

Smart Home Human Activity Simulation Tool for OpenHab-based Research

Asma Alotaibi
Cardiff University
Cardiff, United Kingdom
AlotaibiA3@cardiff.ac.uk

Charith Perera
Cardiff University
Cardiff, United Kingdom
charith.perera@acm.org

ABSTRACT

Over the last few years, many different Internet of Things (IoT) devices has been made their way into the market. As a result, many different smart home solutions have been introduced by different companies. However, at this point, most of the smart home solutions are engineering products built with limited attention to the end-user needs. Many researchers are working towards developing more meaningful smart home solutions and looking at smart home domain from different perspectives such as human-computer interactions, edge computing, psychology to name a few. Another important aspect is to explore how these smart home solutions could be used to facilitate independent living, disable and vulnerable people. Further, these smart homes systems need to be privacy-aware and secure. These research activities, at some point, need some kind of human activity simulation tool. Recognising this problem, this project developed a tool capable of simulating human activities in a smart home. Our tool allows researchers to create smart home scenarios by combining different sequence of events (e.g., switch ON light, switch OFF light, Door OPEN). Once the complete scenario is defined, researchers can play it in real-time (i.e., messages are sent to the OpenHAB smart home solution where OpenHAB assumes that these messages are actually coming from IoT devices). OpenHAB then act accordingly (e.g., trigger pre-defined actions based on the messages it receive).

Demo: <https://youtu.be/YVKkXVcf4bo>

Code: <https://gitlab.com/IOTGarage/smart-home-simulation-tool>

CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous and mobile computing**.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

Technical Report, 2019, United Kingdom

© 2019 Copyright held by the owner/author(s).

ISBN 999-9-9999-99999-9/99/99.

<https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

KEYWORDS

Internt of Things, Smart Homes, Human Activity Simulation

ACM Reference Format:

Asma Alotaibi and Charith Perera. 2019. Smart Home Human Activity Simulation Tool for OpenHab-based Research. In *Technical Report*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

1 INTRODUCTION

Smart devices accumulate data that are then processed and stored for subsequent analysis [22]. The analysis of this generated data can be utilized in several areas. One area attracting particular interest is the intelligent machine learning methods that have the ability to provide services to those living in smart homes. Consequently, the behaviour of the inhabitants can be modelled and anticipated using a combination of machine learning algorithms, artificial intelligence algorithms, activity recognition algorithms, and data mining techniques. As such, these methods can be used to provide insight into how best to address issues relating to home security and which offer the potential to help detect a wide range of security threats [11, 12]. In addition, it will help anomaly detection in the behaviour of elderly people or the recognition ADLs. Furthermore, these are vital in the medical field to help diagnose medical conditions in patients [26]. However, such services are only possible if they are supported by comprehensive datasets that facilitate the testing of results so that they can be validated [7].

It is not easy to gain access to datasets of this nature, however, owing to the considerable financial costs involved and restrictions associated with the development and application of intelligent services to the home [25]. Indeed, the financial cost of developing smart homes and collecting the resulting data is simply too great for many projects [4, 25]. Whilst recognising the need to data, the occupants' privacy must be preserved and those ethical, technical, and legal matters must be given due consideration [2, 4, 11]. Therefore, these problems can be overcome by utilising data simulation methods which offer researchers access to rich datasets when testing innovative data analysis methods, especially when their development is at an early stage [2].

Contribution of this paper as follows: The main aim of this project is to develop tool is capable of simulating human activities in a smart home environment. That by publishing the simulation data through MQTT into the OpenHAB framework so at the end researchers can obtain a database. To achieve this, the focus has been to develop a tool that is able to publish q simulated a set of scenarios in either on real-time or the mimic time into OpenHAB. Where the first aims to get realistic data whilst the other helps researchers publish several data at a time, regardless of the actual time of the data. That was done by setting up the system interface and writing python code to publish the data using MQTT protocol into the OpenHAB interface. Furthermore, the tool is able to create multiple

