BIM_{3D}^{Sim}: Multi-Agent Human Activity Simulation in Indoor Spaces

YiJi Zhao, Farnoosh Fatemi Pour, Shadan Golestan, Eleni Stroulia
Department of Computing Science
University of Alberta
Edmonton, Canada
{yiji1, fatemipour, golestan, stroulia}@ualberta.ca

***Abstract*—Smart buildings are a prevalent example of cyber-physical systems: embedded with sensors, they emit a continuous data stream based on which algorithms are being developed to infer the occupants' activities in order to control the building's ambience to improve the occupants' comfort and safety, and to reduce the building's energy consumption. This type of *sensor-fusion-for-occupant-activity-analysis* research requires large data sets; however, the security and privacy concerns around sharing data about people's activities impedes the collection, curation, and sharing of such data sets. One solution to this issue would be the creation of a human-activity simulator for generating synthetic, yet realistic, data sets. In this paper, we describe our human-activity simulator as a component in our general framework for evaluating activity-recognition methods for indoor spaces. Our simulator, developed in Unity3D, uses the Building Information Model (BIM) of the space as the context in which to simulate multiple agents, with different abilities and tasks. We conclude with a reflection of the pros and cons of our simulator design and implementation and discuss areas for future research.**

***Index Terms*—Smart Buildings, Activities of Daily Living, Activity Simulation, Building Information Modeling, BIM, IFC, agent-based task planning**

I. Introduction

Sensors embedded in our homes and buildings promise to afford us the opportunity to make informed decisions on (a) how to optimize the safety and comfort of the buildings occupants and (b) how to reduce the energy consumed by these buildings and their impact on the environment. Research in the area of IoT-enabled indoor-activity recognition has treated these two concerns independently: under the Smart Homes heading, research seeks to improve the ambient indoor environment or to support frail individuals in their Activities of Daily Living (ADL); on the other hand, under the Smart Buildings heading, research has sought to reduce energy consumption and improve thermal comfort.

In spite of their different motivating problems, both research streams rely on activity recognition, i.e., the process of extracting information about the activities of the buildings occupants. Admittedly, the former research stream typically focuses on the activities of individuals, at high spatio-temporal granularity, i.e., recognizing the activities of daily living of a person in their home, where the latter stream examines aggregate activity indicators at coarser spatio-temporal granularity, i.e., estimating area-occupancy counts over long periods of time. In spite of this difference, a key prerequisite for the development and validation of any solution to the general problem is the availability of large datasets. Due to the substantial effort and resources required for collecting and curating such a dataset, there are few, and not large, such datasets. Furthermore, privacy concerns make them difficult to share publicly and reuse. Therefore, using virtual environments and sensors and simulating occupants' activities is a more cost effective solution, as they would allow for a vast range of synthetic yet realistic datasets to be generated.

This paper provides an overview of the existing research on simulating human activities within a virtual environment and describes our own simulator, BIM_{3D}^{Sim}, that integrates the Building Information Modeling (BIM) standard for modeling buildings with the widely adopted Unity 3D environment in which to flexibly simulate multiple agents, with different abilities and tasks. Building Information Models (BIMs) are the de-facto standard for specifying building infrastructures, and, as Tsigkanos et al. [1] points out, they can be extended for the specification and analysis of cyber-physical spaces. We have chosen Unity 3D as the implementation platform because of its support for rendering BIMs and for avatar path planning.

The structure of the paper is as follows. Section II provides a literature review, identifying the most interesting simulators today, outlining their software architecture, usage, implementation, and contributions. Section III provides a general overview of our own BIM_{3D}^{Sim} simulator and a description of each component more in detail. Section IV is a discussion of our current work, and finally, Section V concludes with a summary of our work to date and outlines future research opportunities.

II. Related Work

There are two broad approaches for simulating activities in indoor spaces [2]. **Interactive** simulators enable users to interact with the simulation software, in order to control at a fine grain the simulated activities and resulting data sets generated. Users of an interactive virtual environment have more intuitive and interactive experience than a model-based simulator. The approach allows users a large degree of control over the activities performed. However, generating more data

