AI $|\{ai_i : ai_i.id = A\}|$ for the activity with ID A (Figure 4 (a)), and the total number of AI $|\{ai_i : ai_i.id = B\}|$ for the activity with ID B (Figure 4 (b)). The diagram to the left ($P(A \rightarrow B|A)$) shows the probability of an AI for Activity B following a given AI for Activity A, and the right diagram ($P(A \rightarrow B|B)$) shows the probability that an AI for Activity A pre-cedes a given AI for Activity B. Entries with high probabilities are likely to denote activities with a significant fraction of dependent AI variants, and thereby guide the construction of AI nexi.

In the diagram to the left, the two sequences $3 \rightarrow 1$ and $4 \rightarrow 2$ stand out as particularly probable. The semantics of the activity names confirm their probability. For instance, it is not unreasonable that the probability is 90% that the activity immediately following "Take shower" is "Leave house". Likewise, it is not unreasonable that the probability is 95% that the activity immediately following "Go to Bed" is "Use toilet". Furthermore, we find in the right diagram that going to bed is immediately preceded by using the toilet for 92% of its instances ($2 \rightarrow 4$). The two sequences with Activity 4 can be concatenated, resulting in the final nexi $2 \rightarrow 4 \rightarrow 2$ and $3 \rightarrow 1$. We re-produce these probabilities using the omit probability of each activity, e.g., Activity 2 is omitted from the nexi with probabilities $1 - 0.92 = 0.08$ and $1 - 0.096 = 0.04$ before and after Activity 4, respectively.

ITD parameters of complete nexi of activities are estimated as with individual activities. However, some activities have both dependent and independent AI variants, e.g., only 61.76% of instances of Activity 1 (Leave House) occur in the above-mentioned nexi. Therefore, special care must be taken to exclude independent AI when estimating the ITD of nexi, and vice versa. An additional benefit of modelling dependent and independent AI variants separately is that their ITD and DD may be more accurately fitted to the empirical data. For instance, the two AI variants of Activity 1 are mostly separated by the two modes in its histogram (to the top left in Figure 5) and are therefore accurately modelled with separate uni-modal distributions. The omit probability of the independent AI variant is finally adjusted to reflect the ratio by which it occurs in the real-world AIS.

VI. EVALUATION

We have implemented ADLSim as a discrete event simulator in Java according to the design in Figure 2. The goal of the evaluation is to show that ADLSim satisfies the three requirements, i.e., it is realistic, enables the simulation of hypothetical behaviour and that it is activity agnostic. Since AR is a new concept in the field of ADL sequence simulation, we also analyse the impact of AR. Our preliminary experiments show that ADLSim can simulate 1.5 million AI per second on a 1.6 GHz quad core Intel i5. This means a high-density AIS across a duration corresponding to a complete human life can be simulated within a few seconds. Since this simulation overhead is negligible we exclude further studies on simulation overhead from this paper.

This section is divided into five sub-sections. Section VI-A elaborates on the eight simulation experiments performed as part of our evaluation, with a detailed discussion on the parametrisation of each experiment. Section VI-B explains the metrics we use to quantify our results. Section VI-C presents the experiment results. Section VI-D analyses the results in the context of the three requirements and the impact of AR. Section VI-E summarises our findings.

A. EXPERIMENTS

This section presents the design of the eight experiments performed as part of our evaluation. To enable a steady-state analysis, we add one simulated day at the beginning and end of all simulations that is discarded from our results. In experiments Sim1, Sim2, Rand*, and NonRand, we perform the simulation 10 times with varying random seeds to obtain statistical significance, and present in Section VI-C the mean and standard deviation (SD) across all 10 runs. In the experiments HypoSim and HybridSleep we instead perform only one run with several times more simulated days. The reason is that these they are designed to demonstrate the ability ADLSim to simulate behaviour that occurs over very long periods of time. We explain each experiment in detail below. Consult Table 1 for the detailed parametrisation of the AM in each experiment.

1) SIM1 AND SIM2

The goal of Sim1 and Sim2 is to determine whether ADLSim can simulate realistic behaviour, and to show that

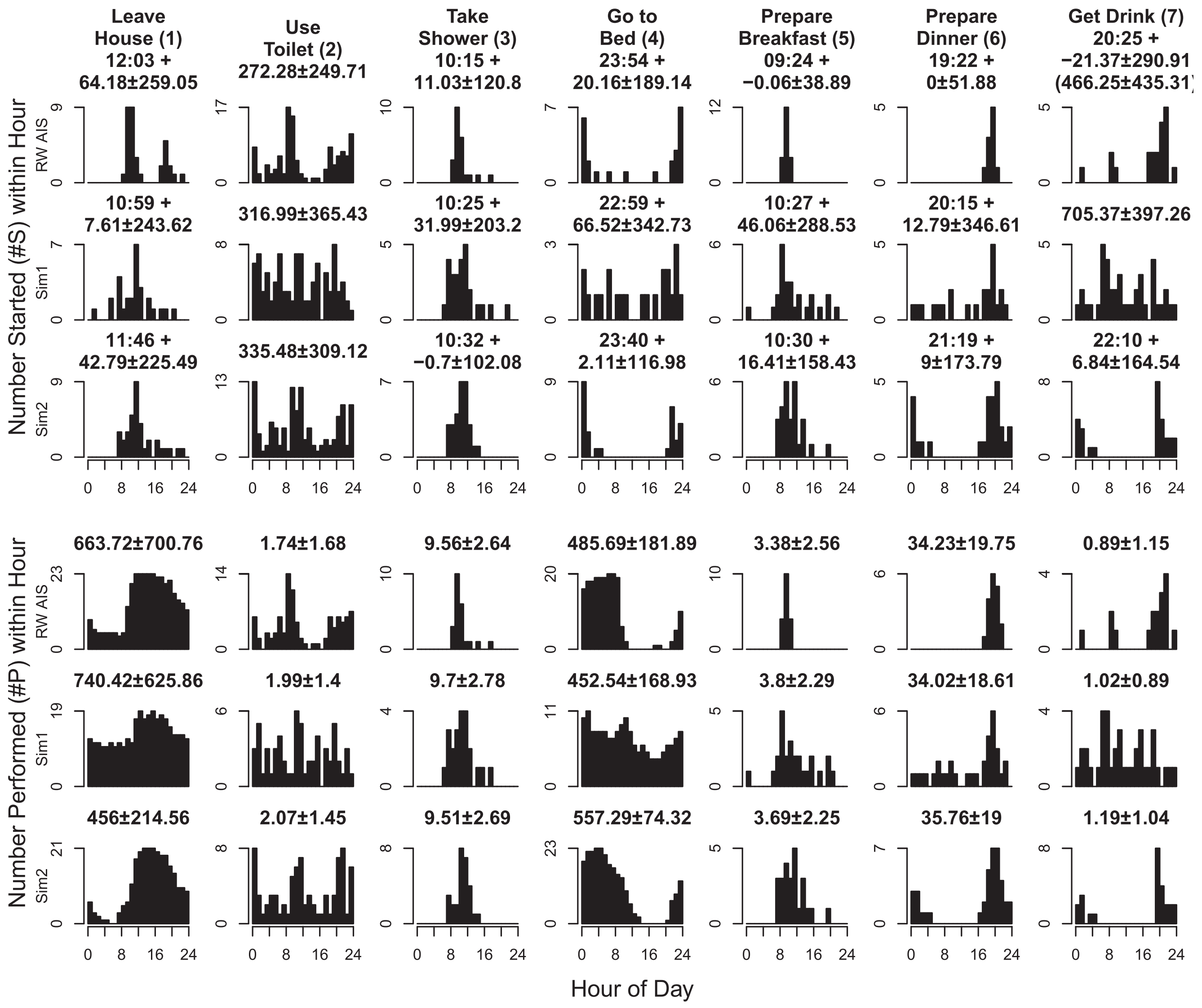


FIGURE 5. Overview of activities in first run. "RW" is an abbreviation for Real-World".

the models can be parametrised to reflect the behaviour of a given, real person (Requirement 1). To achieve this, we apply the parametrisation techniques described in Section V. ADLSim is activity agnostic by design, and should as such be capable of simulating individuals with a wide range of behaviours. It is therefore not important that the utilized real-world AIS is recorded from an individual with any particular traits or behaviours. Instead, it is important that we can show that the models resulting from the parametrisation process properly capture the behaviour present in the real-world AIS we decide to use as input. Very few real-world AIS are publicly available for this purpose. For this reason, we argue that the best criteria for selecting the real-world AIS is that it is of acceptable quality, and that it is widely enough referenced in the literature to facilitate comparable results. Based on these criteria, we decide to use the Ubicomp dataset from Van Kasteren *et al.* [29][3] as the real-world AIS. The dataset contains the sequence of activities in the home of a 26 year old man during a period of nearly 28 days. The recording was performed as non-intrusively as possible using a bluetooth headset such that when a button on the headset is pressed, it captures a time-stamped recording of the voice of the wearer denoting the start and end times of the execution of one of seven pre-defined activities. The selected activities are chosen based on the Katz Index of Independence in ADL [14]. This index is commonly used in Health Care to assess the cognitive and physical ability of elderly people. To enable a one-to-one comparison between this real-world AIS and the simulated AIS from ADLSim, we first truncate the real-world AIS such that it contains a whole number of

[3]Available from *https://sites.google.com/site/tim0306/datasets*

TABLE 1. AM parametrisation.

