



**HAL**  
open science

# Generating and Evaluating Data of Daily Activities with an Autonomous Agent in a Virtual Smart Home

Lysa Gramoli, Julien Cumin, Jérémy Lacoche, Anthony Foulonneau, Bruno Arnaldi, Valérie Gouranton

► **To cite this version:**

Lysa Gramoli, Julien Cumin, Jérémy Lacoche, Anthony Foulonneau, Bruno Arnaldi, et al.. Generating and Evaluating Data of Daily Activities with an Autonomous Agent in a Virtual Smart Home. ACM Transactions on Multimedia Computing, Communications and Applications, 2024, 21, pp.1-23. 10.1145/3665331 . hal-04803327

**HAL Id: hal-04803327**

**<https://hal.science/hal-04803327v1>**

Submitted on 25 Nov 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Generating and Evaluating Data of Daily Activities with an Autonomous Agent in a Virtual Smart Home

LYSA GRAMOLI, Univ Rennes, INSA Rennes, Inria, CNRS, Irista, France and Orange, France

JULIEN CUMIN, Orange, France

JÉRÉMY LACOCHE, Orange, France

ANTHONY FOULONNEAU, Orange, France

BRUNO ARNALDI, Univ Rennes, INSA Rennes, Inria, CNRS, Irista, France

VALÉRIE GOURANTON, Univ Rennes, INSA Rennes, Inria, CNRS, Irista, France

Training machine learning models to identify human behavior is a difficult yet essential task to develop autonomous and adaptive systems such as smart homes. These models require large and diversified amounts of labeled data to be trained effectively. Due to the high variety of home environments and occupant behaviors, collecting datasets that are representative of all possible homes is a major challenge. In addition, privacy and cost are major hurdles to collect real home data. To avoid these difficulties, one solution consists of training these models using purely synthetic data, which can be generated through the simulation of home and their occupants. Two challenges arise from this approach: designing a methodology with a simulation able to generate credible simulated data and evaluating this credibility. In this paper, we explain the methodology used to generate diversified synthetic data of daily activities, through the combination of an agent model to simulate an occupant, and a simulated 3D house enriched with sensors and effectors to produce such data. We demonstrate the credibility of the generated synthetic data by comparing their efficacy for training human context understanding models against the efficacy generated by real data. To achieve this, we replicate a real dataset collection setting with our smart home simulator. The occupant is replaced by an autonomous agent following the same experimental protocol used for the real dataset collection. This agent is a BDI-based model enhanced with a scheduler designed to offer a balance between control and autonomy. This balance is useful in synthetic data generation since strong constraints can be imposed on the agent to simulate desired situations while allowing autonomous behaviors outside these constraints to generate diversified data. In our case, the constraints are those imposed during the real dataset collection that we want to replicate. The simulated sensors and effectors were configured to react to the agent's behaviors similarly to the real ones. We experimentally show that data generated from this simulation is valuable for two human context understanding tasks: current human activity recognition and future human activity prediction. In particular, we show that models trained solely with simulated data can give reasonable predictions about real situations occurring in the original dataset. We also report experimental results regarding statistical analysis and C2ST to assess the credibility of generated data. We discuss the generality of our approach for evaluating the credibility of simulated data from their use as training data.

CCS Concepts: • **Computing methodologies** → **Simulation evaluation**.

Additional Key Words and Phrases: Synthetic data, autonomous agent, activities of daily living, human activity recognition

## 1 INTRODUCTION

Providing autonomous systems able to assist people in their daily lives, such as smart homes, requires a fine understanding of occupant behaviors to provide appropriate services, in particular for sensitive applications such as medical care at home. Most of these systems rely on machine learning models that require a large amount of labeled data. However, capturing representative data in such environments is a major difficulty: there are too many possible home settings (home layout, devices, etc.) and occupant behaviors (preferences, hobbies, etc.) to collect general datasets. In addition, their collection can involve privacy issues and is extremely costly in terms of time, money, and human resources.

A promising avenue to deal with this problem consists in training these models with only synthetic data which can be generated from the simulation of home environments in which virtual occupants perform daily activities [2, 8, 30]. The major challenge of this approach is to ensure that such synthetic data are sufficiently credible and diversified for real situations. In this paper, our definition of *credibility* is taken from the work of Avradinis et al. [5]: the behavior of an intelligent agent is credible when it exhibits coherence in their reaction and consistency when placed in similar situations. Credibility is distinguished from realism, which seeks to reproduce as exactly as possible the physical world. However, simulating an ultra-realistic behavior is complex, and in our use case, reaching such standards is unnecessary since the granularity of data produced by sensors is not high. We rather seek to produce a sufficiently credible behavior so that sensors react correctly and the produced data can be used to train models to recognize real situations.

To generate valuable synthetic data for smart home context understanding, two significant challenges must be addressed. **First, we need to design a simulation able to generate diversified and credible synthetic data of daily living.** The combination of a virtual occupant able to make decisions autonomously, with a simulated smart home able to capture data through virtual sensors and effectors are essential to reach this goal. This implies designing an environment able to accurately simulate interactions between a virtual human, simulated sensors, and appliances [2]. In this kind of environment, the behavior of sensors must be credible: the way they capture data should be similar to the way real sensors capture data [18]. Regarding the virtual occupant, autonomy is required to choose and perform its activities in a comparable way to what a human could do [5]. These autonomous aspects are essential to produce diversified and credible behaviors. Conversely, the virtual human must be controllable to perform desired situations to ensure proper coverage of expected situations. To correctly extend or replicate real activity datasets such as the one used in this paper [12], the virtual occupant must respect a calendar of activities to perform at a specific time. **These controllability aspects led us to favor agent-based approaches rather than data-based approaches such as Generative Adversarial Networks (GAN)[1, 19],** since these aspects impact the generated data and GAN-style approaches are not optimal to manage them, contrary to agents.

**The second significant challenge is the evaluation of the actual credibility of synthetic data.** One approach consists in directly comparing real data with simulated data [23, 25, 36, 37]. Addressing the first challenge detailed above is an *a priori* way to generate credible synthetic data, as the simulation model is designed with credibility in mind, as in [2, 23]. Addressing the second challenge is an *a posteriori* way to check that generated data is credible, through the use of specific metrics, as in [25, 36].

Few works in the literature focus on addressing both challenges at the same time. For the first challenge, few approaches develop agent models offering simultaneous autonomy and control [23, 30], which is essential to obtain diversified behaviors while imposing specific situations on the agent. In addition, few approaches propose both a sophisticated agent model and an environment with credible virtual sensors. Consequently, they produce data with little diversity compared to real situations. Some approaches choose to generate data without 3D environments [25, 36]. This



Fig. 1. Global view of the 3D simulator and the agent performing activities.

is insufficient in our case since we use sensors and effectors whose realistic behaviors are inherently 3-dimensional (such as presence sensors). 3D environments also provide finer granularity in the performed actions of virtual occupants and can thus offer more realistic data. We thus choose to use 3D environments, even though they are harder to implement.

Approaches addressing the second challenge often rely on statistical comparisons between synthetic and real data. However, few works study the behavior of context understanding systems when trained on such synthetic data, which is the actual end-use case for these data. They also rarely use multiple different metrics in experimental studies.

In this paper, **we propose a new approach to generate and evaluate credible synthetic data of human activities in smart homes.** Our approach tackles both previously mentioned challenges. **For the first challenge, we propose to generate synthetic data through the interaction of a virtual agent in a 3D smart home, with special care given to the credibility of interaction and the behavior of the agent, sensors, and effectors.** We use a BDI-based agent model enriched with a scheduler that was presented in our previous work [21] since it can consider both the constraints of the simulation protocol and those coming from the agent’s internal model of human needs. This agent model can thus be adjusted between autonomy and controllability. To improve the diversity of generated data, our model can also trigger interruptions during scheduled activities and choose equivalent activities randomly. Based on Unity<sup>1</sup>, we developed diverse 3D homes containing sensors and effectors able to produce synthetic data when the agent interacts with them. The home replicated in this work is shown in Figure 1.

**For the second challenge, we propose to evaluate the credibility of generated data by replicating the data collection protocol of a real dataset in our simulation environment.** We use the Orange4Home dataset [12] as a reference dataset for our replication strategy. To create Orange4Home, a real occupant performed daily activities for 4 weeks in a real instrumented smart home. The real apartment and sensors used for collecting this dataset were replicated accurately in our virtual environment. Our agent model was configured to replicate the data collection protocol set in Orange4Home, in particular about imposed activity schedules.

To evaluate our synthetic data, we combine several metrics to ensure the reliability of our results. We first compare the frequencies of sensors between real and synthetic data. The results of Classifier Two-Sample Test (C2ST) [31] are also reported to determine if synthetic data can be distinguished from real ones. We then compare the behavior of

<sup>1</sup>Unity: <https://unity.com>

human context understanding models when trained either on synthetic or real data. Current activity recognition and future activity prediction tasks are used for this experimental evaluation. We show that both machine learning models behave similarly (in terms of performance and impact of hyperparameters) when dealing with either synthetic or real data. Finally, we show that we can use only synthetic data to train such models to recognize or predict real situations with reasonable accuracy, compared to those trained with real data.

In Section 2, we discuss existing approaches as well as their limitations in our context. In Section 3, we present our 3D virtual environment, the behavior of virtual sensors, and the way an agent can interact with them. In Section 4, we present our agent model and its adaptability between autonomous and controllable behavior. In Section 5 we describe our replication strategy for credibility evaluation. In Section 6 we report our experimental results on synthetic data credibility evaluation, based on our replication strategy. Finally, in Section 7, we discuss the benefits and drawbacks of our replication strategy to evaluate the credibility of synthetic data, as well as its generalization to other tasks and contexts since our evaluation is purely experimental in this work.