978-1-7281-2282-3/19/$31.00 ©2019 IEEE
DOI 10.1109/SEsCPS.2019.00011

TABLE I
MODEL-BASED HUMAN ACTIVITY SIMULATORS

Study	*Year*	*Multi-agent support*	*intero-perability*[a]
Multilevel Simulation of Daily Activities [3]	2013	✗	✗
Occupancy analysis [4]	2013	✓	✓
Heritage use planning [5]	2016	✓	✓
Simulator for (un)planned activities [6]	2016	✓	✓
Simulator for human spatial behavior [7]	2016	✓	✓
A simulator based on Event Modeling Language (EML) [8]	2017	✓	✗
MASSHA [9]	2017	✓	✗
MOVICLOUD [10]	2018	✓	✗

[a]Defines the feature that a simulator could use or exchange information. In our analysis this feature indicates whether the simulator is based on BIM.

sets requires a person to manually interact with the software, which is time–consuming and burdensome. This is why in our work we have chosen to adopt the model-based simulation methodology. **Model–based** approaches specify activity models to define an order of events, the probability of an event occurring, and the time required for each event/activity to complete. Model-based approaches generate extensive data sets describing activities over long periods of time. However, the quality and accuracy of the resulting data depend heavily on the quality of the activity description model.

Table I reports several recent model–based simulators, which provide the background context for our work.

Kormányos et al. [3] proposed a daily activity simulation for a single agent in a home environment. Activities and exact spatial position of the agent should be defined. The designer is able to easily modify them through the simulator. The sensors and physiological information of the simulated person are also simulated. Their goal is to simulate daily activities, with priorities, at multiple levels of abstraction, with higher levels simulating complex activities and lower levels simulating sensor signals. Their simulator is based on a home editor, where the layout of a home can be described. It is important to note here that, in addition to the elements typically found in an architectural blueprint, the home layout also indicates places that can be used to perform specific activities.

Two studies [4], [5] proposed the use of BIMs for defining the structure of the buildings in which activities are simulated. They used different simulation environments –the former used Cell–DEVS [11] and the latter Unity3D – and they had different motivations: the former as designed to identify bottlenecks of buildings' design that could be problematic in emergency evacuations, while the latter focused on finding a balance between efficacy requirements of heritage spaces and the preservation needs of the artefacts. The integration of BIM enhances the interoperability of the proposed methods, since this standard specification of physical environments can be shared across different pieces of software.

Schaumann et al. [6], [7] proposed an event-driven human-behaviour simulation method, where each event specifies the relevant actors, their activities, and the spaces where the activities take place. Event-driven simulation is a useful method when the events of interest are known and can be scheduled ahead of time; Schaumann et al. [6], however, proposed a method to intertwine scheduled events with another set of events that cannot be scheduled in advance. In their subsequent work [7], they investigated the use of the simulation in order to determine how well a building supports the activities of its inhabitants. In their work, events coordinate agents' behaviour and are self-contained routines with pre-conditions, procedures, and post-conditions. The events can be used to create human behaviour narratives since they can be nested in tree-like structures to create more complex behaviour. Planned events are encoded as a top-down time-based schedule, while unplanned events are a list of possible behaviours that may be performed under certain pre-conditions. All the events are overseen by a narrative manager, which resolves conflicts between events using a rule database. The authors further extended their simulation with the Event Modeling Language (EML) that helps model events in a hierarchical fashion [8]. Specifically, an actors decision making can be proactive or reactive: they can plan proactively based on the state of the world and their goal, but they can also be reactive based on the actions available to them at a certain time. In terms of video-game AI, this would involve decision trees, finite state machines, and behaviour trees. EML enhances the modularity and scalability of the system and allows for more complex events to be modelled.

Recently, Barriuso et al. [10] presented a 3D simulator, MOVICLOUD, in order to investigate the accessibility problems of a built environment. MOVICLOUD is a multi–agent system that can perform social simulations in a 3D model of a physical space, implemented in Unity 3D. Agents have their characteristics, roles, services, and additional features stored in a database. In order to assess the accessibility of indoor places, they provide information regarding completed and failed tasks. The user can then change configurations of the simulation in order to reduce the failed tasks, which results in a more accessible indoor space.

Kamara-Esteban et al. [9] presented an agent-based simulator for emulating human activities in intelligent environments and compared the sensor events emitted by their simulator against real human activity data sets. In order to generate realistic data sets, MASSHA relies on expert definitions of human behaviour. MASSHAs human simulation model is defined by high level concepts of Person, Intelligent Environment, Behaviour, Activity, Action, Object, and Sensor. The Person lives in the Intelligent Environment, interacting with Objects embedded with Sensors with specific Behaviours. Actions are executed by a Person and can be detected by Sensors. The Persons behaviour model is driven by needs, preferences, and activities (i.e., action sequences); at run time the simulated person is aware of three elements: Activities of Daily Living, TODO List, and Done List. The MASSHA multi-agent system implementation remaps their high-level concepts onto a multi-agent system. Each agent is loaded into the environment with their own behaviour algorithm, but they can interact with other agents and the environment. The authors point out that there are still improvements to be made for activity management, in order to improve the realism of the produced data, possibly using probabilistic finite state machines, neural networks, or hidden Markov models.