Experiment	Activity ID(s) (descr.)	% ind. var.	ToD (hh:mm)	ITD (min)	Duration (min)
Sim1	1 (leave house)	38.20	18:16	-70.63 ± 255.63	663.72 ± 700.76
	2 (use toilet)	60.50	Type 1 activity	334.47 ± 303.34	1.74 ± 1.68
	3 (take shower)	8.70	11:45	0.00 ± 240.86	9.56 ± 2.64
	4 (go to bed)	0.00	Always in sequence	0.00 ± 0.00	485.69 ± 181.89
	5 (prepare breakfast)	100.00	09:24	-0.06 ± 38.89	3.38 ± 2.56
	6 (prepare dinner)	100.00	19:22	0.00 ± 51.88	34.23 ± 19.75
	7 (get drink)	100.00	Type 1 activity	466.25 ± 435.31	0.89 ± 1.15
	$2 \rightarrow 4 \rightarrow 2$		23:32	14.43 ± 190.51	
	$3 \rightarrow 1$		10:09	10.29 ± 111.72	
Sim2	1	36.60	17:36	-68.68 ± 257.95	272.63 ± 176.26
	4	0.00	Always in sequence	0.00 ± 0.00	557.18 ± 74.21
	7	100.00	20:18	0.49 ± 98.07	0.92 ± 1.28
	2, 3, 5, 6	As in Sim1	As in Sim1	As in Sim1	As in Sim1
	$2 \rightarrow 4 \rightarrow 2$		23:32	-0.11 ± 77.14	
	$3 \rightarrow 1$		10:09	10.29 ± 111.72	Act. 1: 525.85 ± 169.75
NonRand	1 to 7	As in Sim1		As in Sim1, but with SD = 1.0	
RandPost RandSusp RandTerm	1 to 7	100.00	Type 1 activities	Uniform[0.0:60.0]	Uniform[0.0:60.0]
HypoSim	1 (ToiletShort)	100.00	Type 1 activity	300.00 ± 120.00	5.00 ± 2.00
	2 (ToiletLong)	100.00	Type 1 activity	960.00 ± 240.00	10.00 ± 5.00
	3 (EveningRoutine)	0.00	Always in sequence	Always first in sequence	30.00 ± 15.00
	4 (Sleep)	0.00	Always in sequence	Always in sequence	480 ± 15.00
	5 (MorningRoutine)	0.00	Always in sequence	00.00 ± 00.00	45.00 ± 30.00
	6 (Breakfast)	0.00	Always in sequence	00.00 ± 00.00	30.00 ± 10.00
	7 (Lunch)	100.00	12:00	0.00 ± 15.00	30.00 ± 10.00
	8 (Dinner)	0.00	Always in sequence	Always first in sequence	45.00 ± 20.00
	9 (EveningSnack)	0.00	Always in sequence	180.00 ± 60.00	10.00 ± 5.00
	10 (GroceryShopping)	0.00	Always in sequence	Always first in sequence	45.00 ± 30.00
	11 (Exercise)	100.00	10:00	0.00 ± 30.00	30.00 ± 1.00
	12 (StoreFood)	0.00	Always in sequence	00.00 ± 00.00	5.00 ± 3.00
	13 (Socialize)	100.00	Type 1 activity	1440.00 ± 480.00	120.00 ± 90.00

	14 (AfternoonNap)	0.00	Always in sequence	45.00 ± 10.00	— *Days 0 to 50 (P1):*
					DD = 45.00 ± 10.00
					— *Days 50 to 100 (P2):*
					Avg. of DD = 45.00 $\rightarrow$ 180.00

	15 (TakeMedication)	100.00	10:00 15:00 20:00	— *Days 0 to 50 (P1):* SD of ITD = 100.00 $\rightarrow$ 1.0	2.00 ± 1.00
	$8 \rightarrow$ (9 and 14) $10 \rightarrow 12$		16:45 Type 1 sequence	0.00 ± 30.00 1440.00 ± 720.00	

	3 (EveningRoutine)
	$\rightarrow$ 4 (Sleep)
	$\rightarrow$ 5 (MorningRoutine)
	$\rightarrow$ 6 (Breakfast)
			— *Days 0 to 50 (P1):*
			21:30
			— *Days 50 to 100 (P2):*
			21:30 $\rightarrow$ 03:30
				0.00 ± 45.00	

	8 (Dinner) $\rightarrow$ 14 (AfternoonNap)		16:45	0.00 ± 30.00	

HybridSleep	$2 \rightarrow 4 \rightarrow 2$	As in Sim2	— *Days 0 to 50 (P1):*
			23:32 (As in Sim2)
			— *Days 50 to 100 (P2):*
			23:32 $\rightarrow$ 05:32
				As in Sim2	As in Sim2

	1, 2, 3, 4, 5, 6, 7, $3 \rightarrow 1$	As in Sim2			

24-hour periods lasting from midnight at the beginning of Day 1 to midnight at the end of Day 25. Then we discard the first and last days, to account for the above mentioned shortening of the simulated AIS for a steady state analysis. The final real-world AIS used in our evaluation contains AI for 24 entire days.

Due to the non-deterministic outcome of combining probabilistic instigation times and AR, it is not given a priori that the simulated AIS preserves the key characteristics of the real-world AIS. We determine how accurately the simulated AIS reflects the real-world AIS by means of well-defined metrics introduced in the next section, that capture the similarity of the two AIS in terms of (1) per-activity summary statistics, (2) their day-to-day variability, and (3) the average difference between days in the real-world AIS and days in the simulated AIS.

Based on the sequence diagrams for the real-world AIS in Figure 4, we include two AI nexi in Sim1 and Sim2. The AI for Activity 4 are always dependent and part of the nexus $2 \rightarrow 4 \rightarrow 2$, while Activities 1, 2, and 3 have both dependent and independent AI variants. In Sim1, instigation times and durations are estimated without any preliminary outlier removal, and dependent and independent AI vairants always use the same DD. In Sim2, we improve the parametrisation in both aspects. Outliers are manually removed for Activities 1 (four abnormally long-lasting AI, i.e., longer than 23 hours, mostly during weekends), 4 (four abnormally short AI, i.e., less than 6 hours, mostly during day time), and 7 (four abnormally early AI, i.e., before 12 am). This results in significantly changed distribution parameters, and Activity 7 is more accurately modelled as a Type 1 activity. We also find that the durations of dependent and independent AI variants of Activity 1 differ significantly, and therefore model these with separate DD in Sim2. The AR postpone is used for all pairs of activities.

2) RandPost, RandTerm, RandSusp AND NonRand

Realistic behaviour (Requirement 1) involves a certain amount of day-to-day variability. We show that the simulated AIS from Sim1 and Sim2 exhibit realistic amounts of probabilistic behaviour by comparing their day-to-day variability with that in the real-world AIS. This comparison requires first to determine what we mean by large and small day-to-day variations. We therefore conduct additional experiments where we maximise and minimise day-to-day variability in simulations that include the same number of AM and days as in Sim1 and Sim2. We perform three experiments with highly random behaviour, called RandPost, RandSusp and RandTerm, collectively referred to as Rand*, and one experiment with minimized variability, called NonRand. NonRand is identical to Sim1 except that all SD are set to 1 minute to minimise variability. In Rand*, we maximise variability by using for ITD and DD uniform distributions across values that range from 0 and 60 minutes. The resulting AIS exhibit one meaningful notion of maximum and minimum variability among all 24 days, forming the basis to determine whether Sim1 and Sim2 exhibit realistic degrees of variability.

The real-world AIS used in these studies is relatively sparse, i.e., there are many periods where no activity is performed and the activities are selected relatively crudely, i.e., only seven activities. By comparison, the most recent activity real-world AIS by Kasteren et al. contains up to 16 activities [30]. For this reason, the number of activity conflicts in our simulation is limited, preventing a proper study of the impact of AR. In this context, Rand* and NonRand fulfill a second purpose: they are designed to yield a very high activity density, i.e., with almost no idle periods. As explained in Section III-B, this increases the frequency of activity conflicts, and thus the frequency of AR invocations. In Sim1 and Sim2, only 64.6% and 57.6% of instigations need to be resolved with AR, respectively. In contrast, in RandPost, RandTerm, and RandSusp, 99%, 99%, and 90% of instigations must be resolved with AR, respectively. Simulations with random behaviour also involve a significantly higher number of instigations, further increasing the frequency of AR invocations. This allows us to properly study the impact of AR on simulated AIS. We use the same AR for all activity pairs within each experiment. To investigate the choice of AR impacts the results, we apply different AR in the three different experiments with random behaviour. The AR postpone, suspend and terminate are used in RandPost, RandSusp and RandTerm, respectively.

3) HypoSim AND HybridSleep

ADLSim is not limited to simulate behaviour found in real-world AIS. For instance, in Rand* and NonRand we simulate behaviour with artificially high and low degrees of variation. This does however not suffice to properly evaluate ADLSim in terms of Requirements 2 and 3, which requires that we also demonstrate that ADLSim (1) is not restricted to simulate activities found in any one given real-world AIS, (2) can simulate the (gradual) onset of realistic, user-defined, hypothetical behaviour, i.e., specific hazards, achievements or anomalies, and (3) can seamlessly merge such behaviours into otherwise normal behaviour that is, e.g., learned from a real-world AIS.

We have partly addressed Requirement 2 and 3 in earlier work to test AAL software in the Health Care domain with ADLSim [16]. We constructed a purely hypothetical personality in terms of 15 daily activities, and introduce either instantly or gradually two different events that are of interest in the Health Care domain: (1) the gradual onset of a sleep anomaly, and (2) instantaneous immobility resulting in the cessation of all subsequent activity. In the current work, we conduct two additional experiments, HypoSim and HybridSleep to further evaluate ADLSim in terms of Requirement 2 and 3. HypoSim is based on modified versions of the models in [16] while HybridSleep is based on the models in Sim1 and Sim2.

In HypoSim, we simulate two hypothetical, long-term developments in behaviour that are of special interest in

Health Care, i.e., a gradually improving self-medication routine, and the gradual onset of a sleep anomaly. We simulate 150 days divided into three 50-day periods P1, P2, and P3. During P1, we gradually reduce the SD of the ITD of Activity 15 (TakeMedicines) from 100 minutes to 1 minute. We thereby simulate an improvement in the precision of taking medicines at the three designated ToD 10:00h, 15:00h, and 20:00h. We leave the SD at 1 minute for the remainder of the simulation to clearly show the resulting improved self-medication routine. During P2, we gradually introduce the sleep disorder as a combination of a gradual, but significant shift in sleeping patterns and a significant increase in overall amount of sleep. The central activities are Activity 4 (Sleep) and Activity 14 (AfternoonNap), which are instigated as part of the nexi $3 \rightarrow 4 \rightarrow 5 \rightarrow 6$ and $8 \rightarrow 14$, respectively. We gradually adjust two parameters, i.e., we shift the ToD of Sleep by 6 hours from 21:30h to 3:30h, and increase the mean duration of AfternoonNap from 45 minutes to three hours. Since Sleep is instigated as part of the nexus beginning with Activity 3 (EveningRoutine), the 6-hour shift is achieved by adjusting the ToD of EveningRoutine. In P3, we keep parameter values constant to clearly show the resulting adverse sleeping pattern.

HybridSleep is designed to demonstrate that we can seamlessly merge hypothetical events of interest, i.e., a sleep disorder, into normal behaviour based on a real-world AIS. We use the same activities and parametrisation as in Sim2, but gradually shift the ToD of the nexus $2 \rightarrow 4 \rightarrow 2$ that includes the activity "go to bed" (with ID 4). We simulate 150 days divided into three periods P1, P2, and P3, that unfold as in HypoSleep, i.e., P1 contains normal behaviour, P2 the gradual shift of the ToD of Activity 4 by +6 hours, and P3 a stable, but adverse sleeping pattern. In both HypoSim and HybridSleep, we perform one simulation run using the random seed 1.

B. METRICS

We assess the realism of ADLSim by comparing simulated AIS with the AIS from a real person used for parametrisation. We perform the comparison based on well-defined metrics. In order to understand the results, we perform both mathematical and visual analysis.