## 2 RELATED WORK

In the introduction, two main challenges were put forward: developing simulation environments able to generate credible data, and proposing strategies to evaluate the actual credibility of such data. We assess state-of-the-art works on both of these challenges in this section.

### 2.1 Environments of simulation with an autonomous agent to generate credible human data

*2.1.1 Simulation environment.* In the literature, several works including agent models show more or less sophisticated smart environments. Some of them simulate large outdoor environments such as cities [7, 23, 24] whereas others simulate indoor spaces such as houses [2, 14, 30, 36]. We analyze these works to know whether these simulations are sufficiently developed to produce data as rich as real ones.

This requirement excludes approaches using abstract environments to generate activity schedules where only lists of objects and possible activities are given [4, 25, 33]. Even though they provide global information about used objects and agent’s routines, we ignore how the activity unfolded, when objects are used, and how the agent interacts with them. In addition, correctly simulating a smart home with this kind of environment is hard since some sensor behaviors, such as presence sensors, have their performance directly impacted by their location, as in the case of sensors used in the Orange4Home dataset [12] that we want to replicate. In the same way, we also exclude approaches using 2D environments [26, 36]. Even though these environments provide additional information about the movements of agents and the location of specific devices, this is still not enough for some databases such as Orange4Home using, for example, presence sensors whose height can impact performances. Moreover, all sensors using precise gestures, such as Kinect, or video recording, such as a camera used to create datasets based on visualization [35], cannot be simulated either.

This is why 3D smart home environments are optimal for our use case. Some existing approaches use these environments such as the work of Lee et al. [30], OPENSYS [2] or BIM Sim 3D [46] where a tool is proposed in each of them to construct our proper 3D smart home. OPENSYS is a 3D smart home simulation open-source platform built with the explicit purpose of generating datasets of daily activities. Here, real people are used to performing activities with the use of Virtual Reality. Both other approaches propose a 3D smart home where an agent performs daily activities. The kind of data produced by these simulations is sufficiently complete for the most of real databases and for our use case. However, although these environments are interesting in our case, we will see they do not cover all the requirements to successfully address both challenges, such as the credibility of their agent behavior or the data validation.

2.1.2 *Simulating autonomous agents to perform daily activities.* Human behavior simulation models are essential to generate credible daily activity data. To generate them and meet our challenges, we require agent models able to both offer autonomy to obtain credible behaviors and control to respect the user’s expectations. With this, the agent can respect schedules of activity, as in the Orange4Home dataset of Cumin et al. [12], while obtaining sufficient diversity and credibility through autonomy since the agent could satisfy its proper objectives [5]. In our previous work [21], we showed that existing agent models are usually either fully autonomous or fully controlled. For example, Finite State Machines [29], Petri-Nets [34] or Behavior Tree [8] are fully controlled approaches. We believe such approaches are not sufficient to offer credible and diverse data, compared to real human behavior. We are thus mostly interested in those offering at least partial autonomy.

Most autonomous approaches in the literature propose fully autonomous agents, for which it is difficult to impose some activities to generate specific data. This is the case for reactive-based models, where the agent reacts to changes in the simulation. Even though they are well suited to dynamic environments and multi-agent situations, we cannot easily impose strong constraints on the agent’s behavior. Among reactive-based approaches, we find the works of Avradinis et al. [5], De Sevin and Thalmann [14], and Lee et al. [30] the agent selects its activities according to its physiological needs and performs them in a 3D smart home. Kamara-Esteban et al. [25] also propose the MASSHA model where an agent uses a ToDo List to store activities to perform in a 2D Smart Home. Although MASSHA can manage time constraints, there is no guarantee that the start time and duration of activities are respected since they are selected just before being done. Finally, we can find architectures such as Belief-Desire-Intention (BDI) [41]. In BDI, a perception system interpreting the state of the world is modeled through *Beliefs*, the choice of possible goals is modeled through *Desires*, and the choice of predefined sequences of actions to satisfy them is done by *Intentions*. A lot of BDI implementations exist, giving an interesting adaptability for our use case. However, current implementations are not sufficient since they cannot manage strong constraints.

Other approaches proposing fully autonomous agents include reinforcement learning-based models such as the work of Jang et al. [23] where a double deep Q-network (DQN) is proposed for the selection of goals to be then executed by a Hierarchical Task Network (HTN) in a 3D virtual city. We also have cognitive-based models, such as Act-R and Soar [28] where rules and previous experiences are used to manage the agent behavior. Once again, imposing punctual strong constraints on agent behavior is difficult since the agent reacts to the situation according to past experiences.

Fully autonomous agents can also be obtained through planning-based approaches. Unlike previous approaches, it is easier to impose strong constraints, especially on resource use. This is the case for HTN [17], STRIPS [16], graph-based approaches [24], first principle planning [45], or meta-heuristics such as Genetics algorithms [10]. Unfortunately, most of them cannot accurately manage time constraints. In the approach proposed by Renoux et al. [36], where an agent is simulated in a smart house, constraint-based planning is used to schedule daily activities. The time is thus well managed. However, for all of these approaches, agents often have difficulties reacting to unexpected events that disrupt the plan during a simulation.

Finally, some approaches combine reactivity with planning. This is the case for Reactive-Planners [6, 42], for SMACH [3], for case-based planning [44] or for combinations between BDI models and a planner such as HTN [15] or First Principle Planning (FPP) [43, 45]. These approaches could offer flexibility between control and autonomy, but as discussed in [21], they rarely focus on this ratio and rarely propose sufficient validations (except for SMACH). Although rare, some approaches offer this flexibility, such as BIM SIM 3D proposed by Zhao et al. [46] where an agent in a 3D smart home can manage time constraints and needs. However, BIM SIM 3D does not give sufficient control since the agent’s decisions are based on probabilities.

## 2.2 Methods to validate the credibility of generated data

In this section, we present works that aim at validating the credibility of simulated human data. We find two main paradigms for asserting this credibility. Some contributions generate credible data with *a priori* arguments: the models used to generate data were constructed based on real data, which should lead to credible data by design. Other contributions assess credibility with *a posteriori* arguments: generative models are still based on real data, but credibility is evaluated through specific metrics comparing these data with real ones.

Among methods using *a priori* arguments, Jang et al. [23] proposed a deep Q-network trained on large datasets of social data, which is an *a priori* way to ensure credibility. However, only global surveys coming from a population containing the start times and duration of activities have been used to construct and verify credibility. This is not enough to know whether each agent can behave in credible ways in the choice of their activities, in the diversity of their choices, and in the execution of these activities. Similar studies on travel scheduling were proposed by Roorda et al. for the TASHA system [38]. In addition, both papers do not include the simulation of sensors to retrieve data. OpenSHS [2] also uses *a priori* arguments to construct and validate the credibility of generated data. Since action sequences were set directly by real humans, we could expect that the generated data would be more credible. However, people might not interact in a 3D environment like they would in a real setting. Moreover, requiring real recording sessions is costly and limits the possibilities of control and variability that we hope to obtain with agent models.

In the work of Renoux et al. [36], a 2D simulation is proposed where an agent can perform daily activities. The generated data is validated by *a posteriori* approach since real experimenters are involved to guess if the presented activity calendars are made of real or simulated activities. However, this type of assessment may involve bias or uncertainty due to the subjectivity of experimenters. In addition, people may have difficulty imagining what a real schedule would look like. This is why, even though collecting the subjective opinions of users is relevant, an additional objective method is necessary to validate the credibility of data. In the MASSHA model proposed by Kamara-Esteban et al. [25], where an agent model is proposed to perform activities in a smart home, an *a posteriori* approach is also proposed. Durations and frequencies of simulated activities and the behavior of virtual sensors are compared to real ones. Although the method is interesting and the reported results are statistically close to real ones, the credibility of generated data is difficult to evaluate since the environment is abstract, agent behaviors are conditioned to the definition of a TODO list and the behavior of sensors does not take into account their location and does not depend on direct interaction with the agent (their activation is a purely probabilistic process).

In the SMACH model proposed by Reynaud et al. [37, 40], the generated activity routines data produced in an abstract house are analyzed by *a priori* approaches (surveys and statistics methods to design the model) and *a posteriori* ones (microscale level assessment with 8 real occupants and macroscale level assessment with population surveys). The authors also suggest other ways to evaluate *a posteriori* the credibility of generated data by the use of clustering simulated samples: when clusters are found with both real and simulated samples, then real and simulated ones are hard to separate, and thus simulated data is considered credible. Even though these proposed validations can address our second challenge, SMACH is difficult to use in a database such as Orange4Home, since activities cannot be strictly imposed. In addition, no 3D environment is proposed in this work, limiting the correct simulation of some sensors.

SocialInteractionGAN [1] is a machine learning-oriented approach for generating simulated human activity data. In this work, Airale et al. use a Generative Adversarial Net (GAN) neural network for the generation of multi-person interaction sequences. The motivation is that an adversarial strategy can act as a way to enforce the credibility of generated data. Indeed, the discriminator part of the model is trained to distinguish between generated and real data,

forcing the generator to improve the realism of its generated data. To evaluate the quality of generated sequences, they propose metrics based on the Inception score and the Fréchet Inception Distance, which are commonly used to assess the synthetic image quality coming from GANs [22, 39]. Statistical comparisons based on such metrics provide interesting insight but do not guarantee that data is credible for the task it will be used for. For example, a simulated sample might be very similar to a real sample according to these metrics but lacks subtle information essential for distinguishing human activities since their environment is abstract.