Human activity simulation promises to broaden researchers insights on sensor deployments that can effectively detect occupants' activities in indoor spaces, with minimized effort and cost. We have seen that a number of studies used standard specifications of buildings (namely BIM), which can potentially improve the interoperability between tools. However, there has not been an integration of BIM and IFC files with a simulation engine capable of modeling complex scenarios. The most common method is to generate the physical geometry from an IFC file using Autodesk Revit, and then apply environmental information separately before passing it into the simulation engine. Our work aims at delivering a platform that is composed of different components. Our BIM Editor integrates BIM and Activities of Daily Living (ADL) specifications, that enable users to enhance BIM with the object affordances properties, relevant to activities. These properties then can be used by virtual agents to accomplish their daily activities, e.g., sitting on a "sittable" object like a chair or a couch. Our ADL simulator, BIM_{3D}^{Sim}, is a multi-agent 3D simulator that takes as input a BIM file and a set of tasks for each agent. A hierarchy of activities for every agent based on the given tasks gets generated. As agents perform their activities in the indoor environment, our Sensor Event Generation Process generates sensor events, which can be examined and analyzed by an observation-analysis component, which can then infer the agents' activities in a manner that can be compared against the original simulated activities.

III. Methodology

The overview architecture of our methodology is depicted in Fig. 1. It consists of four main components: (i) the *human-activity simulator*, (ii) the *sensor simulator*, (iii) the *sensor-event analysis and activity-recognition* component, and (iv) the *configuration-deployment evaluation* [12] component. BIM_{3D}^{Sim}, the first of the above components and the focus of this paper, is responsible for simulating the activities of a set of agents in an indoor environment and generating synthetic activity traces. These traces are consumed by the *sensor simulator*, which generates sensor-data streams, which are further consumed by the *sensor-event analysis and activity recognition* component. The fourth component of the framework is an evaluation module that compares the synthetic simulator traces against the traces inferred by these algorithms to compute a number of indicators about the quality of the simulated sensor configuration. Together, this suite of tools supports researchers to systematically explore the properties of different sensor-deployment configurations, in a variety of environments, without actually having to deploy real sensors in real-world environments.

A. *Human-activity simulation with* BIM_{3D}^{Sim}

This component involves four main features. First, it enables the simulation designer to review the BIM elements and specify the interactions they afford. Second, based on a user-configurable model of the simulated agents' daily objectives and the desired time period, it visually simulates and produces a trace of all agents' activities as they pursue their tasks. Finally, this activity trace is consumed by a sensor event