The challenge of formulating proper metrics to compare simulated and real AIS is two-fold. First, we must define metrics that allow the comparison of AIS in terms of individual activities, i.e., that describe the overall similarity between the AIS in terms of when, how often, and for how long the different activities are performed. Second, we must define metrics that allow to compare the activities within individual days. The latter is needed for two reasons: (1) we need to measure how similar the day-to-day variability within a simulated AIS is to the day-to-day variability within the corresponding real-world AIS, and (2) we need to determine how similar, on average, the days in a simulated AIS are to days in the corresponding real-world AIS.

1) METRICS TO COMPARE INDIVIDUAL ACTIVITIES

The summary statistics presented in Section V-A capture the essential characteristics of activities within an AIS. These statistics are used to parametrise simulation models using a real-world AIS. However, due to the influence of AR, such statistics obtained from a simulated AIS ais_{sim} will deviate from those obtained from the real-world AIS ais_{real} that ais_{sim} is based on. By comparing statistics for ais_{sim} with those for ais_{real}, we obtain a quantitative measure of the degree to which ais_{sim} resembles ais_{real} in terms of when, how often and for how long individual activities are performed. These statistics are calculated across all n AI $\{ai_{a,0}, \ldots, ai_{a,n-1}\}$ for a given activity a. We compute the following six summary statistics for a: ToD_a, $\overline{ToDDev_a}$, and $\sigma(ToDDev_a)$ to describe the distribution of start times for Type 1 activities, $\overline{IAD_a}$ and $\sigma(IAD_a)$ to describe the distribution of the IAD for Type 2 activities, and $\overline{DD_a}$ and SD $\sigma(DD_a)$ to describe the distribution of the duration of activities. These metrics are defined as follows:

$$\begin{aligned}
ToD_a &= \frac{\theta_{ToD_a}}{2\pi P}\\
\overline{ToDDev_a} &= \frac{\sum_{i=0}^{i<n}(ai_{a,i}.st - ToD_{a,i})}{n}\\
\sigma(ToDDev_a) &= \frac{\sum_{i=0}^{i<n}\sqrt{(ai_{a,i}.st - ToD_{a,i})^2}}{n}\\
\overline{IAD_a} &= \frac{\sum_{i=0}^{i<(n-1)}(ai_{a,i+1}.st - ai_{a,i}.st)}{n-1}\\
\sigma(IAD_a) &= \frac{\sum_{i=0}^{i<(n-1)}\sqrt{(ai_{a,i+1}.st - ai_{a,i}.st)^2}}{n-1}\\
\overline{DD_a} &= \frac{\sum_{i=0}^{i<n}(ai_{a,i}.et - ai_{a,i}.st)}{n}\\
\sigma DD_a &= \frac{\sum_{i=0}^{i<n}\sqrt{(ai_{a,i}.et - ai_{a,i}.st)^2}}{n}
\end{aligned}$$

where θ_{ToD_a}, P and $ToD_{a,i}$ are defined as in Section V-A.

Since the above metrics are merely summary statistics, they leave out important information about the shape of the distribution of AI throughout the day. We use visual inspection for this purpose, which is a commonly applied technique to assess the similarity of two distributions. This requires proper metrics with values that, once plotted in a histogram, facilitate this visual inspection. The first metric, $\#S_{ais,a,t_1,t_2,P}$ (abbreviated as $\#S$), denotes the number of times an activity a is started during a given time period during the day, or more precisely, the number of times it is started between two points in time t_1 and t_2 (e.g., 9 AM and 10 AM) in a repeating period P (e.g., 24 hours). Plotting these values in a histogram with one interval per hour of the day unveils the shape of the distribution of start times of a given activity. The second metric, $\#P_{ais,a,t_1,t_2,P}$ (abbreviated as $\#P$), denotes the number of times some portion of activity a is performed within such a period. The corresponding histogram for this metric unveils the shape of the distribution of hours in which some portion of the activity is performed. When visualised, $\#P$ provides an intuitive summary of both the start times and the durations of the activity. These two metrics are defined as follows:

$$
\begin{aligned}
\#S_{ais,a,t_1,t_2,P} = |\{&ai_x \in ais : ai_x.a = a \\
&\wedge t_1 \leq st_{x,P} < t_2\}| \\
\#P_{ais,a,t_1,t_2,P} = |\{&ai_x \in ais : ai_x.a = a \\
&\wedge [(t_1 \leq st_{x,P} < t_2) \\
&\quad \vee (t_1 \leq et_{x,P} < t_2) \\
&\quad \vee \ (st_{x,P} \leq t_1 \wedge et_{x,P} > t_2) \\
&\quad \vee \ (st_{x,P} < t_1 \wedge st_{x,P} > et_{x,P}) \\
&\quad \vee \ (ai_x.et - ai_x.st >= P)]\}|
\end{aligned}
$$

where $st_{x,P} = ai_x.st \bmod P$ and $et_{x,P} = ai_x.et \bmod P$ are the start and end times of an AI ai_x within a repeating time period consisting of $P \in \mathbb{N}$ time units, and $t_1, t_2 \in \{x \in \mathbb{N} : 0 < x \leq P\}$ are two points in time P. $\#S_{ais,a,t_1,t_2,P}$ is the sub-set of the AI $\{ai_x \in ais : ai_x.id = a\}$ that start between t_1 and t_2, and $\#P_{ais,a,t_1,t_2,P}$ is the sub-set of AI where at least some portion of the AI falls between t_1 and t_2, i.e., where the AI either (1) starts between t_1 and t_2, (2) ends between t_1 and t_2, (3) starts before t_1 and ends after t_2, (4) starts before t_1, and wraps around the 24 hour period such that $st_{x,P} > et_{x,P}$, or (5) the AI lasts more than P time units, meaning that it must have been performed between t_1 and t_2. The data for the final histograms are obtained by computing these metrics for t_1 and t_2 set to the beginning and end of the 24 different hours of the day, and P set to the number of time units in 24 hours.

2) METRICS TO COMPARE INDIVIDUAL DAYS

We need to measure the difference between days for two purposes. First, we need to quantify the day-to-day variability within one AIS. Second, we need to quantify the difference between a simulated AIS ais_{sim} and the corresponding real-word AIS ais_{real}.

The difference between days is calculated using the Levenshtein distance [17]. Since the Levenshtein distance only applies to sequences of symbols, the days in our AIS must first be represented as such sequences. To achieve this, each 24-hour day in an AIS is first divided into a sequence of T discrete time-intervals of fixed length $T/24$ *hours*. Each day is then represented as a tuple $\langle id_0, \ldots, id_{T-1} \rangle$, called an ID-map, where id_i is the ID of the activity that occupies most of time interval i. When selecting the duration of the intervals, one must consider two important factors. First, the longer the intervals are, the less representative the Levenshtein distance will be, because activities with a duration shorter than half of the interval duration $T/24$ *hours* will not be accounted for in the ID-map. Conversely, the shorter the intervals are, the more time is needed to compute the Levenshtein distance, which is particularly important given that the algorithm to compute this distance runs in quadratic time [4]. When creating ID-maps, it is thus important to strike the proper trade-off to keep the interval length as short as possible without incurring impractically large running times for the computation of the Levenshtein distance. Experiments show that using 30-second intervals, i.e., $T = 24$ hours / 30 seconds $= 2880$, achieves a proper trade-off with the implementation[4] and the 1.6 GHz quad core Intel i5 computer used in this work. The Levenshtein distance $l_{a,b}$ between two ID-maps $a = (id_{a,0}, \ldots, id_{a,T-1})$ and $b = (id_{b,0}, \ldots, id_{b,T-1})$, where a and b represent two different days, is defined as follows:

$$l_{a,b} = lev_{a,b}(|a|, |b|)$$

where

$$
lev_{a,b}(i,j) = \begin{cases} min(i,j) \text{ if } min(i,j) = 0 \\ \text{otherwise:} \\ min \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1_{id_{a,i} \neq id_{b,j}} \end{cases} \end{cases}
$$

where $1_{id_{a,i} \neq id_{b,j}}$ is the indicator function

$$
1_{id_{a,i} \neq id_{b,j}} = \begin{cases} 0 & \text{if } id_{a,i}.a = id_{b,j}.a \\ 1 & \text{otherwise.} \end{cases}
$$

We can now use $l_{a,b}$ to compute the average difference between all pairs of days within one AIS, to measure the probabilistic day-to-day variation with that AIS, or between pairs of days obtained from two separate AIS, to measure the difference between two AIS in terms of individual days. Let $\{a\} = \{day_{a,0}, \ldots, day_{a,n-1}\}$ and $\{b\} = \{day_{b,0}, \ldots, day_{b,m-1}\}$ denote sets of ID-maps for all days in two AIS a and b. The metrics $\overline{l_{\{a\},\{b\}}}$ and $\sigma(l_{\{a\},\{b\}})$ denote the mean and SD across the Levenshtein distances between the pairs of ID-maps resulting from all possible ways to combine an ID-map from $\{a\}$ with an ID-map from $\{b\}$. These are defined as follows:

$$
\overline{l_{\{a\},\{b\}}} = \frac{\sum_{x=0}^{x<n} \sum_{y=0}^{y<m, y \neq x} l_{day_{a,x}, day_{b,y}}}{(n-1)m}
$$

$$
\sigma(l_{\{a\},\{b\}}) = \frac{\sum_{x=0}^{x<n} \sum_{y=0}^{y<m, y \neq x} \sqrt{(l_{day_{a,x}, day_{b,y}} - \overline{l_{\{a\},\{b\}}})^2}}{(n-1)m}
$$

These metrics quantify the average and SD of the number of edit-operations (insertions, deletions or substitutions) required to make an ID-map $day_{a,x} \in \{a\}$ equal another ID-map $day_{b,x} \in \{b\}$. We quantify the day-to-day variability within one AIS by computing the mean and SD of the distance among days taken only from that AIS, i.e., by setting $\{a\} = \{b\}$. The result can thereafter be compared with the probabilistic day-to-day variation within a different AIS, i.e., to compare a simulated AIS ais_{sim} with an AIS ais_{real} from a real person. In other words, we compare $\overline{l_{\{ais_{sim}\},\{ais_{sim}\}}}$ with $\overline{l_{\{ais_{real}\},\{ais_{real}\}}}$ and $\sigma(l_{\{ais_{sim}\},\{ais_{sim}\}})$ with $\sigma(l_{\{ais_{real}\},\{ais_{real}\}})$.