### 2.3 Discussions

Many existing approaches can address some parts of our two challenges but not simultaneously. Firstly, some of them have limitations related to their simulation environment, restricting their use for the generation of daily activity data. Moreover, approaches not using 3D environments are not able to credibly simulate some sensors and activities. Secondly, some approaches have limitations related to the agent’s behavior. Many of them present interesting strategies for the control or the autonomy of their agent, but few of them allow a balance between these aspects. We yet believe this lack of flexibility involves significant drawbacks for the simulation: control is important to simulate situations for which we seek to generate datasets, whereas autonomy is important to have varied and long-term simulations without human intervention. In addition, the search for greater diversity in behavior, by allowing for example interruptions or multiple choices for the same situation, is not emphasized by existing works. Yet this point is essential to obtain credibility. Finally, some approaches have limitations related to validating their simulated data. We have found two main ways to assess the credibility of their data in the literature. *a priori* approaches rely on the use of real data when building generative models to ensure a degree of credibility. *a posteriori* approaches propose experimental protocols to compare the credibility of simulated data with real ones through the use of specific metrics. Often, these metrics directly compare simulated samples with real samples using statistical arguments. However, statistical credibility does not guarantee that simulated data is useful for the application it was generated for. For example, in the case of human activity recognition, simulated samples might be statistically close to real samples but not when samples are taken individually.

Since the existing approaches cannot address both challenges at the same time, **we propose a new approach to generate and validate credible data of daily activities, performed by an autonomous agent in a 3D smart home**. To address the first challenge, we use a 3D environment enriched with sensors to generate data about daily activities. We also use the agent model developed in our previous work [21], which allows flexibility between control and autonomy, while being able to deal with interruptions and time constraints. Activities generated by our agent are executed and sensed in a 3D virtual environment presented in Section 3. To address the second challenge, we propose a new *a posteriori* approach for evaluating the credibility of the resulting simulated data in Section 5, which is then experimented on the Orange4Home real dataset [12] in Section 6. Our assessment approach is based on the replication of Orange4Home by reproducing its environment and its experimental protocol. We propose to compare Orange4Home with our synthetic data on two tasks of human context understanding: activity recognition and activity prediction.

## 3 3D VIRTUAL ENVIRONMENT

In Section 2, we discussed the benefits of using 3D environments for the generation of credible synthetic data in smart environments. With such approaches, the behavior of sensors can be faithfully replicated with respect to real ones: for example, motion sensors are triggered by the nearby motion of people, which can be replicated by the movement of virtual agents. In this section, we present our approach to creating 3D environments, our methodology to simulate virtual sensors with realistic behaviors, and the possible interactions between the agent and its environment.





Fig. 2. Virtual homes within our 3D environment. The bottom images show the floors of the Orange4Home apartment.

### 3.1 3D environment design and automatic navigation

Our 3D Virtual Environment (VE), based on Unity, can be used both for the creation of smart environments based on real buildings, or to create completely fictitious environments. We present in Figure 2 three different virtual homes created within our 3D environment: the top 2 homes are fictitious, and the bottom home is a 2-story real apartment used to collect the Orange4Home dataset [12]. Based on their architecture plans, a Virtual reality (VR) tool given by Lacoche et al.[27] is used to easily place interactive objects such as furniture but also virtual sensors and effectors.

All virtual objects, whether they are sensors, rooms, or tangible objects such as doors or plates, have been defined and classified using a semantic model described in section 3.3. This semantics allows the agent to interact correctly with the VE since all the physical constraints can be described for each object represented by a 3D model. Movements of the agent in the VE are based on a navigation grid, called *NavMesh*<sup>2</sup>, which is part of Unity. Using a collision engine, NavMesh can automatically compute a grid of places where an agent of a given height and width can fit in the VE. This grid forms the basis of a graph used by the agent to choose what path to select to go from point A to point B.

### 3.2 Virtual sensors and effectors

Simulating credible data requires that virtual sensors and effectors have credible behaviors. Real sensor data include artifacts, rather than being the result of perfect sensing capabilities. For example, real presence sensors often do not detect seemingly motionless people, whereas a perfect simulated presence sensor would always detect movement regardless of amplitude. Replicating credible behavior thus requires the replication of artifacts and quirks that occur with real sensors. In this section, we present the general principles for simulating sensors and effectors in our VE.

There are in general two ways to replicate credible sensor behavior from real sensor behavior. First, manufacturer data can be used to know how each sensor or effector generally behaves. For example, we can obtain information on

<sup>2</sup>Unity *NavMesh*: <https://docs.unity3d.com/ScriptReference/AI.NavMesh.html>

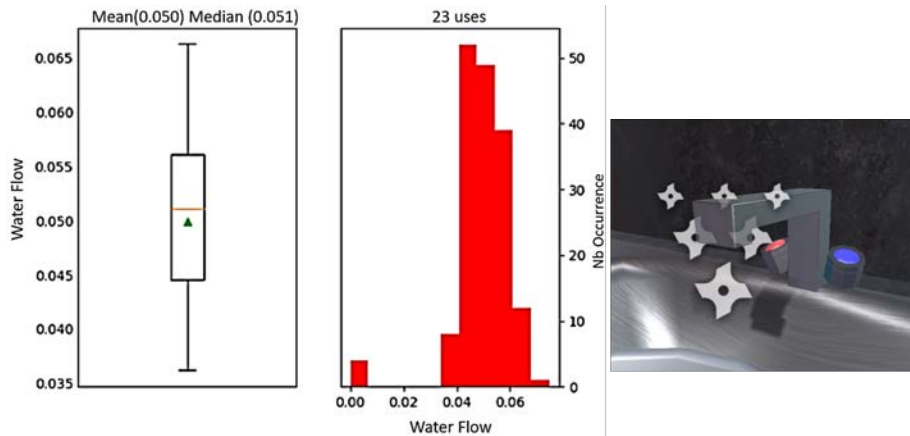


Fig. 3. Real data retrieved from the real water sensor placed on the real kitchen tap. On the left, a box plot of the real water flow values; on the middle, the number of real flow occurrences according to their value; on the right, an illustration of the simulated water sensor placed on the virtual kitchen tap.

the field of view of a presence sensor, the threshold at which a pressure sensor is triggered, or the frequency of data collection. With this data, we can replicate the general behavior of sensors. However, such documentation is not always available for specific device models or versions, or we sometimes do not know which device model or version we want to specifically replicate. We generally cannot replicate artifacts in data from these specifications.

To overcome this issue, the second way to replicate credible behavior is to observe the behavior of real sensors in collected datasets. We can then extract behaviors from statistical analysis, which can include artifacts of sensors. This approach requires that there is enough representative data for the sensor we seek to replicate. Ideally, both approaches should be combined to replicate sensor and effector behavior with maximum credibility, but this is not always feasible.

Most of the simulated sensors in our VE rely on the analysis of both manufacturer and statistical data:

- Sensors notifying when their state changes. For example, door opening sensors (opened/closed), lights (on/off), shutters (opened/closed), various appliances sending a notification when turned on or off, and switches sending a notification when pressed (on) then a notification when released (off).
- Sensors notifying when their state changes, and then a notification of their state with a regular frequency. For example, smart TV sends a notification when turned on or off, and a notification of its state (on/off) every minute as well.

Some sensor behaviors in our VE are based only on statistical analysis, performed on the Orange4Home dataset [12]:

- Water flow sensors, notifying when their state changes as well as the flow of consumed water at a specific frequency while active. Water flow sensors are used for taps (for both hot and cold water independently) and toilet flushes. Virtual sensors measure water flow based on a Gaussian distribution, parameterized based on the analysis of real water flow sensor behavior, which is illustrated in Figure 3.
- Presence sensors notifying when the agent is detected in its field of view. This latter is simulated by Unity *Colliders*<sup>3</sup>, detecting when the agent enters its detection area. We observed in real data that these sensors have a 10-second window of blindness where no new notification is sent after detecting a presence. In addition, the

<sup>3</sup>Unity Collider: <https://docs.unity3d.com/ScriptReference/Collider.html>

person is often undetected after a certain duration of immobility and then detected again. To replicate this behavior, the simulated sensor does not detect the agent if it does not move during a given period of time. After this period, if the agent moves, the sensor detects it again. Otherwise, the sensors detect the immobile agent after a random duration (corresponding to the detection of small movements that occur in real situations).

### 3.3 Interactions between the environment and the agent

In addition to the geometry of the environment, we labeled all objects and rooms used during activities. We use #FIVE, an object-relation-based model by Bouville et al. [9], and its implementation in the XR toolkit *Xareus*<sup>4</sup>. This semantic model allows for the classification of all objects of interest, according to the actions and behaviors they can have. With this model, we can group objects by interaction similarity describing how the objects and the agent can interact. We can also construct a hierarchy between them to produce more specific interactions. For example, we can group sofas and chairs of the VE in the same category, since they share the same interaction: the agent can sit on them. To make two objects interact with each other (for example an agent with a chair, a glass with a water tap, etc.), we can assign a relation between them, which can include a 3D animation representing this relation. For example, for the relation *Washing a plate*, an animation is assigned such that the agent makes circular motions with its hand on the plate.

## 4 AGENT MODEL TO SIMULATE DAILY ACTIVITIES

We previously detailed our 3D environment, our methodology to simulate virtual sensors with realistic behaviors, and the semantics used to define interactions between the agent and its environment. In this section, We now present the agent model that we developed. Since the details of its functioning are already explained in our previous works [20, 21], we focus on its global structure, its concrete implementation into the virtual smart home, and the reasons explaining why this model allows us to meet our challenges. In our previous works, we also brought a first functional validation to know if the outputs of the model are coherent with the inputs. These results are summarized in subsection 4.2.