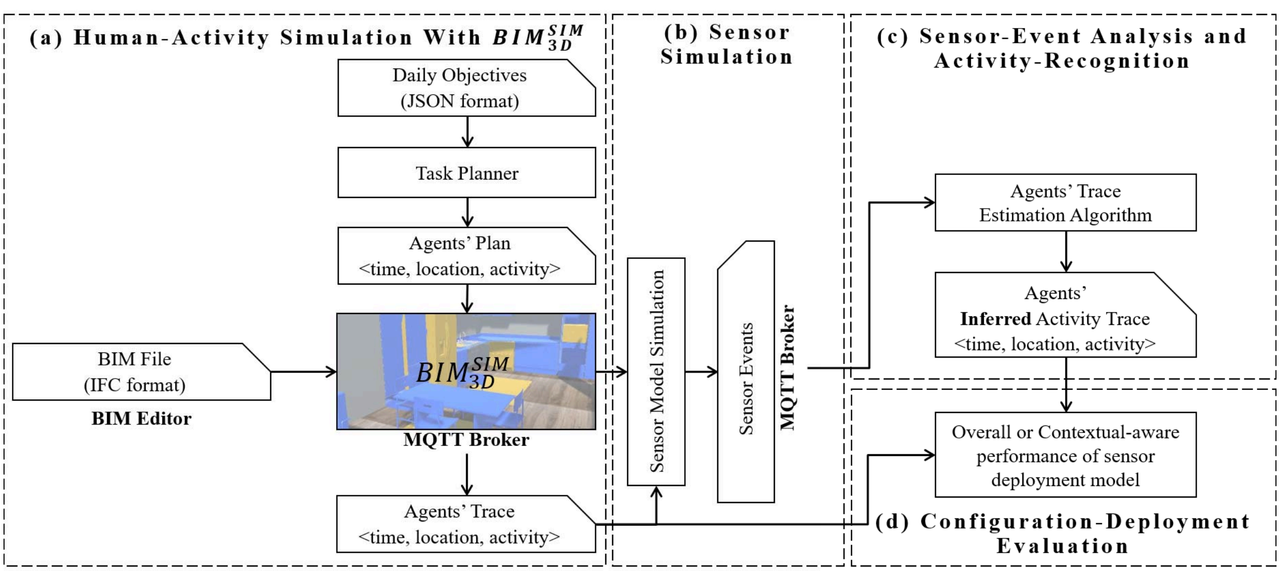

Fig. 1. Overall architecture of our methodology.

generation process, to emit a sequence of sensor events based on the environment's BIM. In the following, each feature is explained in detail.

1) **The BIM Editor:** In order to run a simulation, a BIM file of the intended indoor environment (in IFC format) is needed. We have implemented in Unity 3D a BIM Editor (Fig. 2) that visualizes the input IFC file and allows designers to modify and add a layer of information with the types of interactions that the model elements afford. For instance, if an object is defined as "sittable", agents may choose this element to sit on, when their task becomes to sit. The edited BIM can then be exported as a new enriched IFC file. This is the type of BIM that BIM_{3D}^{Sim} expects, namely a BIM with objects whose properties list the actions that the simulated agents can perform with them.

2) **The Task Planner:** This module takes as input a user-configurable set of *daily objectives* for each agent and outputs a *plan*, i.e., a sequence of time-stamped activities, for each agent. The agent's daily objectives represent the behavioural model of each agent, in terms of the activities they are capable of (and should be) performing during the day, as well as the frequency, duration, pre- and post-conditions of these activities. The agent's daily-objectives specification is non-deterministic. The agent's plan is a deterministic time-stamped and coordinate-aware sequence of activities that meets the constraints implied by the daily objectives.

The daily-objectives specification is stored in a JSON file; a sample excerpt describing sleeping and eating behavior is shown below.

```
{
  "Actions": {
    "Sleep": {
      "name": "Sleep",
      "duration": 480,
      "probability": 100,
      "occurrence": 1,
      "requires": [],
      "post": [],
      "times": [
        [ 0, 8 ]
      ],
      "aliases": []

    },

     "Eat": {
      "name": "Eat",
      "duration": 30,
      "probability": 100,
      "occurrence": 3,
      "requires": [ "Cook" ],
      "post": [ "Wash" ],
      "times": [
        [ 8, 9 ],
        [ 12, 13 ],
        [ 18, 19 ]
      ],
      "aliases": []
    },
  }
}
```

The properties of the actions are explained below:

Duration: How long the activity will last in simulation time, indicated in minutes. A value of 480 would translate to 8 hours of simulation time.

Probability: The program takes in a list of actions and randomly sorts the list every time a new action needs to be selected. The actions are then selected sequentially. This selection method ensures the equal probability of all actions being selected, which may not be desirable for agent behaviour. To remedy this, the probability we define is the chances of performing the selected action, allowing for more variation in behaviour. Probability is indicated from a 0 - 100. If the action is exclusively a precondition, its probability is 0, as it will be forcibly acted upon when needed, and no other time.

Occurrence: The maximum number of times this action can occur per day.

Requires: Actions that must be performed before this action.

Post: Actions that must be performed after this action.

Times: Time constraints on when this action can be performed. The range of times and values are indicated using a 24-hour clock. (Ex. [13, 16])

Aliases: Alternate names for the action the agent is taking to make them seem more lifelike. Prevents the creation of redundant actions, as "Eating lunch" and "Eating dinner" would effectively be the same thing.

Note that each activity has to be associated with at least one (and possibly more) object(s) in the IFC file, which affords this activity.

Within each simulation scenario, multiple behavioral profiles may be specified and each simulated agent is associated with one of them.