2 BACKGROUND AND RELATED WORK

Smart home systems

"Smart home" is an expression that is utilized in reference to home that makes use of a home controller to incorporate various automation systems that home-installed [20]. Furthermore, 'Home automation' or 'Domotics' are names known for Smart home systems. Smart home systems serve various purposes including the provision of energy efficiency, leisure, even the security, and convenience for homeowners by letting them control their installed smart devices using a Home automation app that was installed on personal devices such as a smartphone or any other connected devices [21]. Smart home integration allows devices to communicate with each other using a home controller hence allowing the homeowner to manage his/her home through voice or button [20]. Besides the internet of things (IoT), smart home systems need to operate through IoT frameworks, as this allows them to share consumer usage data among themselves and perform automated actions on the basis of homeowners' preferences [21]. An IoT framework is a platform for the home automation application to react with smart devices. As shown in Figure 1, an IoT framework is a link among the user app that interacts with and control devices of IoT . The user application mostly communicates with the framework by Internet Protocol (IP), while the IoT device communicates through any protocol it supports, such as the MQTT protocol.

There has been an emergence of open-source frameworks that have received significant corporate backing. The main benefit of an open-source framework is its ability to be user expandable as well as manufacturer-neutral. The quality of these platforms depends on developers recognising the importance of considering active communities. We selected OpenHAB to be used in our work due to its open-source nature and the comprehensive set of features its provides.

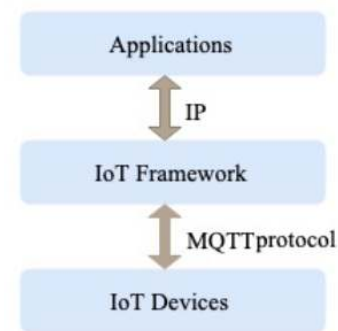


Figure 1: Block diagram for a typical IoT configuration. It consists of Applications and Devices communicating across different communications protocols via an IoT Framework

Dataset Generation

The literature shows that technology efforts concentrate to generate for smart homes more datasets for subsequent analysis. These efforts can be divided into two groups as shown in Figure 2 according to datasets that have been generated from real smart homes testbeds or utilize simulation tools on the smart home. Synnott et al. [25] placed the simulation tools for smart home into two broad categories using two main approaches: model-based and interactive approaches. A third approach uses a combination of model-based and interactive simulation; this approach is known as a hybrid [2]. Different smart home simulators are developed for different reasons based on the users' needs, [19] Some of the purposes of these simulators include dataset generated, visualization, and context awareness.

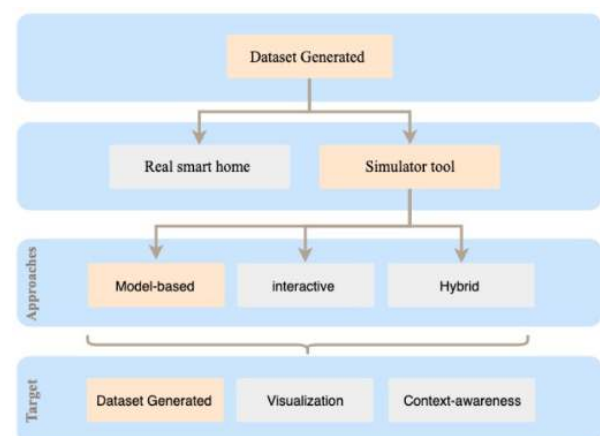


Figure 2: It shows the categories of the methods to generate the Smart home dataset. "The target" means the simulated approaches purpose of the creation of the dataset.

Real Smart Home Testbeds

The Ideal pattern of some modern datasets that have focused on ADLs is the ContextAct@A4H dataset. The generation of the dataset was done through a real apartment that has different types of sensors [9]. As a result, a new annotation method that uses temporal logic has proposed by experts.

A typical example of a project focussed on building a real smart home for the purposes of research is the work done at the Centre for Advanced Studies in Adaptive Systems (CASAS)[10]. A toolkit was created named ‘the smart home in a box’ by the organisation, that can be with ease installed in a house to provide smart service simulation. All the components for the toolkit are small in size and can be fitted into a singular box. In 32 homes These toolkits have been successfully installed for the purpose of capturing the interactions of participants; the collected data placed online and available to use [8].