### 4.1 Global Structure of the agent model

The global structure of our agent model is a combination of BDI architectures [41] and reactive schedulers models [42]. We choose BDI for its intuitive and flexible approach to the human decision model through the Beliefs, Desires, and Intentions models. It was also chosen for its compatibility with our requirements: BDI can theoretically allow us to combine modules managing autonomy (such as needs in desires) and modules dedicated to the control of behaviors (such as the insertion of schedulers in intentions), even though schedulers are rarely integrated into BDI implementations.

BDI implementations focus more on the reaction to current situations than on the anticipation of future constraints. They are thus adapted to manage autonomy but not really for anticipation processes. However, for our use case, where strong constraints coming from the user must be considered to follow the experimental protocol of real databases, control over the behavior is also required. This is why a scheduler is added to our model to obtain this balance. For this, we were inspired by the concept of the reactive scheduler [42] to manage both reactivity and anticipation of strong constraints. In addition, to allow the user to easily impose specific activities, our model can take as input a calendar listing all of these activities. The agent model described in Figure 4 is structured as follows:

*Agent Parameters.* : This process gathers all initial parameters and user constraints that must be considered during the simulation. The user can give an input activity calendar to provide activities that must be performed at a specific

<sup>4</sup>*Xareus Software*: <https://team.inria.fr/hybrid/xareus/>

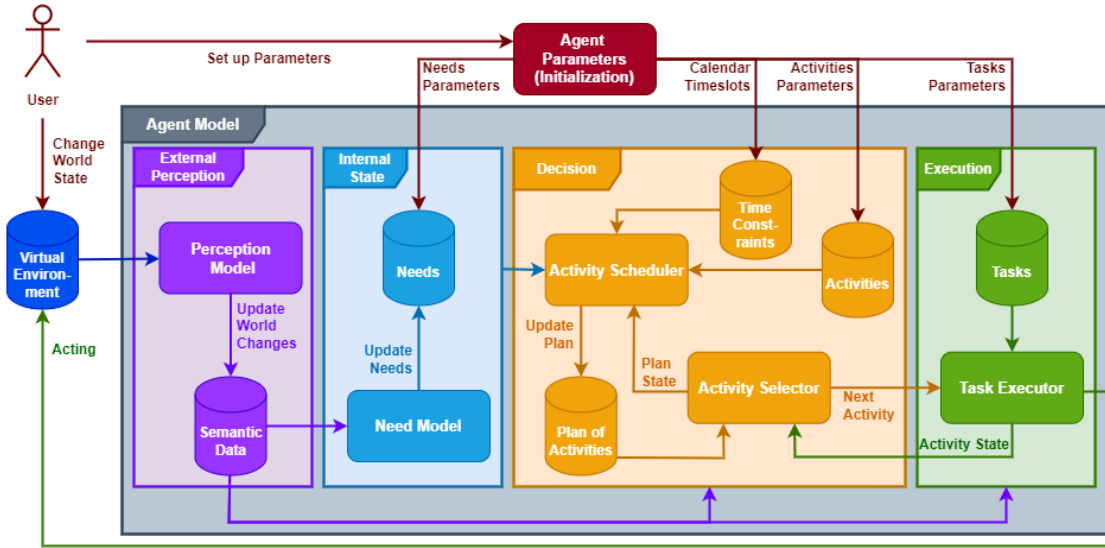


Fig. 4. Our agent model proposed in [21].

period. **This calendar gathers all the mandatory activities given by the user. These activities are mainly used in the Decision-Making Model to be integrated into the plan with the correct times and durations.** For instance, a Mandatory Activity can be a Medical Appointment on Thursday, March 3 from 3 p.m. to 4 p.m. The user is also able to configure activity, need, and task features such as duration or occurrence. By modifying the needs parameters, specific agent profiles can be created (the agent eating more often, going to bed later, etc.), thus increasing the diversity of behaviors. All these constraints are used for the control of the agent’s behavior.

*Internal State Model.* : Similar to desires in BDI models, it models the agent’s motivation to perform an activity. However, in contrast to desires, other cognitive models could be added such as preferences or emotions. In this paper, it is mainly used to update the urgency of needs. Needs are inspired by human fundamental needs defined in Maslow’s theory of needs [32]. They can be physiological, such as hunger, or they can be more sophisticated, such as self-esteem. This model is used for the autonomy of the agent since it defines goals that must be reached during free time periods. It is also involved when an interruption must occur. The needs are configured by using a temporal function, as in the work of De Sevin and Thalmann [14]. Concretely, to calculate the urgency of needs, a *priority* value is evaluated according to a threshold, used to know when a need becomes urgent, as well as the need intensity evolving through time. The user can also indicate whether a need can interrupt activities in case of urgency.

*Decision-making Model.* : This process controls decision-making, and can be related to intentions in BDI models. However, in contrast to BDI, our model does not retrieve a predefined plan but instead sets up plans during the simulation. This plan is built to respect user constraints while producing autonomous choices to satisfy needs during free-time periods. Our decision-making model contains two main processes shown in Figure 4: the *Activity Scheduler* and the *Activity Selector*. On one hand, the *Activity Scheduler* builds an activity plan for a given time window. This scheduler considers both initial constraints and the present and future urgency of needs, computed by the Internal State

Model. Our Activity Scheduler is designed to reschedule at any time, making it compatible with dynamic environments. Interruptions are also managed when a need is too urgent and when an activity can be interrupted. During the creation of the plan, activities satisfying needs are placed after assessing the approximate time when their need becomes urgent. If the agent still has free time after scheduling activities satisfying its needs, default activities are added to keep busy. These default activities are chosen randomly with optional weights affected by the agent's preferences. The approximation of the needs urgency and the random choice of the default ones introduce **diversity in the behaviors** while remaining coherent: the same input constraints do not cause the same sequence of activities in output. On the other hand, The *Activity Selector* gradually retrieves the activities of the plan. If no activity can be performed anymore, a message is sent to the scheduler to rebuild a plan until the next mandatory activity or for a predefined period. The selected activity is also transmitted to the Task Execution Model and its state is received in return.

*Task Execution model.* : This process executes the selected activity given by the *Decision-Making model* in the Virtual Environment (VE) by executing the related sequence of tasks. A task is made of basic actions and animations that can be directly executed in the VE. For instance, the *Showering* includes the task *Getting dry*. In our use case, the sequence of tasks is represented by a Petri-Net-based scripting model called #SEVEN [11] implemented in *Xareus Software*<sup>4</sup>, where a token, moving from one place to another place, triggers an interaction defined by the #FIVE [9] semantics model. The token progression can be managed by checkpoints verifying if conditions are achieved.

*External Perception model.* : This model is similar to Beliefs in BDI approaches since all relevant data from the VE are stored in its semantics database managed by the #FIVE model. It is used to filter activities according to their physical constraints and provide the objects and interactions needed to make the execution possible in the VE.

Since our model can adapt the level of autonomy according to constraints, **we can strictly replicate data collection protocols or give the agent more freedom to obtain variability and novelty in the generated data.** In addition, to improve the diversity of data, the agent can randomly select between equivalent effectors (such as two switches that turn on lights in the same room) or equivalent activities if these are not imposed (such as choosing between reading and watching TV for entertainment). This random choice can be weighted by the preferences given to the agent.

## 4.2 Agent Model Implementation and Validity

In Figure 5, we show the global structure of our implementation used in this paper and our previous work [21]. Firstly, the user can give initial parameters to configure the agents (needs values, preferences, etc.). Among them, a calendar can also be given. This calendar enables the insertion of mandatory activities to allow the user to simulate activities at specific times, or to respect a specific protocol as in our case, where we want to replicate the Orange4Home protocol. The decision-making process is then initiated and chosen activities are executed with a #SEVEN scenario storing all interactions needed to perform an activity. This scenario is linked to the 3D character to allow the execution of requested interactions in the 3D environment. The agent navigates in the environment with the Unity Navmesh<sup>5</sup>. Our animations are based on two main blocks: Inverse kinematics with FinalIK<sup>6</sup> and daily activities animations taken from *CMU Graphics Lab Motion Capture Database*<sup>7</sup> and *Mixamo*<sup>8</sup>.

<sup>5</sup>Unity Navmesh: <https://docs.unity3d.com/ScriptReference/AINavMesh.html>

<sup>6</sup>FinalIK: <https://assetstore.unity.com/packages/tools/animation/final-ik-14290>

<sup>7</sup>Mocap: <http://mocap.cs.cmu.edu/>

<sup>8</sup>Mixamo: <https://www.mixamo.com/>

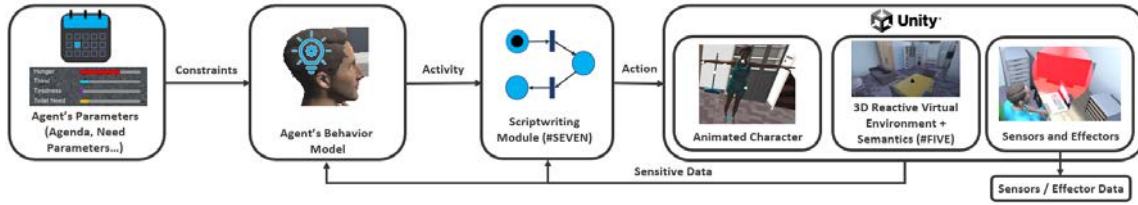


Fig. 5. Implementation of our agent proposed in [21].