We quantify the difference between two AIS by determining how different, in average, a randomly chosen day

[4]A version of https://rosettacode.org/wiki/Levenshtein_distance#Java was modified to work with our ID-maps.

from a simulated AIS is to a randomly chosen day from the corresponding real-world AIS. For this purpose, we introduce two additional metrics that are defined as follows:

$$\overline{ais_{sim} \leftrightarrow ais_{real}} = |\overline{l_{\{ais_{sim}\},\{ais_{real}\}}} - \overline{l_{\{ais_{real}\},\{ais_{real}\}}}|$$
$$\sigma(ais_{sim} \leftrightarrow ais_{real}) = |\sigma(l_{\{ais_{sim}\},\{ais_{real}\}}) - \sigma(l_{\{ais_{real}\},\{ais_{real}\}})|$$

The first metric denotes the difference between a simulated and a real AIS in terms of the mean day-to-day variance, and the latter denotes their differences in terms of the SD of the day-to-day variance. If both are close to 0, we can conclude that the overall difference between days in ais_{sim} and ais_{real} is similar to the difference among days within ais_{real}. Most importantly, we design these metrics to enable the evaluation of the realism of our simulated AIS in one very essential aspect: the closer $\overline{ais_{sim} \leftrightarrow ais_{real}}$ and $\sigma(ais_{sim} \leftrightarrow ais_{real})$ are to 0, the more difficult it is to distinguish a randomly chosen day from a simulated AIS from a randomly chosen day from a real-world AIS, in terms of the difference in the location and duration of activities.

Although $l_{a,b}$ provides a quantitative measure on day-to-day variability, it does not define what is a "small" or "large" difference in variability. For this purpose, we use the results from the Rand* and NonRand experiments. We compute a minimized and a maximized value for the distance between the real-world AIS and a simulated AIS by computing the distances between the real-world AIS and (1) a highly non-random AIS $ais_{NonRand}$ from the experiment NonRand, resulting in the minimized value, and (2) a highly random AIS ais_{Rand*} from Rand*, resulting in the maximized value.

C. RESULTS

1) PER-ACTIVITY COMPARISON

The purpose of our first two experiments, Sim1 and Sim2, is to assess the realism of ADLSim, i.e., the similarity between simulated AIS and the corresponding real-world AIS. For ease of comparison, we present the per-activity summary statistics from a simulated AIS side by side with those from the real-world AIS used for parametrisation. Figure 5 shows seven columns and six rows of histograms. There is one column per activity, and two rows per activity in the AIS ais_{sim1} from Sim1 (Rows 1 and 4), two rows per activity in the AIS ais_{sim2} from Sim2 (Rows 2 and 5), and two rows per activity in the real-world AIS ais_{real} (Rows 3 and 6). The histograms for ais_{sim1} and ais_{sim2} contain information about the first of 10 simulation runs. Each bar in the histograms in Row 1-3 present $\#S_{ais,a,t_1,t_2,P}$ with t_1 and t_2 set to the beginning and end of each of hour in the 24 hours period P (x-axes). The bars in the histograms in Rows 4-6 present $\#P_{ais,a,t_1,t_2,P}$ with the same t_1 and t_2. In Rows 1-3, we present above each histogram for activity a $\overline{IAD_a}$ (in minutes) and $\sigma(IAD_a)$ (in minutes) if a is a Type 1 activity and ToD_a, $\overline{ToDDev_a}$ (in minutes), and $\sigma(ToDDev_a)$ (in minutes) if a is a Type 2 activity. Since Activity 7 is modelled as a Type 2 activity in Sim1 and a Type 1 activity in Sim2, we present both types of information above its histogram in Row 1 (Type 1 information in parenthesis). In Rows 4-6, we present above each histogram the mean $\overline{DD_a}$ and SD $\sigma(DD_a)$ of the duration of activity a.

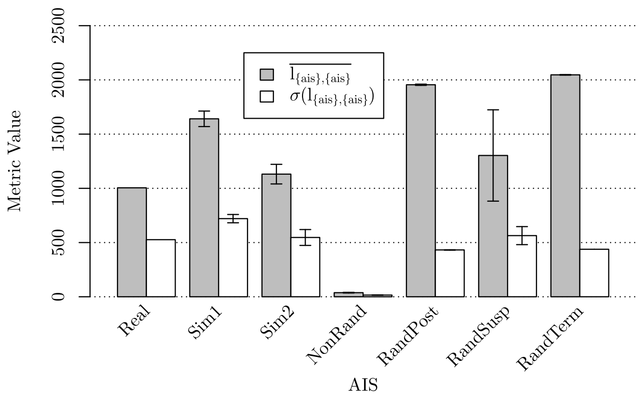


FIGURE 6. Comparison of day-to-day variability between simulated and real-world AIS.

2) PER-DAY COMPARISON

We calculate one $\overline{l_{ais,ais}}$ and one $\sigma(l_{ais,ais})$ for the real-world AIS as well as for each AIS from 10 simulation runs with different seeds per experiment. For the simulations, we present in Figure 6 the mean (boxes) and SD (error bars) of $\overline{l_{ais,ais}}$ (grey boxes) and $\sigma(l_{ais,ais})$ (white boxes) across the 10 simulation runs, in addition to the corresponding values from the real-world AIS.

We calculate $\overline{ais_{sim} \leftrightarrow ais_{real}}$ and $\sigma(ais_{sim} \leftrightarrow ais_{real})$ for each AIS ais_{sim} from the 10 runs per simulated experiment and for the real-world AIS ais_{real}. However, instead of using ais_{sim} and ais_{real} directly, we perform preliminary filtering of the AIS to extract seven AIS per ais_{sim} and ais_{real}, each of which only contain the AI for a given activity a. That is, for each activity a we end up with one $ais_{sim,a} = \{ai_x \in ais_{sim} : ai_x.a = a\}$ per simulation experiment, and one $ais_{real,a} = \{ai_x \in ais_{real} : ai_x.a = a\}$ from the real-world AIS. By comparing these, we achieve a detailed, per-activity comparison of the day-to-day variability between simulated and real-world AIS. The results are presented in Figure 7. The histogram to the left shows mean values for $\overline{ais_{sim,a} \leftrightarrow ais_{real,a}}$ with $sim \in \{RandPost, Sim1, Sim2\}$ from the experiments from Sim1, Sim2, and RandPost, and where a is one of the seven activities in the AIS. The histogram to the right shows the mean values for $\sigma(ais_{sim,a} \leftrightarrow ais_{real,a})$ with sim and a set in the same way. Here, lower values are better, since values close to 0 indicate a high degree of similarity between simulated and real AIS. We chose RandPost for comparison since it uses the same AR as in Sim1 and Sim2, but in contrast has highly random AIS.

3) VISUAL PRESENTATION OF AIS

Since the important characteristics of the AIS from Sim1, Sim2, and real-world AIS are extensively captured in Figures 5, 6, and 7, a visualisation of the raw content in the AIS from Sim1, Sim2, and the real-world AIS is excluded from the main text, and instead included in Appendix A.

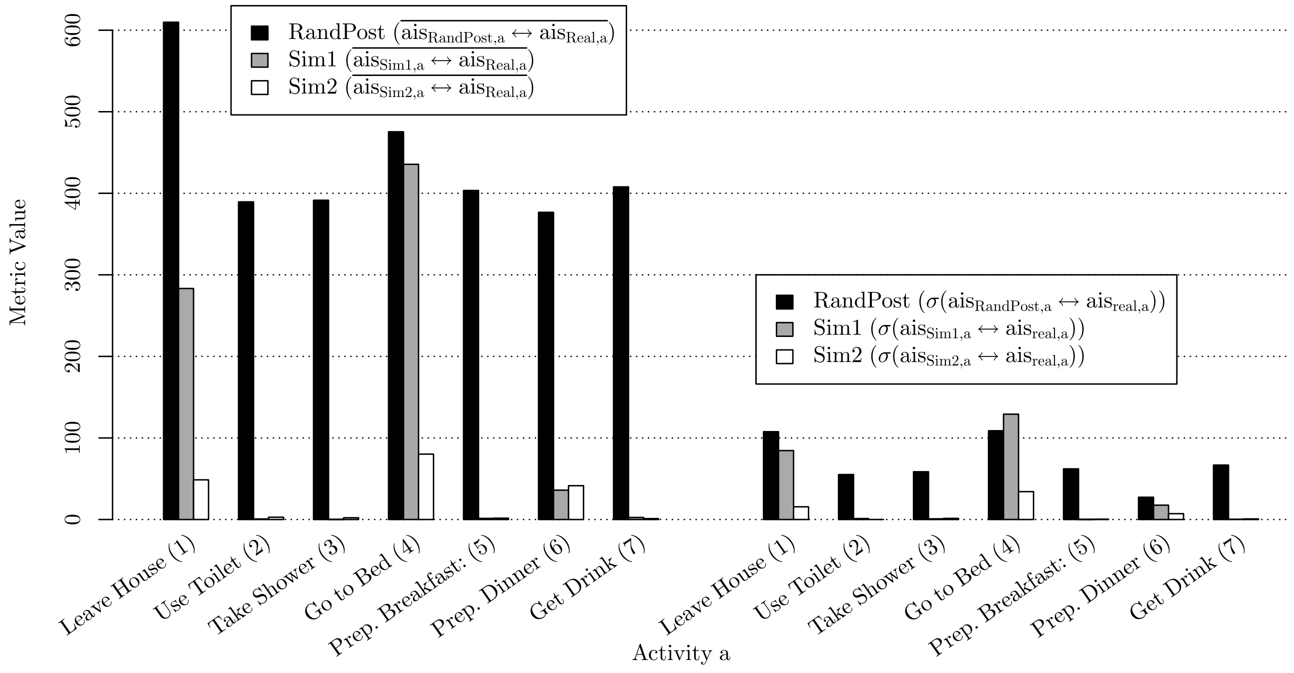


FIGURE 7. Values for $\overline{ais_{sim} \leftrightarrow ais_{real}}$ and $\sigma(ais_{sim} \leftrightarrow ais_{real})$, where $sim = RandPost, Sim1$ *and* $Sim2$. The lower the values are, the more similar the respective simulated AIS are to the real-world AIS.

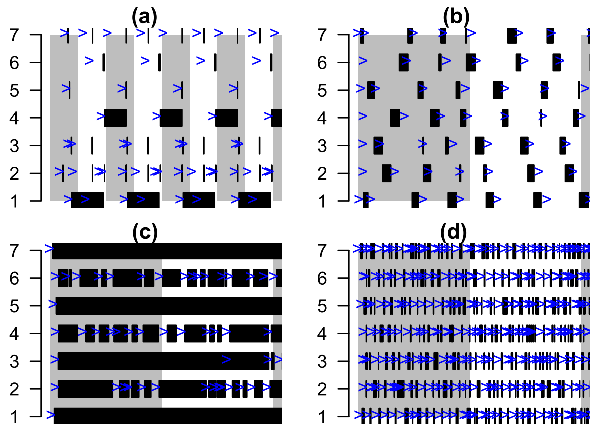


FIGURE 8. Snapshots of experiments with varying probabilistic behaviour. (a) NonRand, Run 1, 4 days. (b) RandPost, Run 1, 1 day. (c) RandSusp, Run 1, 1 day. (d) RandTerm, Run 1, 1 day.