Furthermore, in 2013, Alemdar et al. Export Activity Recognition with Ambient Sensing, known as the ARAS dataset, have complicated scenarios, it acts as a dataset for multiple residents [1]. Through a period of two months, the capturing for the dataset occurred over with multiple houses and inhabitants. Indeed, for assessing ADLs’ classification algorithms, ARAS was used [1, 15].

Despite the public availability of these datasets [1, 9, 10], there is a limitation on the number of activities that they can handle, and the difference of the aims in the usage of the sensors. Testing and training algorithms are an important activity that is used to secure obtaining data for a particular script or particular kinds of people. Indeed, the particular reason for this is those typical activities in a student’s day are very different from a typical day for the frail and the elderly. For that reason, current publicly available datasets are insufficient for fulfilling the required needs for some research. However, gathering new data is a very difficult and costly mission and needs a long time.

Model-Based Smart Home Simulation Approach

Model-based approaches for the simulation data for the purposes of generating synthetic devices data include specific activity models that determine the arrangement of the events, the likelihood of an event taking place, and the amount of time that the performance of specific activities takes [2, 25]. Model-based simulators, for instance, those that are characterized by Bouchard et al. and Helal et al. use the pre-defined pattern of the user’s recorded conduct. These axioms could be in realty time or otherwise. They have the capability to generate data for long periods of activities and time [5, 13]. Moreover, data collected from these simulators rely heavily on the quality of the model used.

According to the rise costs involved in constructing the actual smart environment as well as the ethical issues and privacy needed by humans, Helal et al. Created a simulation tool ‘eventdriven’ to be used by researchers in the discipline of the smart home. The product simulator is referred to as the “Persim”, it can create realistic datasets for complicated scripts of the occupants’ behaviours [13].

As an improvement on discovered disadvantages, a “Persim” upgrade was developed, which Helal et al. Referred to as ‘PerSim 3D’. This tool can be used in generating real datasets from the occupants’ activities depending on a smart home scenario. The main improvement from the older PerSim was adding 3D simulations. It also has a Graphical User Interface (GUI) support to visualize the 3D events. However, this tool is not publicly and freely available [16]. SIMACT is a 3D simulator for the smart home that has been formatted and developed for the purpose of carrying out recognition of the activities. SIMACT contains numerous preregistered scripts that researchers obtained through clinical experimentation, and it could be used to make datasets with the purpose of ADLs recognition [5]. As stated, it has an opensource as well as a cross-platform project that has been coded in Java; it uses Java Monkey Engine (JME) and has a 3D engine [25]. However, there is a gap regarding the details relating to the format of data output, was generated. Moreover, this tool relies on a 3D simulation and control script, with the control script written in a XML language. The script lacks simplicity and brevity; this means that the volume of data is the large and less readable relative to other text-based data transmission formats such as JSON [23].

Given that there is a need to serve as input for micro-simulation patterns and to produce general-use files out of kept data as a tool to develop “augmented datasets”, there is a proposal to create a synthetic dataset simPop as a statistical detection control resolution [27]. In relative terms, simPop is user-friendly and has an R package that is based on a concept that is modularobject- oriented. It supplies a highly optimized S4 of various way class implementation. These methods include standardisation by repeated proportional fitting and annealing of simulation [27]. Despite that, simPop focuses on a synthetic population dataset to generate a dataset. This kind of dataset has a particular appeal for policymakers and development experts because they can be used to assess the ex-ante distributional effect of programs and policies.

Jahromi et al. [15]Propose the development of a smart house simulator that possesses the characteristics and ability to design an architectural model or load a top view plan for a house based on the house plan. This simulator is used to simulate a house very close to the real architectural model.

Despite the effectiveness that has been demonstrated by these tools to generate the dataset, there are limitations to

their abilities. For instance, they need 3D [5, 16] and 2D designs to complete the simulation. This thesis focusses on simulated human activity based on scenario creation performed by developers or researchers in a simplified approach, i.e., through user interfaces that give the researchers the ability to create the data in the form of a JSON file and use them to create a real environment inside a smart home and ultimately push it through MQTT protocol to the OpenHAB framework.