A functional evaluation of our model was made in our previous work to know whether the produced behaviors are diversified and coherent in comparison to the input parameters. By comparing the input and output of our model, we showed that the agent performs all the mandatory activities coming from the user’s schedule on time, whether the schedule is strict or moderate. We then showed that calendar activities can be interrupted if a need allowed to interrupt is too urgent. During the free time period, we proved that all needs were satisfied within the expected time frame (average satisfaction in the emergency range of needs). Concerning diversity, we showed that for the same input constraints, our agent model does not produce the same output activities and the needs are not satisfied at the same time. *a posteriori* validation based on statistic methods was thus made in our previous works. However, as we said in section 2, statistical approaches are not sufficient to prove credibility. This is why we make a second validation in this paper by evaluating if the produced synthetic data can substitute real ones.

## 5 APPROACH FOR EVALUATING THE CREDIBILITY OF DATA GENERATED FROM SIMULATIONS

Section 3 and 4 showed how we can respect the first challenge through the creation of a 3D environment enriched with virtual sensors and the design of an agent model both giving control and autonomy. Now, we present our validation approach to meet the second challenge. To validate the credibility of our synthetic data despite the absence of clear specifications to finely assess credibility, we propose to replicate a real data collection experiment in a virtual home, using our agent model in place of a human subject. Since direct comparison between real and synthetic data might be too sensitive, due to particular behavior of the human or small replication inaccuracies of the VE, **we propose to evaluate the credibility of simulated data by using them for context-aware machine-learning models**. For fair comparisons between real and simulated data, it is important to replicate as closely as possible the real home, the real sensors, the real subject, and the real experimental protocol used to collect the real dataset.

In this section, we detail the real Orange4Home dataset [12] that we choose to replicate as well as the details about the simulation of the home, sensors, and experimental protocol for this specific dataset.

### 5.1 Real dataset selected for replication

Orange4Home dataset [12] was chosen for our experiments due to multiple reasons:

- Orange4Home is freely open to researchers<sup>9</sup>, which improves the reproductibility of our experiments.
- Orange4Home comprises 4 weeks of labeled daily living activities at home. We will thus judge the credibility of varied activities simulated by our approach on relatively long time scales.

<sup>9</sup>Orange4Home: <https://amiqual4home.inria.fr/en/orange4home/>

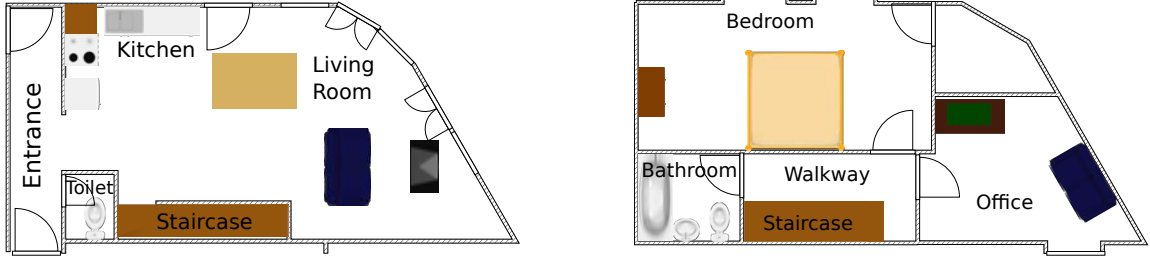


Fig. 6. Plan of the ground floor (left) and first floor (right) of the home used in Orange4Home and replicated in the virtual environment.



Fig. 7. A section of the living room, as seen in the real Orange4Home apartment (left), and in our virtual environment (right).

- Orange4Home was recorded in a two-story home comprising 8 different rooms, instrumented with 236 heterogeneous sensors and effectors. Our virtual environment will thus replicate a realistic home setting, and we have a large selection of sensors and effectors to choose for replication.
- Some of the authors of this paper have previously participated in the recording of Orange4Home. This allows for more accurate replication of the dataset and thus fairer comparisons between real and simulated data.

## 5.2 Simulation of the home

The home occupied for the recording of Orange4Home is a two-story apartment, with 8 rooms (not necessarily separated by doors or walls), connected by a staircase. This home was recreated in our VE, following the plans established for the real apartment, reported in Figure 6. This replication also includes most of the furniture, objects, and appliances that the real subject interacted with during Orange4Home’s collection.

In Figure 7, we present a side-by-side view of the living room captured from a similar point of view, in both the real apartment and our simulated replication. Photorealism is not sought-after since our data are not impacted by this.

## 5.3 Replication of sensors and effectors

Data is generated from the selection of simulated sensors we have replicated based on the original Orange4Home experiment, which contains 236 real sensors. We have replicated 63 of these sensors in the VE which seemed to be the most relevant to recognize activities: lights, door and cupboard openings, presence detection, switches, electricity instantaneous consumption, hot and cold water instantaneous consumption, shutters, and TV statuses. These virtual sensors were placed accurately to their corresponding real equivalent. Functional dependencies for certain sensors were

also properly replicated: links between switches and lights, between water consumption and faucets, door opening sensors and doors, and so on. Sensors that were not replicated fall into one of 2 categories:

- Sensors for which we do not have a virtual equivalent yet: temperatures, CO2 levels, weather information, noise levels, etc. Replicating some of these sensors accurately requires specific simulations of environmental variables, which we have not investigated in this work.
- Sensors that correspond to settings, are redundant or seldom triggered: heater settings, total accumulated consumptions, unused switches, unused appliances, etc.

#### 5.4 Replication of activity routines and experimental protocol

To match Orange4Home’s experimental protocol, we generated 4 weeks of simulated data, excluding evenings, nights, and weekends since they were excluded in the real dataset. The agenda of the real subject followed in Orange4Home was given as an input calendar in the agent. *Cleaning* activity was simulated only in the kitchen, living room, bathroom, and office. In total, 23 different activity classes were simulated (24 including moments without activity, labeled *None*)<sup>9</sup>.

Following the experimental protocol set in [13], we use the first two weeks of data for training, the third week as validation data to optimize training parameters, and the fourth week for testing (which are the results reported in Section 6), for both real and simulated data.

## 6 EXPERIMENTS

As presented in Section 5, we want to evaluate the credibility of simulated data through its use for machine learning models. We have chosen current activity recognition and future activity prediction tasks to illustrate this. The end goal of evaluating credibility is to show if we can use only simulated data to train models to recognize and predict real situations. **Activity recognition** helps assess the credibility of sensor behaviors and interactions between the agent and the VE. By comparing the gap in performance of recognition between the use of real and synthetic data, we can assess the possible inaccuracies of virtual sensors and interactions as well as the impact of missing ones. **Activity prediction** is helpful to assess the credibility of our agent model. By comparing the gap in performances of prediction between the use of real and synthetic data, we can assess whether the agent can schedule a credible activity routine. To better investigate the difference between synthetic and real data but also between the activity detection and prediction results, we use two additional metrics: A statistical comparison between simulated and real data, and Classifier Two-Sample Test (C2ST). **Statistical comparison** compares the frequencies of activation between synthetic and real sensors. **C2ST** is a metric where a machine learning model is trained to distinguish real samples from simulated ones [31].

### 6.1 Statistical evaluation of sensor notification frequencies

In this section, we analyze statistical deviations between virtual and real sensors. We draw inspiration from the metrics used in [25] to assess the behavior of sensors. The results are presented in Figure 8. The first 4 columns report the mean, standard deviation, minimum, and maximum gaps of notification frequency of real VS simulated sensors for each activity. The remaining columns only consider the sensors (real and simulated) that are activated at least once during each activity. This is interesting since for each activity, a large number of sensors is never activated. In the columns depicting the number of sensors, we examine the sensors/effectors that behaved differently by looking at the number of notifications sent in one condition (real/simulated) and not in the other. We observe significant differences between activity classes: for example, the average notification gap is low for *Cleaning* but quite high for *Computing*. Compared



| Activities         | Average gap in notification frequency | Standard Deviation | Minimum gap in number of notifications | Maximum gap in number of notifications | Number of sensors returning zero in one of the two databases | Average gap in notification frequency without zero | Standard Deviation without zero | Sensor involving minimum frequency deviation | Sensor involving maximum frequency deviation |
|--------------------|---------------------------------------|--------------------|--|--|--|--|---------------------------------|--|--|
| Going_up           | 0.047                                 | 0.141              | 1                                      | 83                                     | 10   | 0.140  | 0.218                           | office_door                                  | livingroom_presence_table                    |
| Going_down         | 0.058                                 | 0.170              | 2                                      | 108                                    | 4  | 0.306  | 0.283                           | livingroom_switch2_top_right                 | livingroom_presence_table                    |
| Leaving            | 0.171                                 | 0.372              | 2                                      | 84                                     | 4  | 0.632  | 0.476                           | livingroom_switch2_top_left                  | entrance_switch_left                         |
| Using_the_toilet   | 0.206                                 | 0.428              | 1                                      | 71                                     | 22   | 0.498  | 0.550                           | office_presence                              | bathroom_presence                            |
| Cleaning           | 0.251                                 | 0.794              | 1                                      | 310                                    | 10   | 0.395  | 0.971                           | kitchen_sink_coldwater_instantaneous         | office_tv_status                             |
| Dressing           | 0.249                                 | 0.828              | 1                                      | 275                                    | 8  | 0.825  | 1.365                           | bathroom_sink_coldwater_instantaneous        | office_tv_status                             |
| Entering           | 0.265                                 | 0.825              | 4                                      | 177                                    | 5  | 0.983  | 1.374                           | livingroom_shutter1                          | livingroom_tv_status                         |
| Preparing          | 0.347                                 | 1.144              | 2                                      | 240                                    | 12   | 1.286  | 1.947                           | livingroom_light1                            | livingroom_tv_status                         |
| Washing_the_dishes | 0.519                                 | 1.640              | 1                                      | 414                                    | 1  | 1.722  | 2.661                           | livingroom_switch1_top_right                 | livingroom_presence_table                    |
| Eating             | 0.582                                 | 2.277              | 3                                      | 652                                    | 8  | 2.037  | 3.972                           | kitchen_sink_coldwater_instantaneous         | livingroom_presence_table                    |
| Using_the_sink     | 0.610                                 | 2.345              | 1                                      | 1176                                   | 5  | 2.136  | 4.075                           | kitchen_sink_coldwater_instantaneous         | bathroom_sink_coldwater_instantaneous        |
| Watching_TV        | 1.240                                 | 4.923              | 0                                      | 1991                                   | 10   | 3.124  | 7.515                           | livingroom_light2                            | livingroom_presence_couch                    |
| Napping            | 1.011                                 | 4.525              | 4                                      | 1044                                   | 7  | 3.979  | 8.488                           | livingroom_presence_table                    | bedroom_presence                             |
| Reading            | 1.342                                 | 7.047              | 1                                      | 1650                                   | 6  | 6.038  | 14.363                          | bathroom_sink_coldwater_instantaneous        | bedroom_presence                             |
| Cooking            | 1.548                                 | 8.235              | 1                                      | 2354                                   | 2  | 5.134  | 14.634                          | kitchen_light1                               | kitchen_cooktop_power                        |
| Showering          | 1.532                                 | 7.195              | 0                                      | 2062                                   | 3  | 7.423  | 14.836                          | bathroom_switch_top_left_instantaneous       | bathroom_shower_hotwater_instantaneous       |
| Computing          | 4.354                                 | 22.965             | 2                                      | 21939                                  | 7  | 12.467   | 38.093                          | kitchen_sink_coldwater_instantaneous         | office_presence                              |