For Rand* and NonRand, it is sufficient to present a snapshot of the corresponding AIS in order to obtain a proper understanding of their most important characteristics, since these characteristics are similar for the remaining simulated days. Figure 8 presents snapshots for NonRand (Figure 8 (a)), RandPost (Figure 8 (b)), RandSusp (Figure 8 (c)), and RandTerm (Figure 8 (d)). The y-axes denote the ID of seven different activities, and the x-axes represents time across one complete day for Rand*, and four days for NonRand. The black boxes show the AI, and the blue arrows their instigation times.

The results for HypoSim and HybridSleep are presented in Figures 9 and 10, respectively. We present a complete AIS for each experiment where the x-axes represent time and the y-axes the 24 hours within each simulated day. The boxes across the y-axes show the AI across a given day. The AI for all activities except the central ones in each experiment are coloured light grey. The central activities in HypoSim are TakeMedicine (black circles), Sleep and AfternoonNap (blue), the four activities Breakfast, Lunch, Dinner and EveningSnack (green), and Exercise (red). The central activity in HybridSleep is Sleep (blue).

For HypoSim, we also show to the right of the AIS in Figure 9 a snapshot for Days 50 and 125, to clearly show the impact of the altered sleep routine. The right-arrows denote instigation times, and the boxes show the AI. We highlight three key observations per day, i.e., D50-A, D50-B and D50-C for Day 50, and D125-A, D125-B and D125-C for day 125, which are analysed in the next section.

D. RESULT ANALYSIS

1) REQUIREMENT 1 (REALISM)

The summary statistics above the histograms in Figure 5 have similar values for the simulated and real-world AIS, which indicates that the AM properly capture the mean and SD of the start times and durations of activities. Such summary statistics nevertheless leave out information that is essential to fully assess the realism of the simulated AIS. For instance, the mean $\overline{DD_4} = 452.54$ minutes and SD $\sigma(DD_4) = 168.93$ minutes of the duration of Activity 4 is significantly affected by outliers, resulting in values that are closer to the values for the real-world AIS in Sim1 than in Sim2. A closer investigation of the histograms, and the values in Figure 7

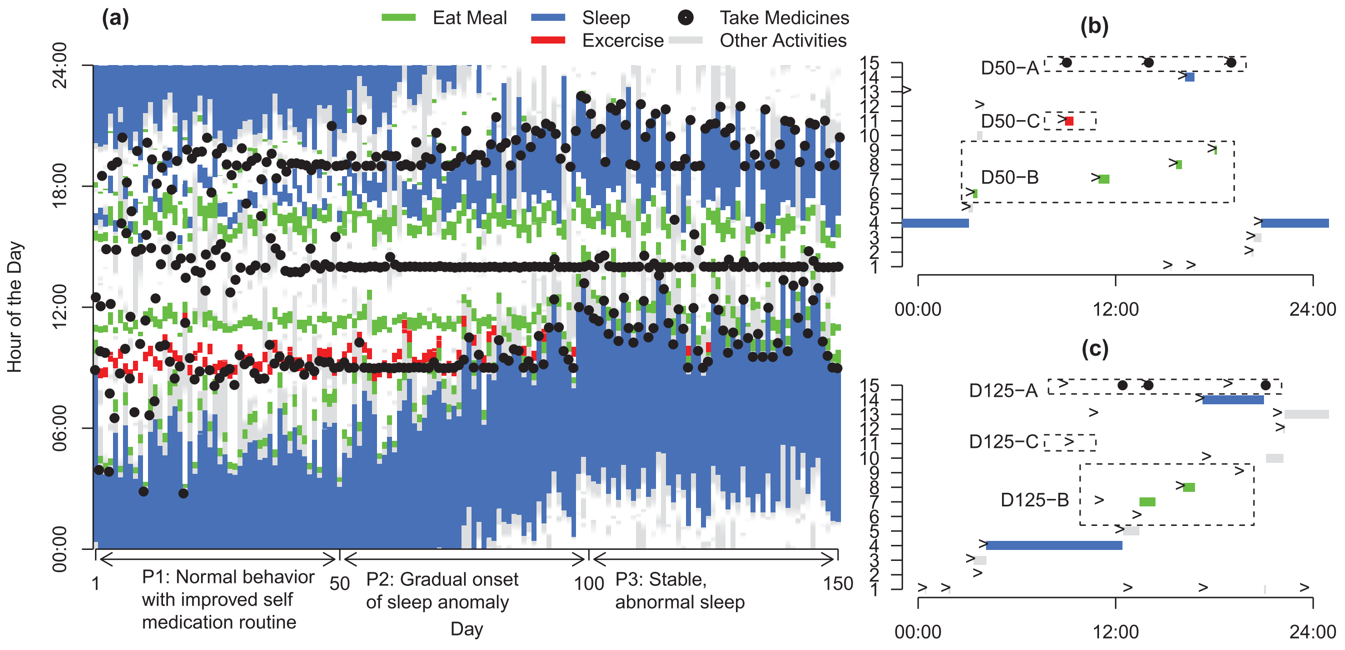


FIGURE 9. Experiment HypoSim. (a) HypoSim, 150 Days Seed: 1. (b) HypoSim, Day: 50, Seed: 1.

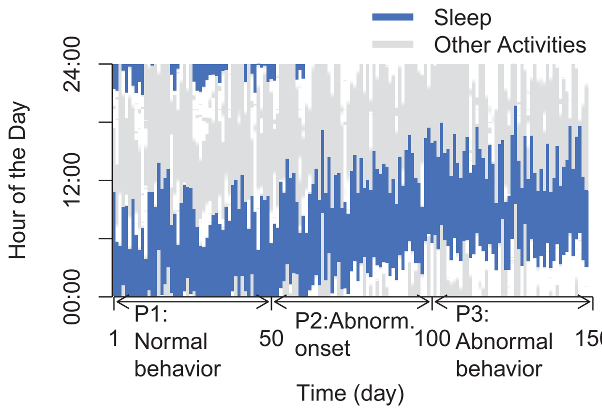


FIGURE 10. Experiment HybridSleep, 150 days, seed 1.

(discussed below), is required for a proper evaluation of the realism of the AIS.

The histograms for Sim1 and Sim2 generally resemble those in the real-world AIS, indicating that our AM are realistic. Certain visible differences are however introduced for two main reasons: (1) we use normal distributions for all ITD, and (2) we use the same AR, postpone, to resolve all activity conflicts. In the presence of outliers, normal distributions become significantly dispersed, resulting in much more variability in the simulated AIS than in the real-world AIS. This effect is particularly pronounced for Activities 1 and 4 in Sim1 due to the absence of outlier removal. The effect is exacerbated in the histograms in the three bottom rows, since they show the result of combining two normal distributions: one for the instigation time and one for the duration of the activity. We see significant improvements for Sim2 after outlier removal, in both the top three and the bottom three rows of the histograms, indicating that in Sim2 we have more accurately captured the overall statistical characteristics of the activities in the real-world AIS.

Figure 6 shows that AIS from Sim1 ($\overline{l_{\{ais_{Sim1}\},\{ais_{Sim1}\}}} = 1641 \pm 71.56$ and $\sigma(l_{\{ais_{Sim1}\},\{ais_{Sim1}\}}) = 720 \pm 38.31$) has significantly higher day-to-day variability than that for Sim2 ($\overline{l_{\{ais_{Sim2}\},\{ais_{Sim2}\}}} = 1130.96 \pm 90.47$ and $\sigma(l_{\{ais_{Sim2}\},\{ais_{Sim2}\}}) = 546.85 \pm 73.17$), and that the one from Sim2 has nearly the same day-to-day variability as the real-world AIS ($\overline{l_{\{ais_{real}\},\{ais_{real}\}}} = 1004.8$ and $\sigma(l_{\{ais_{real}\},\{ais_{real}\}}) = 526.43$). The difference between Sim1 and Sim2 is explained by the above-mentioned effect of outliers causing a dispersion of the normal distributions, and the fact that we use the AR postpone to resolve all activity conflicts. Together, these factors yield a slightly higher day-to-day variability than what is introduced by the ITD alone. We expect these inaccuracies to be reduced by a more elaborate choice of distributions and AR. We furthermore see that the day-to-day variability for NonRand is much lower ($\overline{l_{\{ais_{NonRand}\},\{ais_{NonRand}\}}} = 37.81 \pm 3.19$ and $\sigma(l_{\{ais_{NonRand}\},\{ais_{NonRand}\}}) = 15.75 \pm 1.6$) than in Sim1 and Sim2. The snapshot of NonRand in Figure 8 shows why it exhibits such a low value for $l_{\{ais\},\{ais\}}$, i.e., all days are nearly identical. In contrast, the variability for RandPost ($\overline{l_{\{ais_{RandPost}\},\{ais_{RandPost}\}}} = 1955.1 \pm 5.66$ and $\sigma(l_{\{ais_{RandPost}\},\{ais_{RandPost}\}}) = 431.92 \pm 1.69$) and RandTerm ($\overline{l_{\{ais_{RandTerm}\},\{ais_{RandTerm}\}}} = 2046.42 \pm 2.83$ and $\sigma(l_{\{ais_{RandTerm}\},\{ais_{RandTerm}\}}) = 438.13 \pm 0.56$) are significantly higher than in Sim1 and Sim2. The value for RandSusp, however, is lower than that for Sim2. This is explained by the choice of AR in RandSusp, and is discussed further in Section VI-D3. Compared to the values for NonRand, RandPost, and RandTerm, we see that the AIS from Sim2 exhibits variability that is close to the real-world AIS. This shows that (1) we can adjust day-to-day variability at will, and (2) the simulator can accurately capture the probabilistic behaviour found in the real world as long as outliers are removed.

The results in Figure 7 quantify the difference between simulated and real-world AIS. Therefore, lower values are better. They show clearly that values from RandPost significantly differ from Sim2 for all activities, and from Sim1 for most activities. For Activities 2, 3, 5, and 7, the mean $\overline{ais_{sim2} \leftrightarrow ais_{real}}$ and mean $\sigma(ais_{sim} \leftrightarrow ais_{real})$ never exceeds 3 and 2, respectively. The corresponding values from RandPost are almost 400 and above 50, respectively. We see the equivalent difference between RandPost and Sim1 for these four activities. For Activities 1 and 4, the values for Sim2 are slightly higher, but nevertheless significantly lower than for RandPost. The same is not true for Sim1 where we see very high values for Activities 1 and 4. These results reflect what is found by visual inspection of Rows 4-6 in Figure 5. First, the shapes of the histograms for Activites 1 and 4 for Sim1 are significantly more dispersed than those for the real-world AIS, resulting in high values in Figure 7. Second, those for Sim2 are much more similar to those for the real-world AIS, explaining the much lower values in Figure 7. As mentioned, these differences between Sim1 and Sim2 are caused by the presence of outliers in the real-world AIS. These observations strengthen our conclusion that (1) ADLSim can perform realistic simulation of AIS, also in terms of day-to-day variability, and (2) outlier removal is important to obtain such realistic results in ADLSim.