Interactive Smart Home Simulation Approach

An interactive approach differs from a model-based approach in that it depends on possessing an avatar that a researcher able to control a human or simulated participant. [2, 25] Park et al. offered simulate virtual space that could be used for generating occupants' data for the purposes of classifying issues [18]. In order to ensure that the pattern occupants' activities are in a 3D format, Unity3D (unity.com) technology was used to build the simulator.

The Intelligent Environment Simulation (IE Sim) is another tool that is utilized for the purposes of generating simulated datasets that possess the capability to capture the normal as well as abnormal ADLs of the occupants [24]. IE Sim grants the investigators to develop smart home designing by producing 2D photographic overhead-view for the outline of the floor. The investigators are also allowed to add various kinds of sensors, like temperature, pressure sensors. Afterwards, with the aid of an avatar, the simulation could be carried out in such a way that it captures ADLs [24]. The structure for this created dataset is HML. However, in the public domain, IE Sim is not available.

Ariani et al. [3] Created a simulation tool for smart home that made use of ambient devices for capturing the occupant's interactions. The simulation tool possesses a map editor in order to let the researcher for design or create a flooring scheme similar to the intelligent home through patterning the shapes on a 2D frame; afterwards, ambient devices to the virtual home are able to add by the researcher.

Interactive approaches assume the form that a human or a researcher controls as an avatar where the simulated environment activities were performed through the avatar. These different approaches in simulation allow a realistic and coherent behaviour given that involved a real human in this process. However, in order to obtain a near-perfect simulation, the interaction between the simulated environment and the controlling human near-real-time is required [3, 18, 24]. For that reason, the rapid production of the massive dataset is extremely difficult. Thus, there is a restriction confining it, within a short time to create a dataset, and centring on particular status or activities.

Hybrid Smart Home Simulation Approach

The hybrid approach aims to combine the best among approaches, i.e., the model-based approach and interactive approach, to come up with additional complicated activity models that also means the work on human controllers and modellers will be reduced [17, 19]. Utilize of an interactive approach for the purposes of generating data from the interactions between the avatar and smart objects. On the other hand, the model-based approach is used for the purposes of generating data from passive controlling simulators, such as located of movement sensors [17]. IE Sim was developed to allow the use of a model-based approach (Poisson distribution) for increasing the interactive data that was created by IE Sim [24]. For that reason, the new version of IE Sim makes use for the purposes of simulation in a hybrid way [2].

In order to generate huge representative datasets for smart home, of expanding and extending a dataset, a replication algorithm is designed for purposes. A small pattern dataset created by OpenSSH can be extended using a hybrid approach with keeping the events logical order. In developing this design, Alshammari et al. The combination of the interactive gathering of data acquired in short duration of time, and the model-based approach to collect this duration into whole days [2].

3 OVERVIEW OF OPENHAB

OpenHAB (openhhab.org) is developed using Java and is based mainly on the Eclipse Smart Home framework. The system uses Apache Karaf together with Eclipse Equinox to develop an Open Services Gateway initiative (OSGi) runtime environment. Jetty is also relevant in such a system given that it is used as a HTTP server for the ecosystem. OpenHAB essentially is a platform that plays the role of combining various technologies with home automation systems which allow to overarching mechanism rules by providing a single solution well as offering uniform user interfaces. Moreover, it is a modular software that is scalable through "add-ons"; which equip OpenHAB with a wide variety of capabilities. These include user interfaces and the ability to interact with a larger and inevitably growing number of physical things. Add-ons can be sourced from OpenHAB 1 or OpenHAB 2 distribution and the Eclipse Smart Home project.

OpenHAB has at its core a hub-based system that can be run on any device supported by Java Virtual Machine (JVM). Thus, OpenHAB can be run on various popular platforms such as Linux, Windows, and iOS. A simpler way to experiment with OpenHAB is through the Raspberry Pi. However, although OpenHAB offers a streamlined and simple way to set up a smart home, it has a limitation in as far as completely automating the entire home is concerned.

In the same way as other open-source structures, OpenHAB is intended to coordinate existing devices together under one regular control plot. To do this, OpenHAB depends on the client created 'bindings'; these are suitable library modules that enable it to control individual IoT items or groups of items. At the time this research was conducted, there were 231 bindings accessible for use. These bindings range in unpredictability from a straightforward module that tells a client when a specific Bluetooth gadget is in range to a binding that supports all Nest gadgets by means of their API.