Fig. 8. Statistics about notification frequencies of real-simulated sensor pairings for each activity.

to shorter activities, longer activities have bigger differences in mean notification frequencies and standard deviations between real and simulated sensors.

We find significant differences in the number of notifications for some real-simulated sensor pairings. For example, office presence triggered only 2169 times in simulated data, compared to 24108 times in real data. Conversely, the sink water consumption during *Using the sink* was notified 1296 times in the simulation compared to 120 times in real data. Some sensor pairings have very close notification frequencies, such as kitchen presence during *Cleaning* (34 simulated, 37 real) or living room lights during *Watching TV* (36 in both). There are also cases where a sensor is triggered during simulation but not in real data, and conversely. For example, the virtual agent uses the entrance left switch during *Leaving*, but not the real occupant.

These results show that overall, most virtual sensors activate with similar frequencies compared to their real counterparts. However, certain sensor pairings have bigger deviations, especially for longer activities. The actions of the virtual agent also create discrepancies, as it sometimes interacts with switches and other instrumented devices that were not used in real situations for specific activities. This is due to the desire to diversify the synthetic data by letting the agent interact with objects that would be consistent in performing the activity. Such analysis is useful to improve the credibility of specific sensors in future works.

## 6.2 Online activity recognition

To judge the credibility of sensor behaviors and interactions between the agent and the VE, we propose to use our simulated data for the task of online activity recognition. For this task, we seek to recognize the current activity for

Table 1. Online activity recognition accuracy using simulated and real data.

| Training data    | Test data | Test accuracy |
|------------------|-----------|---------------|
| Real             | Real      | 88.70%        |
| Simulated        | Simulated | 79.40%        |
| Simulated        | Real      | 80.10%        |
| Real             | Simulated | 55.00%        |
| Real + simulated | Real      | 87.20%        |

every new incoming sensor event, based on the list of the last reported values of each sensor. The main factors in the quality of simulated data for this task are the interactions with VE and the behavior of virtual sensors. The credibility of the overall activity routines still has an impact here, but much less than both the actual quality of activity performed by the agent and the low-level data collected by the simulated sensors. These experiments are thus an indicator of the ability of interactions and sensors to generate credible activity instances.

To compare the results of activity recognition on simulated and real data, we use a simple Multi-Layer Perceptron (MLP) model. The goal is not to obtain the best performance but to have close results between the use of real and simulated data for recognition models. The input to this MLP is a vector of the last reported value of each sensor. For a fair comparison, we restrict real data to sensors that were replicated in the simulated environment (see Section 5.3). Both the real and simulated data thus share the same input feature size.

*6.2.1 Recognition performances on real and simulated data.* We report in table 1 the recognition accuracy of the MLP on simulated and real data. We see that the model is quite accurate (88.70%) at recognizing current activities from real situations, with only the selected subset of real sensors also replicated in our VE.

We see that the recognition accuracy in simulated situations is also high, although lower than in real situations (79.40% compared to 88.70%). This shows that virtual activities are globally recognized with simulated training data, even if some are harder to identify than with real instances. Several reasons can explain these difficulties: virtual sensors necessary to identify a specific activity might not be as informative as the real ones. Another possibility is that the agent interacts in simplified movements, triggering less useful sensor events compared to a real occupant.

Comparing performance in different activity classes can help identify which parts of the simulation can be improved. For example, the *Using the sink* activity in the bathroom reaches an accuracy of 37.14% on real data, compared to 3.91% on simulated data. We noticed that most of the misclassifications are with the *Showering* activity, which occurs in the same place. Therefore, we can assume that some human actions related to both activities are important to distinguish them, and were not integrated into our agent. We can also assume that the replicated bathroom and sensors might not capture all significant information related to these activities.

*6.2.2 Using simulated data for real situations.* Ideally, simulated data could reduce the need for labeled datasets collected in real situations, to train machine learning models. In this case, simulated data must be representative of real situations, so that a model trained on simulated data can provide accurate recognition when processing real situations. To illustrate this case, we train our activity recognition model with only simulated data by following the same process, and we test it on only real data (using the last real week). The resulting model, as reported in table 1, reaches a recognition accuracy of 80.10%. These performances, although inferior to the ones obtained when real data are used for training (88.70%), **show that it is possible to reach high recognition accuracy of real situations, from purely simulated training data.**

Most class misclassifications occur when the simulated model already exhibits bad performances on simulated test data (such as *Using the sink*). Conversely, classes that are well-classified on simulated test data are also well-classified on real test data. Some situations are thus not accurately simulated, limiting the recognition of their real counterparts.

For completeness sake, we also tried to recognize simulated data from a model trained on real data. We observe a large drop in performances compared to a model trained on simulated data as reported in Table 1 (55.00% vs 79.40%). Classes that were well classified when the model was trained and tested on the same type of data exhibit very low accuracies in this case. This drop is not observed in all classes: some well-classified classes with the previous model stay well-classified. The reason would be that real data contain richer information than our simulation. A model trained on real data might rely on subtle features in sensor events to categorize some classes, which are not present in simulated data and thus lead to drops in performances. Conversely, the drop was much less significant for real data with a model trained on simulated data, since our model relies on features captured by both simulated and real sensors.

Although our goal is to completely replace real data collection with synthetic generation (and not to augment real datasets), we still experimented with combining real and simulated training data to recognize real data. The model is nearly as accurate as one trained only on real data (87.20% vs 88.70%), which indicates that simulated data introduces very little information that contradicts real data for the training process. It also indicates that it does not bring new information to help recognize real activities in this example. Future work is thus needed if we seek to use our simulator for data augmentation, which is outside the scope of this work.

**6.2.3 Comparing performances on real data with all sensors.** In previous experiments, we only used the list of sensors that were replicated in the simulation (see Section 5.3) to ensure fair comparisons with the simulator. However, it can be valuable to compare these performances obtained on real data using all available sensors in Orange4Home. As expected, this model reaches a recognition accuracy of 93.20%, which is slightly more accurate than the model trained with the limited list of sensors (88.70%). The performance gap is not so large, showing that the sensors we have chosen to replicate in the simulation provide most of the required information to recognize activities accurately. This experiment also identifies certain sensors that should be simulated. For example, the model trained with all sensors is very accurate for *Napping* (99.19%), compared to the one trained with limited sensors (46.13%). This discrepancy helps us identify *a posteriori* that the bed pressure sensor, which was not replicated, is needed to correctly classify this activity.

### 6.3 Classifier Two-Sample Test (C2ST)

C2ST consists in training a model (here, XGBoost) to learn to distinguish real samples from simulated samples [31]. Our experiment shows that our simulated data is perfectly distinguishable from the real data (C2ST score of 1). Considering that each sample contains data from 63 different sensors, it is not surprising that some of the simulated sensors have behaviors that differ consistently enough from their real counterpart, such that the model can learn to use these differences to distinguish real samples from simulated samples.

In practice, we have shown in Section 6.2 that such simulated data can still be used effectively to train models for downstream tasks such as activity recognition with high accuracy. These two results combined show that simulated data does not require to be indistinguishable from real data to still be useful.

### 6.4 Future activity prediction

To judge the credibility of simulated activity routines, we propose to test the use of simulated data for the task of future activity prediction. For this, we seek to predict the next activity using previous activities and other contextual

Table 2. Activity prediction accuracy (non-Markovian depth of 3) on simulated and real data.

| Data                             | Accuracy |
|----------------------------------|----------|
| Real                             | 91.18%   |
| Simulated                        | 89.01%   |
| Simulated, without interruptions | 91.86%   |

information such as place and time. Therefore, the main factor impacting the quality of synthetic data is the agent model. The credibility of interactions and sensors is less important since we do not use them to predict future activities. These experiments are thus an indicator of the agent model’s ability to generate credible activity routines.

To compare behaviors of activity prediction on simulated and real data, we use a model proposed in [13], which was previously evaluated on Orange4Home. This model is a Dynamic Bayesian Network (DBN) in which nodes representing activity, room, hours of the day, and day of the week, are represented for both current and future situations. This model predicts the future activity that will occur. Cumin et al. [13] proposed different ways to improve activity prediction. One proposition is that sequences of human activities do not satisfy the Markov property, that is, the choice of a human to perform an activity does not only depend on their current activity but can also depend on past activities as well. In this work, we will focus on testing different non-Markovian depths for prediction, as in [13]. For example, a non-Markovian depth of 3 indicates that the model uses the current activity as well as the 2 previous activity instances for prediction.