2) REQUIREMENTS 2 (SIMULATING HYPOTHETICAL BEHAVIOUR) AND 3 (ACTIVITY AGNOSTICISM)

The results from experiments HypoSim and HybridSleep clearly show that (1) arbitrary activities can be simulated in ADLSim, and (2) hypothetical events can be introduced in a highly controlled manner by scheduling simulation events to change parameters of distribution functions at specific points in virtual time. In both experiments, the change in behaviour is introduced gradually to demonstrate that we can simulate long-term trends, which is often a key challenge for AAL systems. ADLSim can also simulate the sudden onset of events, e.g., instantaneous immobility due to a fall or unconsciousness, which is demonstrated in our previous work [16].

There are two key factors that enable flexible simulation of hypothetical behaviour with arbitrary activities. Activities can easily be added or removed by (1) adding the proper AM, and (2) extending M_{CR} amd M_{PAC} with one column and one row per new AM. Behaviour is modelled using probability distributions and AR, both of which can easily be added and modified to model any type of behaviour. The AM are executed in a discrete event simulator, e.g., allowing these changes to happen at any user-defined point in virtual time during simulation. In HypoSim, we demonstrate this in three ways. At the beginning of the simulation, medicines are taken at highly variable points in time. As a result of decreasing the SD of the ITD of TakeMedicines, we see a gradual reduction during P1 in the day-to-day variability in the time medicines are taken. By the time P2 starts, this SD has reached 1 minute, and medicines are taken almost precisely at 10:00h, 15:00h, and 20:00h (the subsequent deterioration of the self-medication routine in P2 and P3 is caused by the introduced sleep disorder, as explained in the next section). During P2, we introduce two additional changes in parallel, i.e., a gradual but significant shifting of nightly sleep and an increase in the duration of the afternoon nap. These results demonstrate one aspect of the flexibility of discrete event simulation, i.e., that any number of changes can be introduced in parallel or in series with ease. Furthermore, this flexibility is not restricted to simulations where all behaviour is hypothetical. As the results from HybridSleep demonstrate, we can also introduce purely hypothetical behaviour seamlessly into behaviour obtained from a real-world AIS. During P1, we simulate behaviour as obtained from the real-world AIS without modification. During P2, we introduce a gradual shift of the sleeping pattern. This is only made possible by modelling Sleep using parametric probability distributions that we can modify with simulation events at arbitrary points in virtual time.

3) THE IMPACT OF AR

AR are particularly influential in HypoSim where the effect of many different AR impacts several activities in different ways. Before we analyse HypoSim in detail, we shortly analyse the impact of AR in Sim1, Sim2, Rand*, NonRand, and HypoMeds in order to more easily understand the details of HypoSim.

The ITD in Sim1 are more dispersed than in Sim2 due to the absence of outlier removal. This probability dispersal results in an increased frequency of overlapping activities, and thus activity conflicts, in Sim1 compared to Sim2. As a result, AR are invoked most frequently in Sim1, i.e., in 64.6% of instigations compared to 57.6% in Sim2. Since we use the AR postpone for all activity conflicts in Sim1 and Sim2, we see that the start times in the simulations tend to be delayed compared to those in the real-world AIS, e.g., for Activities 3, 5, 6, and 7. Most activities are postponed by the two long-lasting Activities 1 (Leave House) and 4 (Go to Bed). For instance, Activities 3 (Take Shower) and 5 (Prepare Breakfast) are postponed by Activity 4 in average 33.2% and 40.3% of the times they are instigated, respectively. This results in up to an hour increase in the average start times compared to the real-world AIS, and stresses the importance of selecting appropriate AR for realistic ADL simulation.

The AIS from Rand* do not correspond to any particular real-world case, and the high activity density results in a high number of activity conflicts which in turn result in a high rate of AR invocations. Rand* are therefore ideal to study the impact of AR in the general case. The snapshots in Figure 8 show that RandTerm has a much higher AI frequency than RandSusp. This explains the low values for $\overline{l_{\{ais\},\{ais\}}}$ and $\sigma(l_{\{ais\},\{ais\}})$ for RandSusp seen in Figure 6, since a lower number of AI decreases the variability in the AIS. Notice how the postpone AR affects both NonRand and RandPost, frequently separating instigation and start times. This separation is not found for RandSusp and RandTerm where the AI is always started upon instigation, potentially interrupting any ongoing AI. RandSusp and RandTerm differ in one essential aspect: in RandSusp, interrupted AI are resumed upon completion of the interrupting AI, while in RandTerm, interrupted AI are terminated and re-scheduled for future instigation. This causes some AI to be suspended for very long periods of time in RandSusp, reducing significantly the number of AI alternations per time-unit, which in turn suppresses the variability in RandSusp. Which and how frequently AI are affected by this is highly sensitive to initial conditions, resulting in highly non-deterministic behaviour in RandSusp. This shows up as a very high SD for RandSusp in Figure 6. These results clearly demonstrate that the choice of AR has a significant impact on ADL simulation in general, especially with high-granular activity sets, i.e., when overall activities are simulated in detail using shorter, more specific sub-activities, and/or when the AIS is dense, i.e., with a high number of AI per time unit.

Our analysis of HypoSim helps to understand the importance of AR for concrete examples of interest in Home Care. The choice of AR has at least three severe consequences of the altered sleep pattern: reduced activity level, a reduced intake of meals and a deteriorated self-medication routine. The reduced activity level is seen in the AIS to the left in Figure 9. Exercise (in red) is performed increasingly seldom in Period P2, and almost never in Period P3. This is explained by Observations D50-C and D125-C to the right in Figure 9, for Days 50 and 125, respectively. D125-C shows that Exercise is instigated, but never started in Day 125 like in the other days when Exercise is omitted. We see that Exercise is instigated during the execution of Sleep, which is resolved by omitting Exercise. If the AR would have been set to postpone, the activity level would not have been affected.

The second consequence is a reduction in the number of meals (in green). The AIS shows that the number of meals is reduced from four in P1 to an average of two in P3. To understand this effect, consider Observations D50-B (for Day 50) and D125-B (for Day 125) in the snapshots. During Day 50, we see that breakfast, lunch, dinner, and the evening snack are started right after instigation (i.e., after the morning routine, around ToD 12:00h, around ToD 17:00h, and around three hours after dinner). During Day 125, only lunch and dinner are performed. The reason breakfast is omitted is explained by an interplay between two AR. Breakfast is first postponed during Sleep. This introduces a significant delay due to the shifted sleeping pattern. When it is finally scheduled to start, the activity Lunch is instigated. In this case, the selected AR causes breakfast to be omitted in favour of eating only one meal, i.e., Lunch, effectively merging the two meals. The EveningSnack meal is omitted, because it is instigated during the significantly extended AfternoonNap. As a result, two of four meals are omitted.

The final consequence is a deteriorated medication routine. Although the SD of the ITD of TakeMedication is minimised in P2 and P3, we clearly see a significant deviation between its start times (black circles) and the assigned ToD. TakeMedicines is instigated three times per day, i.e., 150 times per period. 59, 90, and 113 of these instigations cause activity conflicts in periods P1, P2, and P3, respectively. The increased frequency of activity conflicts is a result of the altered sleep pattern, i.e., the fraction of conflicts with Sleep and AfternoonNap increases from 13.6% in P1, to 44.4% in P2, and 76.1% in P3. All these conflicts are resolved by postponing TakeMedicines, since we assume that the person does not take medicines until he/she wakes up after sleep. Observations D50-A (for Day 50) and D125-A (for Day 125) show clearly this effect. On day 125, TakeMedicines is postponed during Sleep and AfternoonNap. Since Sleep is significantly shifted, and AfternoonNap is significantly delayed, the first and last AI of TakeMedicines are significantly delayed, while the second AI is unaffected and starts immediately upon instigation. This results in an extremely short duration between the two first AI, and an extremely long duration between the second and last AI. With different AR, the results would potentially look very different. For instance, if the AR suspend was used (e.g., to simulate the use of an alarm-clock that wakes the person up to take medicines) the trend of precise self-medication would persist throughout P2 and P3.

The findings in this section show that the choice of AR has a significant impact on simulated behaviour, in particular in aspects of particular interest for AAL systems, e.g., the activity level, and routines for eating and medication. Our results demonstrate that AR have a very important role in ADL simulation. ADLSim is to the best of the authors' knowledge the only simulator that provides the possibility to select different AR for different activity conflicts.

E. SUMMARY

We have evaluated ADLSim in terms of the requirements for ADL simulation discussed in Section III-A, i.e., realism, hypothetical behaviour, and activity agnosticism, and studied the impact of AR. The key findings from our experimental analysis are:

- **ADLSim produces realistic AIS**, i.e., they exhibit the necessary key properties like the non-deterministic day-to-day variations seen in real life. This facilitates reliable evaluation of ADL analysis software. The models in ADLSim can be parametrised to realistically reflect the behaviour in a real-world AIS using the technique proposed in Section V, as long as outliers are removed from the real-world AIS before the parameter values are extracted.
- **It is possible to simulate hypothetical behaviour**, including both the simulation of a complete, hypothetical person as well as introducing hypothetical behaviour