As mentioned above, the task planner takes as input the daily objectives JSON file and provides a plan for each agent ($< time, location, activity >$). This plan describes what activities the agent should be doing together with the time and location for each. Our simulation is defined by the space, the objects in the space, the agents, and their activities. The agent is an entity that lives and acts in the space and has its own behavioural model. The agent's behaviour can be as complex as desired and depends on spatial restrictions such as physical barriers. The agent interacts with the objects in the space, and it is these objects that enable the agent to perform its tasks. These objects also have their own constraints regarding the agents interactions.

3) **Visual Animation:** This module visualizes the environment's IFC file in Unity3D and simulates multiple agents performing their daily tasks.

For that purpose, we use avatars from [13] to perform the activities in the agents' plans. In this way, the agents' avatars have different physical characteristics, of children, men, and women, and they realistically simulate a variety of activities, such as working out, sleeping, and cooking. Fig. 2, (d) shows a screenshot of an example avatar working out, from different cameras' views. The avatar's movements during each activity are controlled by a state machine, corresponding to the activity in question. Fig. 3 shows a sample state machine for the sleeping activity, which consists of a three-step sequence of "lying down", "sleeping", and "getting up" actions.

The agent plan provided by task planner includes a list of activities to be done by the agent with specified times and locations. The BIM_{3D}^{Sim} provides an action trace for each agent ($< time, location, activity >$), describing what it should be doing throughout the simulation time, using the agent plan. That is to say, the agent trace includes the location and activity of the agent in small time periods e.g. each 2 seconds. In order for the agents' avatars to navigate through

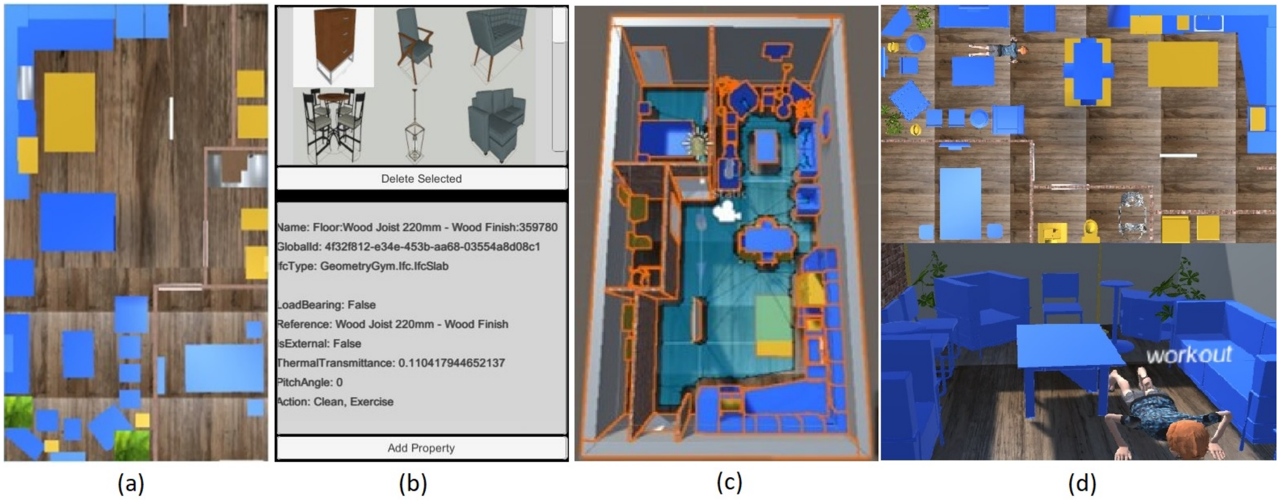

Fig. 2. Screenshots of the BIM Editor and simulation environment: (a) a rendered IFC file; (b) a panel for adding/deleting objects, or adding properties to the model objects. (c) A dynamically generated NavMesh for an IFC file, after it is loaded and rendered in a Unity3D scene. (d) A screenshot of an agent avatar working out from two different camera views.

the space, we use the built-in Unity3D path-finding methods, which require a Navigation Mesh (NavMesh) of the space. BIM_{3D}^{Sim}, when it imports and renders the IFC model of a building, it dynamically creates a NavMesh for it. Fig. 2 shows, with yellow highlighting, a dynamically generated NavMesh for an IFC file structure.

The simulation-configuration process involves the following steps. Through the simulation engine, the user selects the IFC file to be used for the simulation and the duration of the simulation. The duration cannot be too short, because the length may not be enough for the agents to visually move and perform their actions, which would compromise the quality of the output dataset. The simulation also takes as input the number of agents, the agent types, and the behaviour files for each agent.