Following the experimental protocol set in [13], we use the first 2 weeks of data for training, week 3 for validation to optimize parameters, and week 4 for test (which are the results reported here), for both real and simulated data.

*6.4.1 Predicting performances on real and simulated data.* We report in Table 2 the prediction accuracy of this model on simulated and real data. We see that real data from Orange4Home is quite predictable (91.18%), since the occupant followed a strict schedule of activities. Simulation of this schedule using our agent model without possible interruptions also leads to very predictable activities (91.86%). This first result indicates that the agent model is able to reproduce real behaviors with similar levels of predictability compared to real data.

We also obtain slightly less predictable activity sequences when our agent model enables interruptions (89.01%). Such interruptions, allowing the satisfaction of needs when there is not sufficient free time, can thus be used to introduce variability in generated data, in a controlled manner. In this way, our agent model could be used to generate different datasets of the same environment using different degrees of activity predictability, which is valuable to design and improve such predictive models. Prediction accuracies obtained in those first experiments are all significantly higher than a random predictor (5%, assuming identical predictive randomness across the 20 activities).

*6.4.2 Impact of non-Markovian depth.* We present in Table 3 the prediction accuracy obtained with varying non-Markovian depth in the activity prediction model. A depth of 1 implies that only the current activity (with other nodes of place, the hour of the day, and the day of the week) is used to predict the future activity, whereas a depth of 3 implies that the current and previous two activities are used to predict the future activity.

We see that performances are higher with a depth of 3 or 4 (89.01%), and lower with a depth of 1 or 2 (79.12% or 85.71%). This behavior was also observed on 5 real datasets as reported in [13], where multiple datasets reached their highest performance with a depth of 3, except for one dataset (the least predictable) reaching a depth of 1.

**These experiments show that the sequences of daily activities generated through our agent model exhibit similar non-Markovian behaviors to those of real datasets reported in previous works [13].** Whatever the data

Table 3. Prediction accuracy with varying non-Markovian depth on simulated data.

| Non-Markovian depth |        |        |        |         |
|---------------------|--------|--------|--------|---------|
| 1                   | 2      | 3      | 4      | Average |
| 79.12%              | 85.71% | 89.01% | 89.01% | 85.71%  |

used (real or synthetic), the past sequence of 3 activities provides the most information on which next activity will occur for this dataset. Only using 1 past activity is generally not informative enough, whereas using sequences longer than 5 activities introduces noise and unneeded complexity to our predictive model.

6.4.3 *Using simulated data for real situations.* Ideally, simulated data could reduce the need for labeled datasets collected in real situations to train machine learning models. In this case, simulated data must be representative of real situations, so that a model trained on simulated data can provide accurate predictions when processing real situations.

To illustrate this case, we trained the activity prediction model on only simulated data (following the same protocol used previously), and tested it on only real data (using the last real week). The resulting model reaches a predictive accuracy of 82.35% (with a non-Markovian depth of 3). Even though it is lower than a model trained directly on real data (91.18%), these performances are still very high and vastly superior compared to a random model, **indicating that simulated data can thus be very useful to train these kinds of models.**

For completeness sake, we also trained our activity prediction model on real data and tested it on simulated data. The resulting model reached a predictive accuracy of 82.42% (with a non-Markovian depth of 3). There are still performance gaps to a model trained on simulated data (89.01%), **but performances are again far beyond random prediction, which reaffirms that simulated data are in some way representative of real data.**

## 7 DISCUSSIONS

### 7.1 Insights on replicating datasets

We have shown in Section 6 that data generated by our simulation can be effectively used as training data for activity recognition and prediction tasks applied to real situations in Orange4Home. The results obtained in our experiments are convincing, showing that using simulated data as training sets to counterbalance the lack of real data is a promising research avenue. The C2ST results indicate that generated data can be used effectively to train models to recognize real situations, without requiring this generated data to be indistinguishable from real data. Our experiments also allowed us to identify the improvements we could make to refine the results.

Among the improvement to make, some specific activity classes are badly classified when using simulated training data, compared to real data. This indicates that credibility discrepancies exist, depending on which activity class is simulated. One source of discrepancies we have discussed in Section 5.3 is the behavior of virtual sensors. Our efforts on replicating realistic behaviors of sensors have helped limit the drop in performances, but not completely remove it. In the general case, we might not have access to the behavior of all sensors of the environment we are trying to replicate. Limiting such discrepancies would thus prove difficult. Another source of discrepancies can be the actions of the agent which can be not enough accurate compared to humans. Observing drops in performances in certain activity classes can thus help identify which parts of the simulation model are not credible enough and need to be improved. Similarly, a drop in performances is observed when we restrict real recognition models to only sensors that were replicated. This experiment allows us to know which sensors are actually essential to correctly detect some activities. In future

work, this information can be used to enrich our simulation with such missing sensors. Conversely, we could use our replication strategy to check the importance of different sensors (measured by the drops in activity recognition and prediction performances), for applications aiming at reducing the number of redundant sensors in smart environments.

## 7.2 Generalization to other contexts

Our credibility validation approach, which uses 4 different metrics (Statistics, C2ST, activity recognition, and activity prediction), has only been tested on the Orange4Home dataset. Since this study is purely experimental, the generalization of our results to other contexts is thus limited. Consequently, we cannot extrapolate our results to other datasets or generalize our observations to any smart environment. It is required to test this simulation and validation approaches in various other contexts that offer accurate real data for comparison. This work is an additional validation showing that our agent model can generate credible data from a specific dataset. A first statistical validation of the agent model can be found in [21]. However, testing it against many more varied datasets is required to ensure that generated data will be robust, reliable, diverse, and credible enough to apply to any connected environment.

To generate synthetic data, we use real-world knowledge, such as sensor behaviors or user activity planning. This is not always available, especially in commercial solutions for the general public. Moreover, this information introduces biases in evaluating credibility. The features used to recognize activities are generated by sensors designed to replicate Orange4Home in the first place. Our assessment is thus more about the ability of our simulator to produce data that behaves like real data, rather than an objective credibility metric. Designing meaningful yet non-biased credibility evaluations for simulated data is difficult to achieve. Further work should focus on evaluating the impact of such biases.

In this work, we use activity models as indirect measures of the credibility of simulated data. Our focus on human activity has multiple justifications. First, we wanted to evaluate the credibility of our VE and the captured interactions with the agent. Activity recognition is suitable since it encompasses those elements. Second, we wanted to evaluate the credibility of routines generated by our agent model. Future activity prediction is suitable since it allows comparisons between the predictability of real and simulated routines. In general, credibility evaluation could use a large array of different human context analysis tasks. For example, room occupation detection or occupant positioning can be used as another way to evaluate the credibility of simulated data. Applicable tasks are limited by data simulated by the agent model and VE. For example, if future agent models include the simulation of human emotions, then an emotion recognition task could be used for credibility evaluation. By adding virtual cameras in the simulation, we could retrieve videos that are useful to train vision-based algorithms [35]. We believe that the use of human context understanding models to evaluate simulated data is a promising avenue of research.

## 8 CONCLUSION

In this paper, we propose a methodology to simulate daily activity data and estimate their credibility. We describe a configurable simulation environment to produce credible and diverse data through a 3D smart home enriched with sensors where an autonomous agent can perform activities. We also propose a new approach to validate the credibility of simulated data: we replicated a real smart home environment, used to collect the Orange4Home dataset, with our virtual environment. We implemented virtual versions of sensors, taking into account the actual behaviors of their real counterparts. We use an autonomous agent to perform daily activities in this environment, which allows for both controllability and autonomy. To generate comparable data to the real dataset, we replicate the experimental protocol and constraints of the real collection phase. To evaluate the credibility of simulated data, we compare performances of two machine learning models, used to recognize and predict activities, on real and simulated data. We also use two other

metrics: an analysis of activation frequencies between real and simulated data, and C2ST. Through our experiments on Orange4Home, we showcase the usability of simulated environments for such tasks and their limitation.

Replicating environments and contexts used in real datasets to evaluate the credibility of simulated data is a promising avenue of research. In future work, we will test our simulation and agent model in other environments and with other datasets. We also retroactively improve our simulation aspects (sensors, agent, etc.) with credibility experiments. Our agent model will be improved to deal with multi-agent situations to generate data of several occupants simultaneously present in environments. This will also be an opportunity to showcase simulations of dynamic environments.

## ACKNOWLEDGMENTS

We thank David Cibaud for the analysis of real sensor behaviors.