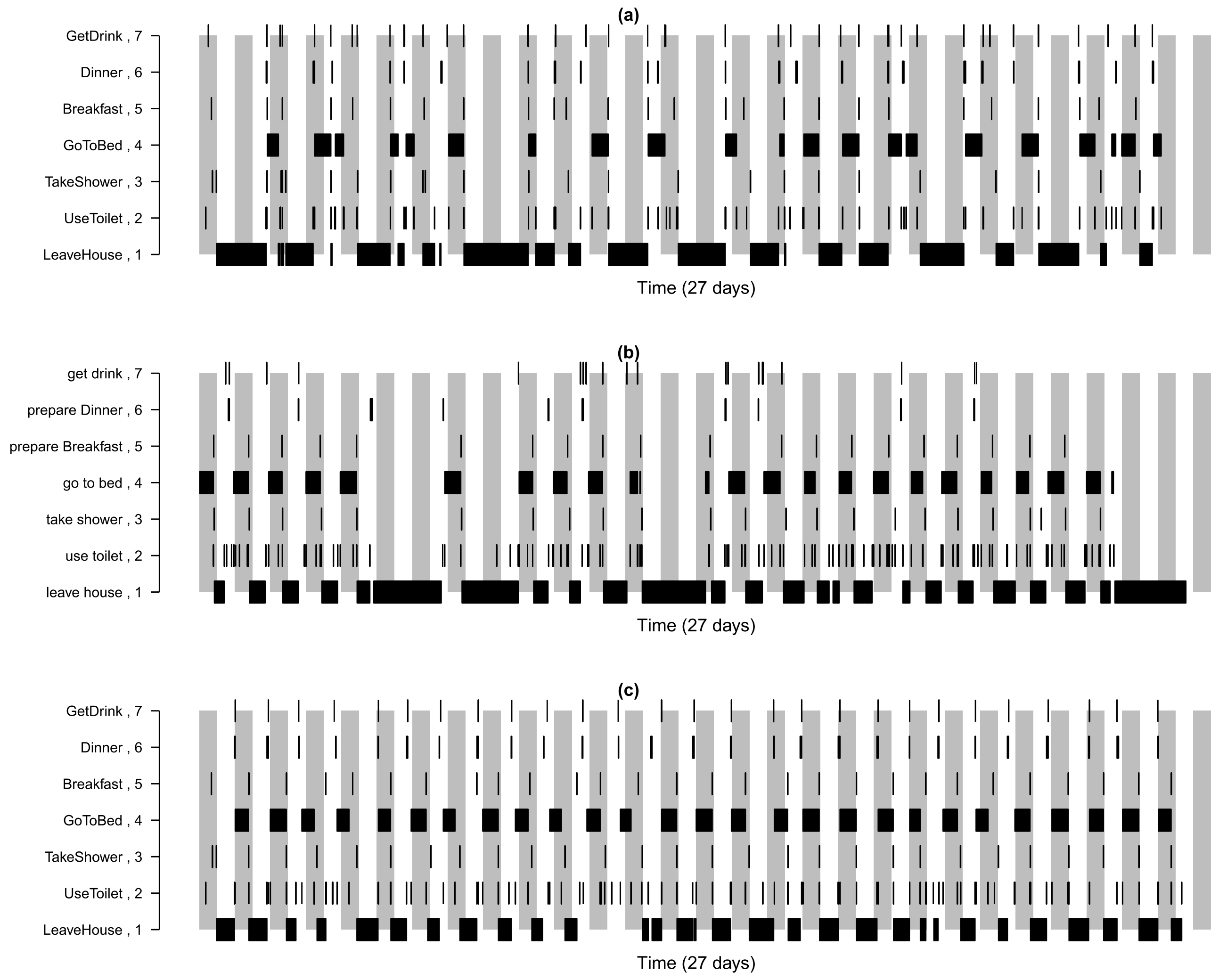


FIGURE 11. AIS from the first run in Sim1 (a) and Sim2 (c), and real-world AIS (b).

seamlessly into real behaviour obtained from a real-world AIS. This allows the evaluation of ADL analysis software targeting events like anomalies and hypothetical hazards or achievements that are rare or non-existent in available data sets with real-world AIS.

- We demonstrate in simulations with only hypothetical behaviour that **ADLSim is activity agnostic**, i.e., that we are not restricted to simulate a set of activities found in any particular real-world AIS. Activities can easily be added or removed in ADLSim by (1) adding and parametrising AM, and (2) extending M_{CR} and M_{PAC} with one column and one row per new AM.
- The **AR** to handle the activity conflicts between given pairs of activities **have a significant impact on simulation results**. This emphasises the importance of using different AR for different activity conflicts to enable realistic ADL simulation. To the best of the authors' knowledge, ADLSim is the only ADL simulator that supports this concept of AR.

VII. CONCLUSION

According to [20], approximately 50% of the elapsed time and more than 50% of the total cost of a programming project are expended in testing. The process of identifying and eliminating errors is important especially if the software is to be used for life critical applications like AAL. It is the goal of ADLSim to support developers and tester in this process by simulating ADL sequences to (1) substantially increase the set of test cases in form of AIS, and (2) simulate AIS that could not be captured in the real-world. As such ADLSim can substantially reduce effort and costs of testing ADL sequence analysis software (as for example experienced in our previous work [16]). However, the inherent tradeoff in testing economy between cost and effort and probability to find most errors in the software cannot be resolved by ADLSim. It is the decision of the developer and tester how detailed the test cases respectively AIS are, including the number and type of the activities considered. Therefore, ADLSim is designed to be activity agnostic and developers

TABLE 2. AR in M_{CR} used in HypoSim.

	Activity ID															
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
1	T	H	O	O	SU	O	P	P	SU|E	P	P	SU|E	P	P	P	P
2	T	T	H	O	SU	O	P	P	P	P	P	P	P	P	P	P
3	T	T	T	H	T	H	H	P	P	P	P	P	P	P	T	P
4	T	P	P	O	H	H	H	P	P	T	P	P	SU	SU	T	P
5	T	T	T	H	H	H	H	H	H	H	SU|E	SU|E	P	SU|E	P	P
6	T	P	P	H	H	H	H	O	O	H	P	P	P	O	P	P
7	T	P	P	O	P	P	O|R	H	O|R	T	P	P	P	O|R	P	P
8	T	P	P	P	P	P	T	T	H	T	P	P	P	O|R	P	P
9	T	P	P	O	O	O	O	O	O	H	P	P	P	O	O	P
10	T	P	P	P	P	P	P	P	P	P	H	P	H	P	P	P
11	T	P	P	O|R	O|P|R	P	P	P	P	P	P	H	P	P	O|P|R	P
12	T	P	SU|E	SU|E	P	SU|E	SU|E	SU|E	SU|E	SU|E	H	SU|E	H	SU|E	SU|E	H
13	T	P	P	P	O|R	P	P	P	P	P	P	P	P	H	P	P
14	T	P	P	O	P	P	P	P	P	P	P	P	P	P	H	P
15	T	P	P	SU|E	P	SU|E	SU|E	SU|E	SU|E	P	P	SU|E	P	P	P	H

and testers can choose which activities to use. Adding new activities to ADLSim is rather easy, in part because our models are not based on Markov models (as opposed to most related works).

ADLSim reproduces the behaviour found in the real world by combining probabilistic activity instigation times and durations with a novel concept called activity rules that capture how different individuals resolve activity conflicts. We provide a methodology to use a real AIS to instantiate models that capture the essential properties of the behaviour of a person. We can use these models to simulate real behaviour into which purely hypothetical behaviour can be seamlessly introduced. Our evaluation of ADLSim shows that our simulator is realistic and faithfully reproduces the behaviour found in real-world AIS, including the correct amount of variability and non-deterministic behaviour. We demonstrate that it is possible to simulate purely hypothetical behaviour (e.g., specific hazards or anomalies) and even introduce such behaviour into behaviour learned from a real-world AIS. Our experiments also show that activity rules can have a significant impact on simulation results, stressing the important role of activity rules in realistic ADL simulation. Possible future work includes evaluation with additional real-world AIS and automating model instantiation using data mining techniques like clustering and sequence mining. The simulator can also be improved in several aspects, e.g., by using more realistic probability distributions, by supporting composite activities and by determining instigation times based on motivation models.

APPENDIX A
FULL AIS

Figure 11 presents the full content of the real-world AIS (b) from [29] used in our experiments, and from the first run from Sim1 (a) Sim2 (c). The y-axes denote activity names and ID, and the x-axes the time. The black boxes denote the time periods each activity is executed. The time periods from 00:00 to 12:00 are shown in grey.

APPENDIX B
DETAILED EXPERIMENT PARAMETRISATION

Table 1 presents the AM parametrisation used in the eight experiments, and Tables 2 and 3 present the M_{CR} and M_{PAC} used in HypoSim, respectively.

In Table 1, from left to right, the columns present the experiment name, the ID and names of activities or activity nexi involved in the experiment, the fraction of independent activities for activities that have both dependent and independent AI vairants, the ToD for Type 2 activities, the ITD parameter values, and the DD parameters. Since DD determine activity durations, they only make sense for individual activities, i.e., not for activity nexi. The DD used for AI in the activity nexi are the same as those specified for the individual activities, except for Activity 1 in Sim2 (for the reasons discussed in V-B). The sequence 8 $\rightarrow$ (9 and 14) for HypoSim denotes that Activity 8 (Dinner) is followed by both Activity 9 (EveningSnack, after an average of 3 hours) and Activity 14 (AfternoonNap, after an average of 30 minutes).

TABLE 3. AR in M_{PAC} used in HypoSim.

	Activity ID															
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
1	RE	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
2	RE	RE	H	H	H	H	H	H	H	H	H	H	H	H	H	H
3	RE	RE	RE	H	H	H	H	H	H	H	H	H	H	H	H	H
4	RE	YP	YP	YP	H	H	H	H	H	H	H	H	H	H	H	H
5	RE	RE	RE	H	H	H	H	H	H	H	H	H	H	H	H	H
6	RE	YP	YP	YP	H	H	H	H	H	H	H	H	H	H	H	H
7	RE	YP	YP	RP	YP	YP	RE	H	H	H	H	H	H	H	H	H
8	RE	YP	YP	RP	YP	RP	RE	RR	H	H	H	H	H	H	H	H
9	RE	YP	RP	RP	YE	YP	YE	YE	YR	H	H	H	H	H	H	H
10	RE	YP	YP	RP	YP	YP	YP	RP	YP	RP	H	H	H	H	H	H
11	RE	YP	YP	RP	YR	YP	YP	RP	RP	YP	RP	H	H	H	H	H
12	RE	YP	RP	RP	RP	RP	RP	RP	RP	RP	H	RP	H	H	H	H
13	RE	YP	YP	YP	YR	YP	YP	YP	YP	YP	YP	RP	YP	H	H	H
14	RE	YP	YP	YE	YP	RE	YP	RP	RP	RP	YP	RP	YP	RP	H	H
15	RE	YP	YP	RP	YP	RP	RP	RP	RP	RP	RP	RP	YP	RP	RP	H

Table 2 present the AR in M_{CR} used in HypoSim. The bold numbers in the first row and column denote the activity ID of a newly instigated activity A_{new} and a current activity A_{cur}, respectively. A_{new} and A_{cur} have the same semantics as in the explanation of conflict resolution in Section IV-B. P implies to postpone A_{new} until the completion of A_{cur}. SU implies to start A_{new} and suspend the execution of A_{cur} until the completion of A_{new}. Combining SU with E using the logical and-operator (e.g., $SU|E$) implies to also postpone the time of completion of A_{cur} by the amount of time A_{cur} was suspended. This is often the most realistic choice, since otherwise A_{cur} will be completed as scheduled before conflict resolution, in effect simulating that A_{cur} was merged with, or executed in true parallel with, A_{new}. T implies to terminate A_{cur} and start A_{new}. O implies to omit A_{new} according to its omission probability. In the case A_{new} was omitted, it is rescheduled for instigation according to its ITD if O is combined logically with R (e.g., $O|R$). If A_{new} was not omitted, A_{new} will be suspended, or postponed if O is combined logically with P (e.g., $O|P$). H indicates to halt the simulation with feedback to the experimenter about the conflicting activities. This is useful when certain activity conflicts are either highly improbable, or even impossible, under certain AM parametrisations, and are thus likely results of mistakes in the parametrisation. The AR H can therefore be used during model verification to signal to the experimenter about such situations.