B. Sensor Simulation

This component is responsible for consuming agents' traces ($< time,\ location,\ activity >$) as input and produce sensor events accordingly. Based on the application in hand, several sensor types, e.g. motion sensors, pressure sensors, and beacon sensors can be defined. In this paper, we focus on using motion sensors in order to localize an agent in an indoor environment, and restrict our discussion to simulating motion-sensor behaviour.

We simulate the behavior of a motion sensor based on two basic factors:

- **Random Reading:** Based on [14], sensors are subject to false readings (or random measurements). This is due to several environmental factors such as light, air particles. In order to model this behavior in motion sensors, we simply denote a probability, P_{rand}, for each sensor. In every snapshot of the simulation, there is a chance, for each sensor, equal to P_{rand} to fire, while there is no movement in it's sensing area. Therefore, in each snapshot, a random subset of sensors have false readings.
- **Detection Probability:** The closer a subject is to the center of the sensing area of a motion sensor, the more likely it is that the sensor fires. We model this behavior using Mahalanobis distance [15]. Mahalanobis distance measures the probability that a test data point, D_{test}, belongs to a set of given data points, $S = \{D_1, D_2, ..., D_n\}$. We define the agent's location as D_{test} and assume that, for each sensor, the set S follows a normal distribution, while $n \to \infty$, with known parameters. Mahalanobis distance, in this case, denotes the probability that a sensor fires if an agent is located inside of it's sensing area. The probability is highest if the agent is at the center of the sensing area.

C. Sensor-event Analysis and Activity Recognition

This component takes as input the stream of sensor events and estimates agent's location in real-time. We use the localization algorithm described in [16]. The output of this component is an inferred activity trace ($< time,\ location,\ activity >$). Since this paper does not focus on recognizing activities, we put activity to null in each tuple.

D. Configuration-Deployment Evaluation

In the final step of our methodology, this component compares the agents' ground truth activity trace from simulation with the inferred activity trace. The comparison here is based on three aspects described in [12]: Overall performance, and Contextual-aware performance. These tools aid designers to evaluate their sensor deployment model in an environment, and change their model accordingly. Overall performance calculates the average of error that localization committed in each area. Contextual-aware performance shows the average of the estimation's error throughout an indoor space while considering the frequency of usage of each area. Therefore, during a simulation, areas that agents went there more frequently, obtain more importance despite of a small error in localization (such as the dining room or bathroom). Similarly, less frequently used areas get less importance despite of high error in localization (such as the closet).

As an example, We import the IFC file of Smart Condo™ [17], a one-bedroom condo equipped with different types of sensor, and add objects with various properties. We also place 14 motion sensors, each with a circular sensing area with radius equals to 60 centimeters. We then put into the model An agent with specifications depicted in Fig. 3, and a set of actions in behavioral file. By running the simulation, we compare the inferred location of the agent, in each time, against the actual location in the simulation. Fig. 4 shows the average error, from overall and contextual-aware aspects, of our estimation in every location of the indoor space. These figures suggest areas that need attention. A designer can change the placement of the sensors in order to obtain the desired performance.

IV. Discussion

Our research has focused on two main components: 1) creating an IFC file editor (BIM Editor), and 2) visualization within Unity3D and creating a multi-agent human behaviour simulator (BIM_{3D}^{Sim}) to simulate Activities of Daily Living (ADL) within a model of an indoor space.

One of our objectives in this work is reuse and interoperability. The BIM Editor enables designers to import and render any BIM-defined indoor space, add/delete objects, and add properties. Given an indoor space, different ADL scenarios with different sets of objects can be tested. Thereby, designers are able to quickly and effortlessly evaluate several ADL scripts with different object layouts in the same space before their actual real-world experiments.

The agents' behavioural file offers flexibility in defining agents' behaviour, from simple to complicated. Based on the attributes of actions in the behavioural file and the validation of actions during task planning, incredibly complex agents can

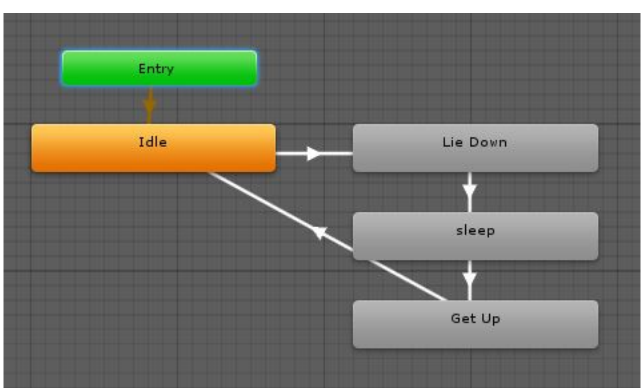

Fig. 3. A sample state machine controller for animator's sleeping activity.