## REFERENCES

- [1] Louis Airale, Dominique Vaufreydaz, and Xavier Alameda-Pineda. 2021. SocialInteractionGAN: Multi-person Interaction Sequence Generation. *CoRR* abs/2103.05916 (2021). arXiv:2103.05916
- [2] Nasser Alshammari, Talal Alshammari, Mohamed Sedky, Justin Champion, and Carolin Bauer. 2017. Openshs: Open smart home simulator. *Sensors* 17, 5 (2017), 1003.
- [3] Edouard Amouroux, Thomas Huraux, François Sempé, Nicolas Sabouret, and Yvon Haradji. 2013. Simulating human activities to investigate household energy consumption. In *5th International Conference on Agents and Artificial Intelligence (ICAART 2013)*.
- [4] Édouard Amouroux, Thomas Huraux, François Sempé, Nicolas Sabouret, and Yvon Haradji. 2014. SMACH: agent-based simulation investigation on human activities and household electrical consumption. In *Agents and Artificial Intelligence: 5th International Conference, ICAART 2013, Barcelona, Spain, February 15-18, 2013. Revised Selected Papers 5*. Springer, 194–210.
- [5] Nikos Avradinis, Themis Panayiotopoulos, and George Anastassakis. 2013. Behavior believability in virtual worlds: agents acting when they need to. *SpringerPlus* (Dec. 2013).
- [6] Behnam Azvine, David Djian, Kwok Ching Tsui, and Wayne Wobcke. 2000. The intelligent assistant: An overview. *Intelligent Systems and Soft Computing* (2000), 215–238.
- [7] Anton Bogdanovych and Tomas Trescak. 2017. To Plan or Not to Plan: Lessons Learned from Building Large Scale Social Simulations. In *Intelligent Virtual Agents*, Jonas Beskow, Christopher Peters, Ginevra Castellano, Carol O’Sullivan, Iolanda Leite, and Stefan Kopp (Eds.). Vol. 10498. Springer International Publishing, Cham, 53–62. Series Title: Lecture Notes in Computer Science.
- [8] Bruno Bouchard, Sébastien Gaboury, Kévin Bouchard, and Yannick Francillette. 2018. Modeling human activities using behaviour trees in smart homes. In *Proceedings of the 11th Pervasive Technologies Related to Assistive Environments Conference*. 67–74.
- [9] Rozenn Bouville, Valerie Gouranton, Thomas Boggini, Florian Nouviale, and Bruno Arnaldi. 2015. #FIVE : High-level components for developing collaborative and interactive virtual environments. In *2015 IEEE 8th Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS)*. IEEE, Arles, France, 33–40.
- [10] David Charypar and Kai Nagel. 2005. Generating complete all-day activity plans with genetic algorithms. *Transportation* 32, 4 (2005), 369–397.
- [11] Guillaume Claude, Valérie Gouranton, Rozenn Bouville Berthelot, and Bruno Arnaldi. 2014. Short Paper: #SEVEN, a Sensor Effector Based Scenarios Model for Driving Collaborative Virtual Environment. (Dec. 2014), 5.
- [12] Julien Cumin, Grégoire Lefebvre, Fano Ramparany, and James L. Crowley. 2017. A Dataset of Routine Daily Activities in an Instrumented Home. In *Ubiquitous Computing and Ambient Intelligence - 11th International Conference, UCAmI 2017, Philadelphia, PA, USA, November 7-10, 2017, Proceedings (Lecture Notes in Computer Science, Vol. 10586)*, Sergio F. Ochoa, Pritpal Singh, and José Bravo (Eds.). Springer, 413–425.
- [13] Julien Cumin, Grégoire Lefebvre, Fano Ramparany, and James L. Crowley. 2021. PSINES: Activity and Availability Prediction for Adaptive Ambient Intelligence. *ACM Trans. Auton. Adapt. Syst.* 15, 1 (2021), 1:1–1:12.
- [14] Etienne de Sevin and Daniel Thalmann. 2005. A motivational model of action selection for virtual humans. In *International 2005 Computer Graphics*. IEEE, Stony Brook, NY, USA, 213–220.
- [15] Lavindra de Silva. 2018. HTN acting: A formalism and an algorithm. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. 363–371.
- [16] Richard E. Fikes and Nils J. Nilsson. 1971. Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* 2, 3-4 (Dec. 1971), 189–208.
- [17] Ilche Georgievski and Marco Aiello. 2014. An overview of hierarchical task network planning. *arXiv preprint arXiv:1403.7426* (2014).
- [18] Shadan Golestan, Ioanis Nikolaidis, and Eleni Stroulia. 2020. Towards a Simulation Framework for Smart Indoor Spaces. *Sensors* 20, 24 (2020), 7137.
- [19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

- [20] Lysa Gramoli, Jérémy Lacoche, Anthony Foulonneau, Valérie Gouranton, and Bruno Araldi. 2021. Needs Model for an Autonomous Agent during Long-term Simulations. In *IEEE International Conference on Artificial Intelligence and Virtual Reality, AIVR 2021, Taichung, Taiwan, November 15-17, 2021*. IEEE, 134–138.
- [21] Lysa Gramoli, Jérémy Lacoche, Anthony Foulonneau, Valérie Gouranton, and Bruno Araldi. 2022. Control Your Virtual Agent in its Daily-activities for Long Periods. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*. Springer, 203–216.
- [22] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (Long Beach, California, USA) (NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 6629–6640.
- [23] Hyeonjun Jang, Shujia Hao, Phuong Minh Chu, Pradip Kumar Sharma, Yunsick Sung, and Kyungeun Cho. 2020. Deep Q-network-based multi-criteria decision-making framework for virtual simulation environment. *Neural Computing and Applications* (April 2020), 15.
- [24] Carl-Johan Jorgensen and Fabrice Lamarche. 2014. *Space and Time Constrained Task Scheduling for Crowd Simulation*. Research Report PI 2013.
- [25] Oihane Kamara-Esteban, Gorka Azkune, Ander Pijoan, Cruz E Borges, Ainhoa Alonso-Vicario, and Diego López-de Ipiña. 2017. MASSHA: an agent-based approach for human activity simulation in intelligent environments. *Pervasive and Mobile Computing* 40 (2017), 279–300.
- [26] Laura Klein, Jun-young Kwak, Geoffrey Kavulya, Farrokh Jazizadeh, Burcin Becerik-Gerber, Pradeep Varakantham, and Milind Tambe. 2012. Coordinating occupant behavior for building energy and comfort management using multi-agent systems. *Automation in construction* 22 (2012), 525–536.
- [27] Jérémy Lacoche, Morgan Le Chénéchal, Eric Villain, and Anthony Foulonneau. 2019. Model and tools for integrating iot into mixed reality environments: Towards a virtual-real seamless continuum. In *ICAT-EGVE 2019-International Conference on Artificial Reality and Telexistence and Eurographics Symposium on Virtual Environments*.
- [28] John E Laird. 2022. An Analysis and Comparison of ACT-R and Soar. *arXiv preprint arXiv:2201.09305* (2022).
- [29] David Lee and Mihalis Yannakakis. 1996. Principles and methods of testing finite state machines-a survey. *Proc. IEEE* 84, 8 (1996), 1090–1123.
- [30] Wonsik Lee, Seoungjae Cho, Phuong Chu, Hoang Vu, Sumi Helal, Wei Song, Young-Sik Jeong, and Kyungeun Cho. 2016. Automatic agent generation for IoT-based smart house simulator. *Neurocomputing* 209 (Oct. 2016), 14–24.
- [31] David Lopez-Paz and Maxime Oquab. 2017. Revisiting Classifier Two-Sample Tests. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. <https://openreview.net/forum?id=SjkXfE5xx>
- [32] Abraham Harold Maslow. 1943. A theory of human motivation. *Psychological review* 50, 4 (1943), 370.
- [33] Konrad Meister, Martin Frick, and Kay W Axhausen. 2005. Generating daily activity schedules for households using Genetic Algorithms. In *5th Swiss Transport Research Conference (STRC 2005)*. STRC.
- [34] James L Peterson. 1977. Petri nets. *ACM Computing Surveys (CSUR)* 9, 3 (1977), 223–252.
- [35] Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. 2018. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8494–8502.
- [36] Jennifer Renoux and Franziska Klugl. 2018. Simulating daily activities in a smart home for data generation. In *2018 Winter Simulation Conference (WSC)*. IEEE, 798–809.
- [37] Quentin Reynaud, Yvon Haradji, François Sempé, and Nicolas Sabouret. 2017. Using Time Use Surveys in Multi Agent based Simulations of Human Activity.. In *ICAART (1)*. 67–77.
- [38] Matthew J Roorda, Eric J Miller, and Khandker MN Habib. 2008. Validation of TASHA: A 24-h activity scheduling microsimulation model. *Transportation Research Part A: Policy and Practice* 42, 2 (2008), 360–375.
- [39] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved Techniques for Training GANs. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (Eds.). 2226–2234.
- [40] Mathieu Schumann, Quentin Reynaud, Nicolas Sabouret, François Sempé, Yvon Haradji, Benoit Charrier, Jérémy Albuys, and Christian Inard. 2021. Multi-Agent based simulation of human activity for building and urban scale assessment of residential load curves and energy use. (2021).
- [41] Lavindra de Silva, Felipe Meneguzzi, and Brian Logan. 2020. BDI Agent Architectures: A Survey. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*. Yokohama, Japan, 4914–4921.
- [42] Stephen F Smith. 1995. Reactive scheduling systems. In *Intelligent scheduling systems*. Springer, 155–192.
- [43] Iago Trentin, Olivier Boissier, and Fano Ramparany. 2019. Insights about user-centric contextual online adaptation of coordinated multi-agent systems in smart homes. In *Rencontres des Jeunes Chercheurs en Intelligence Artificielle 2019*. 35–42.
- [44] Tomas Trescak and Anton Bogdanovych. 2018. Simulating complex social behaviours of virtual agents through case-based planning. *Computers & Graphics* 77 (2018), 122–139.
- [45] Mengwei Xu, Kim Bauters, Kevin McAreavey, and Weiru Liu. 2018. A formal approach to embedding first-principles planning in BDI agent systems. In *International Conference on Scalable Uncertainty Management*. Springer, 333–347.
- [46] Yiji Zhao, Farnoosh Fatemi Pour, Shadan Golestan, and Eleni Stroulia. 2019. BIM Sim/3D: multi-agent human activity simulation in indoor spaces. In *2019 IEEE/ACM 5th International Workshop on Software Engineering for Smart Cyber-Physical Systems (SESCPS)*. IEEE, 18–24.