Table 3 presents the AR in M_{PAC} used in HypoSim. The numbers in the first row and column denote the activity ID of A_{cand} and C_i, respectively, with the same semantics as in the explanation of post-activity contention in Section IV-B. The AR H has the same purpose as in M_{CR}, i.e., to signal an error condition. Since activities are stored in the contending list in increasing order of activity ID, A_{cand} will always be lower than C_i. Therefore, AR in the top-right half of M_{PAC} should never be invoked, and are therefore set to H to indicate an error condition. The remaining AR are composed of two characters. The first determines whether the activity C_i should replace (R) the activity A_{cand} as the candidate for execution, or if C_i should yield (Y) such that A_{cand} remains as the candidate for execution. The second letter indicates what to do with the activity that was not selected as a candidate for execution. P indicates that it should be inserted into the contention list of the AI ai_{x+1} (see Section IV-B) that "wins" the post activity contention, to be re-considered for execution upon the completion of AI ai_{x+1}. E means that it is not inserted into the contention list of ai_{x+1}, meaning that it will not be considered for execution during the next post contention period. This also means that it will never be executed unless it is re-scheduled for instigation. To solve this, we include AR with the second letter R which ensures that the activity is re-scheduling for instigation according to its AM. The exception is with the activity "Idle", with activity ID 0, which is always executed whenever no other activity executes.

REFERENCES

[1] M. R. Alam, M. B. I. Reaz, and M. A. M. Ali, "A review of smart homes—Past, present, and future," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 1190–1203, Nov. 2012.

[2] H. Alemdar, T. L. M. van Kasteren, M. E. Niessen, A. Merentitis, and C. Ersoy, "A unified model for human behavior modeling using a hierarchy with a variable number of states," in *Proc. 22nd Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2014, pp. 3804–3809.

[3] T. A. Arentze and H. J. Timmermans, "A need-based model of multi-day, multi-person activity generation," *Transp. Res. B, Methodol.*, vol. 43, no. 2, pp. 251–265, 2009.
[4] A. Backurs and P. Indyk. (2014). "Edit distance cannot be computed in strongly subquadratic time (unless SETH is false)." [Online]. Available: https://arxiv.org/abs/1412.0348
[5] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Proc. Int. Conf. Pervasive Comput.*, 2004, pp. 1–17.
[6] T. R. Bennett, H. C. Massey, J. Wu, S. A. Hasnain, and R. Jafari, "Motion-synthesis toolset (MoST): An open source tool and data set for human motion data synthesis and validation," *IEEE Sensors J.*, vol. 16, no. 13, pp. 5365–5375, Jul. 2016.
[7] K. Bouchard, A. Ajroud, B. Bouchard, and A. A. Bouzouane, "SIMACT: A 3D open source smart home simulator for activity recognition," in *Advances in Computer Science and Information Technology*. Heidelberg, Germany: Springer, 2010, pp. 524–533.
[8] F. Cardinaux, S. Brownsell, D. Bradley, and M. S. Hawley, "A home daily activity simulation model for the evaluation of lifestyle monitoring systems," *Comput. Biol. Med.*, vol. 43, no. 10, pp. 1428–1436, 2013.
[9] S. Helal, J. W. Lee, S. Hossain, E. Kim, H. Hagras, and D. Cook, "Persim-simulator for human activities in pervasive spaces," in *Proc. 7th Int. Conf. Intell. Environ. (IE)*, 2011, pp. 192–199.
[10] S. S. Intille, K. Larson, J. Beaudin, J. Nawyn, E. M. Tapia, and P. Kaushik, "A living laboratory for the design and evaluation of ubiquitous computing technologies," in *Proc. CHI Extended Abstracts Hum. Factors Comput. Syst.*, 2005, pp. 1941–1944.
[11] S. S. Intille *et al.*, "Using a live-in laboratory for ubiquitous computing research," in *Proc. Int. Conf. Pervasive Comput.*, 2006, pp. 349–365.
[12] S. R. Jammalamadaka and A. Sengupta, *Topics in Circular Statistics*, vol. 5. Singapore: World Scientific, 2001.
[13] T. Janssen, C. Van Oers, L. van der Woude, and A. P. Hollander, "Physical strain in daily life of wheelchair users with spinal cord injuries," *Med. Sci. Sports Exerc.*, vol. 26, no. 6, pp. 661–670, 1994.
[14] S. Katz, T. D. Downs, H. R. Cash, and R. C. Grotz, "Progress in development of the index of ADL," *Gerontologist*, vol. 10, no. 1, pp. 20–30, 1970.
[15] B. Kormányos, B. Pataki, "Multilevel simulation of daily activities: Why and how?" in *Proc. IEEE Int. Conf. Comput. Intell. Virtual Environ. Meas. Syst. Appl. (CIVEMSA)*, Jul. 2013, pp. 1–6.
[16] S. Kristiansen, T. Plagemann, and V. Goebel, "Smooth and crispy: Integrating continuous event proximity calculation and discrete event detection," in *Proc. 10th ACM Int. Conf. Distrib. Event-Based Syst.*, 2016, pp. 153–160.
[17] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Phys. Doklady*, vol. 10, no. 8, pp. 707–710, 1966.
[18] W. Liu, Y. Shoji, and R. Shinkuma, "An indoor-movement simulator for ambient assisted living systems," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2015, pp. 1–6.
[19] M. Mubashir, L. Shao, and L. Seed, "A survey on fall detection: Principles and approaches," *Neurocomputing*, vol. 100, pp. 144–152, Jan. 2013.
[20] G. Myers, *The Art of Software Testing*, vol. 15, 2nd ed. Hoboken, NJ, USA: Wiley, 2004.
[21] N. Noury and T. Hadidi, "Computer simulation of the activity of the elderly person living independently in a health smart home," *Comput. Methods Programs Biomed.*, vol. 108, no. 3, pp. 1216–1228, 2012.
[22] E. Osnes *et al.*, "Consequences of hip fracture on activities of daily life and residential needs," *Osteoporosis Int.*, vol. 15, no. 7, pp. 567–574, 2004.
[23] F. Pitta, T. Troosters, M. A. Spruit, V. S. Probst, M. Decramer, and R. Gosselink, "Characteristics of physical activities in daily life in chronic obstructive pulmonary disease," *Amer. J. Respiratory Crit. Care Med.*, vol. 171, no. 9, pp. 972–977, 2005.
[24] T. Plötz, N. Y. Hammerla, A. Rozga, A. Reavis, N. Call, and G. D. Abowd, "Automatic assessment of problem behavior in individuals with developmental disabilities," in *Proc. ACM Conf. Ubiquitous Comput.*, 2012, pp. 391–400.
[25] S. Ranasinghe, F. A. Machot, and H. C. Mayr, "A review on applications of activity recognition systems with regard to performance and evaluation," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 8, pp. 1–21, 2016.
[26] D. Roggen *et al.*, "Collecting complex activity datasets in highly rich networked sensor environments," in *Proc. 7th Int. Conf. Netw. Sens. Syst. (INSS)*, Jun. 2010, pp. 233–240.
[27] M. Skubic, G. Alexander, M. Popescu, M. Rantz, and J. Keller, "A smart home application to eldercare: Current status and lessons learned," *Technol. Health Care*, vol. 17, no. 3, pp. 183–201, 2009.
[28] E. M. Tapia, S. S. Intille, and K. Larson, "Activity recognition in the home using simple and ubiquitous sensors," in *Proc. Int. Conf. Pervasive Comput.*, 2004, pp. 158–175.
[29] T. Van Kasteren, A. Noulas, G. Englebienne, and B. Kröse, "Accurate activity recognition in a home setting," in *Proc. 10th Int. Conf. Ubiquitous Comput.*, 2008, pp. 1–9.
[30] T. L. van Kasteren, G. Englebienne, and B. J. Kröse, "Human activity recognition from wireless sensor network data: Benchmark and software," in *Activity Recognition in Pervasive Intelligent Environments*. Heidelberg, Germany: Springer, 2011, pp. 165–186.
[31] G. Virone, B. Lefebvre, N. Noury, and J. Demongeot, "Modeling and computer simulation of physiological rhythms and behaviors at home for data fusion programs in a telecare system," in *Proc. 5th Int. Workshop Enterprise Netw. Comput. Healthcare Ind. (Healthcom)*, 2003, pp. 111–117.
[32] G. Virone, N. Noury, and J. Demongeot, "A system for automatic measurement of circadian activity deviations in telemedicine," *IEEE Trans. Biomed. Eng.*, vol. 49, no. 12, pp. 1463–1469, Dec. 2002.

STEIN KRISTIANSEN received the B.E. degree in computer engineering from the Oslo University College in 2006 and the M.E. and Ph.D. degrees in computer science from the Department of Informatics, University of Oslo, in 2008 and 2013, respectively. Since 2014, he has been with the Department of Informatics, University of Oslo, as a Post-Doctoral Researcher and a Researcher. His research interests include modeling and simulation, data mining, machine learning, and complex event processing in the context of operating systems, mobile systems, and e-health.

THOMAS P. PLAGEMANN received the Dr.Sc. degree in computer science from the Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, in 1994. He has been a Professor with the University of Oslo, Oslo, Norway, since 1996, where he currently leads the Research Group in Distributed Multimedia Systems, Department of Informatics. He has published over 150 papers in peer reviewed journals, conferences, and workshops in his field. His research interests include protocol architectures and middleware solutions for multimedia communication and mobile systems, future Internet, and multimodal sensor systems and event processing. He is a member of the Association for Computing Machinery (ACM). He received the Medal of ETH Zurich in 1995. He serves as an Associate Editor for the *ACM Transactions on Multimedia Computing, Communication, and Applications*, as an Area Editor for the *Computer Communications* (Elsevier), and as the Editor-in-Chief for the *Multimedia Systems* (Springer).

VERA GOEBEL received the M.S. degree in computer science from the University Erlangen-Nuremberg, Erlangen, Germany, in 1989, and the Ph.D. degree in computer science from the University of Zurich, Zurich, Switzerland, in 1994. She has been a Professor with the Department of Informatics, University of Oslo, Oslo, Norway, since 1997. She has published over 150 papers in peer reviewed journals, conferences, and workshops in her field. Her research interests are data management, distributed systems, future Internet, multimodal sensor systems, and complex event processing.