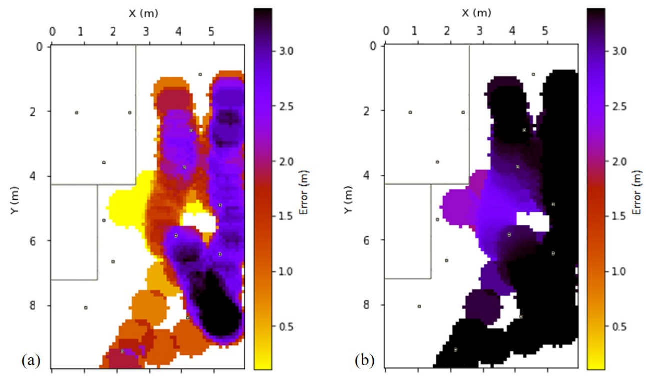

Fig. 4. Average error of the localization algorithm throughout the Smart Condo™. The floor plan of the condo is shown with thin solid lines and the center of the motion sensors are depicted with small dots. (a) shows contextual-aware performance (performance with respect to the agent's trace), and (b) shows the overall performance.

be simulated. pre-conditions and post-conditions can link a series of actions together to form a more complex behaviour and restrictions can be placed on the agent to better replicate a daily routine.

Moreover, The navigation methods in Unity3D are based on randomized algorithms which makes the behaviour of agents more natural. For instance, an agent would select different paths for walking from a point to another. This behaviour benefits our Configuration-Development Evaluation from two perspectives. First, it generalizes our evaluation of sensor deployment models since different agents' trace can be generated. Second, using a few number of scripts, we can make sure that all of the space regions are covered, and hence, tested.

The simulation could also be improved by having a more complex path-finding algorithm. BIM_{3D}^{Sim} uses the built-in Unity3D navigation system, which can be problematic when attempting to simulate agents with certain disabilities such as blindness. This is because the navigation system assumes the agents know the state of their environment, and these assumptions can make the generated dataset invalid. Furthermore, Our methodology still needs to be evaluated in order to determine its efficacy of the generated data set. Out of all of the related work, the evaluation method used for MASSHA [9] that compares the simulations generated data set to a real-world data set would be most effective.

It is important to note that our methodology is based on a loosely-coupled architecture, meaning that each component has no or little knowledge about other components. Our loosely-coupled architecture offers flexibility and extensibility features to our methodology. Each component in our architecture can be changed or replaced with another similar component, with respect to the input(s) and output(s) it accepts and produces. For instance, Sensor-event Analysis and Activity Recognition component can be equipped with Machine Learning techniques. More importantly, implementing more complex Sensor Model Simulation is easier and faster in our architecture. It is possible to utilize different covariance matrices or asymmetry distributions in order to model environmental noises or sensors' aging.

Finally, overall and contextual-aware performance plots assist designers to examine different configuration for sensor placements and choose the most suitable one based on their needs. For instance, a health–care application for older adults probably should monitor more frequently used areas (such as the bathroom) in order to detect the risk of falling. Therefore, the contextual-aware performance analysis comes into play.

V. Conclusion and Future Work

In order to study the deployment of Internet of Things in indoor environments, utilizing human-activity simulators has recently increased. The simulators generate the large data sets necessary for the development and evaluation of activity-analysis algorithms. In this paper, We presented a multi-agent human activity simulator, BIM_{3D}^{Sim}, integrated with Building Information Models (BIM), with capability of using different agents' types and behaviours.

BIM_{3D}^{Sim} was designed with interoperability in mind: the simulation scenario is configured with (a) the BIM of the indoor space, specially annotated with the affordances of its elements, and (b) the tasks that simulated agents have to carry out and the actions they are capable of performing. BIM_{3D}^{Sim} is one of the components of our sensor-deployment-analysis framework [12] responsible for generating synthetic agent activity traces. These traces are consumed by a downstream sensor simulation, which generates sensor-data streams, which are further consumed by our sensor-event analysis and activity-recognition algorithms. The synthetic BIM_{3D}^{Sim} traces are compared against the traces inferred by these algorithms to compute a number of indicators about the quality of the simulated sensor configuration.

In the future, we would like to expand the agents' task-and-actions model, and the planner module, to consider the interactions among agents as they work and collaborate in a space.

There are also many opportunities for expanding this work to other domains, including the simulation of workers' activities in factories and construction sites, possibly extending the specification of the simulation context with CityGML information.

References

[1] C. Tsigkanos, T. Kehrer, C. Ghezzi, L. Pasquale, and B. Nuseibeh, "Adding static and dynamic semantics to building information models," in *Proceedings of the 2Nd International Workshop on Software Engineering for Smart Cyber-Physical Systems*, ser. SEsCPS '16. New York, NY, USA: ACM, 2016, pp. 1–7. [Online]. Available: http://doi.acm.org/10.1145/2897035.2897042

[2] J. Synnott, C. Nugent, and P. Jeffers, "Simulation of smart home activity datasets," *Sensors*, vol. 15, no. 6, pp. 14 162–14 179, 2015.

[3] B. Kormányos and B. Pataki, "Multilevel simulation of daily activities: Why and how?" in *Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1–6.

[4] S. Wang, G. Wainer, V. S. Rajus, and R. Woodbury, "Occupancy analysis using building information modeling and cell-devs simulation," in *Proceedings of the Symposium on Theory of Modeling & Simulation-DEVS Integrative M&S Symposium*. Society for Computer Simulation International, 2013, p. 26.

[5] D. Simeone, U. M. Coraglia, S. Cursi, and A. Fioravanti, "Behavioural simulation for built heritage use planning," in *Complexity & Simplicity-Proceedings of the 34th eCAADe Conference*, vol. 2, 2016, pp. 503–510.

[6] D. Schaumann, Y. E. Kalay, S. W. Hong, and D. Simeone, "Simulating human behavior in not-yet built environments by means of event-based narratives," in *Proceedings of the Symposium on Simulation for Architecture & Urban Design*. Society for Computer Simulation International, 2015, pp. 5–12.

[7] D. Schaumann, M. G. Morad, E. Zinger, N. P. Pilosof, H. Sopher, M. Brodeschi, K. Date, and Y. E. Kalay, "A computational framework to simulate human spatial behavior in built environments," in *SimAUD 2016 Symposium on Simulation for Architecture and Urban Design*, 2016.

[8] D. Schaumann, K. Date, and Y. E. Kalay, "An event modeling language (eml) to simulate use patterns in built environments," in *Proceedings of the Symposium on Simulation for Architecture and Urban Design*. Society for Computer Simulation International, 2017, p. 21.

[9] O. Kamara-Esteban, G. Azkune, A. Pijoan, C. E. Borges, A. Alonso-Vicario, and D. López-de Ipiña, "Massha: an agent-based approach for human activity simulation in intelligent environments," *Pervasive and Mobile Computing*, vol. 40, pp. 279–300, 2017.

[10] A. L. Barriuso, F. De la Prieta, G. Villarrubia González, D. H. De La Iglesia, and Á. Lozano, "Movicloud: Agent-based 3d platform for the labor integration of disabled people," *Applied Sciences*, vol. 8, no. 3, p. 337, 2018.

[11] Cell–DEVS, "Cell based discrete-event simulation," Online http://www.sce.carleton.ca/faculty/wainer/celldevs/introduction.html, Accessed Jan. 2019.

[12] S. Golestan, A. Petcovici, I. Nikolaidis, and E. Stroulia, "Simulation-Based deployment configuration of smart indoor spaces," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT) (WF-IoT 2019)*, Limerick, Ireland, Apr. 2019.

[13] "Mixamo: Animation in seconds," [accessed 31-January-2019]. [Online]. Available: https://www.mixamo.com/

[14] S. Thrun, "Learning occupancy grid maps with forward sensor models," *Autonomous robots*, vol. 15, no. 2, pp. 111–127, 2003.

[15] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart, "The mahalanobis distance," *Chemometrics and intelligent laboratory systems*, vol. 50, no. 1, pp. 1–18, 2000.

[16] P. Mohebbi, E. Stroulia, and I. Nikolaidis, "Sensor-data fusion for multi-person indoor location estimation," *Sensors*, vol. 17, no. 10, 2017. [Online]. Available: http://www.mdpi.com/1424-8220/17/10/2377

[17] "Smartcondo™," https://www.ualberta.ca/health-sciences-education-research/simulation-experiences/hserc-spaces/smart-condo, accessed: 2019-01-31.