OpenHAB is intended to be an arrangement of frameworks. In other words, it does not give another approach to devices to convey data; it sits in the centre and performs transformations as required. Clients can choose how much ability they need to give their OpenHAB centre point. Somebody can essentially utilize an old PC to have the OpenHAB administration or a lightweight PC like a Raspberry Pi, yet on the possibility that they need it to have the option to utilize Bluetooth, ZigBee, or Z-Wave, they should buy equipment connectors to help these interchanges. Additionally, OpenHAB will expect that any underlying arrangement for a gadget is now finished. For instance, if a client needs to set up a shrewd light with OpenHAB controls, the bulb, from that moment, should be on the home system utilizing whatever strategy the bulb's maker has set up to achieve that. Basically, OpenHAB is a solution to the day by day utilization of a framework as opposed to setting up a framework. Regarding OpenHAB's internal functions, there are several central ideas. OpenHAB uses the ideas of things and items. Things are generally physical gadgets with various capacities accessible, yet likewise, they can be a virtual administration that fills in as a gathering of various capacities. Things are properties that can be controlled. In the case of a shrewd light, there might be an item for shading, power, and brilliance; these are the settings that clients control to control gadgets. There is additionally the possibility of sitemaps, which OpenHAB uses to make UIs for controlling items[22]. Sitemaps are set up by altering a design record where clients physically announce and name catches, sliders, and other GUI components. Ultimately, OpenHAB has a rule's record, which is where robotization comes in. The rules document is likewise arranged physically in a setup record.

4 OVERVIEW OF THE PROPOSED APPROACH

The target of the proposed tool is to facilitate the ability to simulate human activities in the environment of smart home and the try to fix the related issues in the previous sections. That by publishing the simulation data through MQTT into the OpenHAB framework so at the end researchers can obtain a database.

This part will present the structure of the simulation tool and the implementation of the technical details, which involves simulation Tool, OpenHAB and MQTT. In this part, the researcher will be referred to as an entity and the Scenario as inputs. The researcher is accountable for the major of the task with the tool by applying the given Scenario. Figure 3 shows the mutual effects among important components in general.

The OpenHAB framework

As discussed earlier, the landscape of Internet frameworks is already wide, so a way to display and control IoT devices of human events can be valuable if it is done through a suitable framework. In this thesis, simulating human events at a smart home is the first step towards the goal. The configuration of the MQTT broker files is the second step, and connecting a third-party framework is the third step.

OpenHAB is a platform that plays the role of incorporating diverse home automation frameworks and advancements. These incorporate User Interfaces and the capacity to associate with a bigger and definitely developing the number of physical things. In addition, it is a secluded programming that is adaptable through "Add-ons"; Add-ons provide OpenHAB with a wide variety of capabilities. Considering on Add-ons, OpenHAB has a complete list of supported service known as "persistence" which is able to store data over time plus the ability to retrieve them later. That add-on OpenHAB persists know the states of connected Items. Therefore, It is easy for the researcher to obtain a dataset for earlier the simulated scenarios. In addition, OpenHAB relies upon the customer made Bindings, which are sufficient library modules that empower its power over individual IoT things or gatherings of things [6, 14]. There are 231 Bindings available for use. Our consideration depends on MQTT Binding, which OpenHAB has prepared for MQTT, accommodating our objective. However, OpenHAB itself is not an MQTT Broker and needs to connect to one as a regular client. That why we necessity the Mosquitto to configure the communication of MQTT protocol. To conclude, the value of using the OpenHAB platform is the ability to publish and store simulation data from the simulation tool to then create a database ready for analysis by researchers.

MQTT

MQTT is a server/client architecture. This particular binding can configure connections to brokers via OpenHAB things. MQTT is one of the most generally used agreements in IoT (Internet of Things) projects. It stands for "Message Queuing Telemetry Transport". It is structured as a lightweight, making a known pact that used published/subscribed operations to exchange information between the server and the customers. MQTT servers are called brokers and the clients

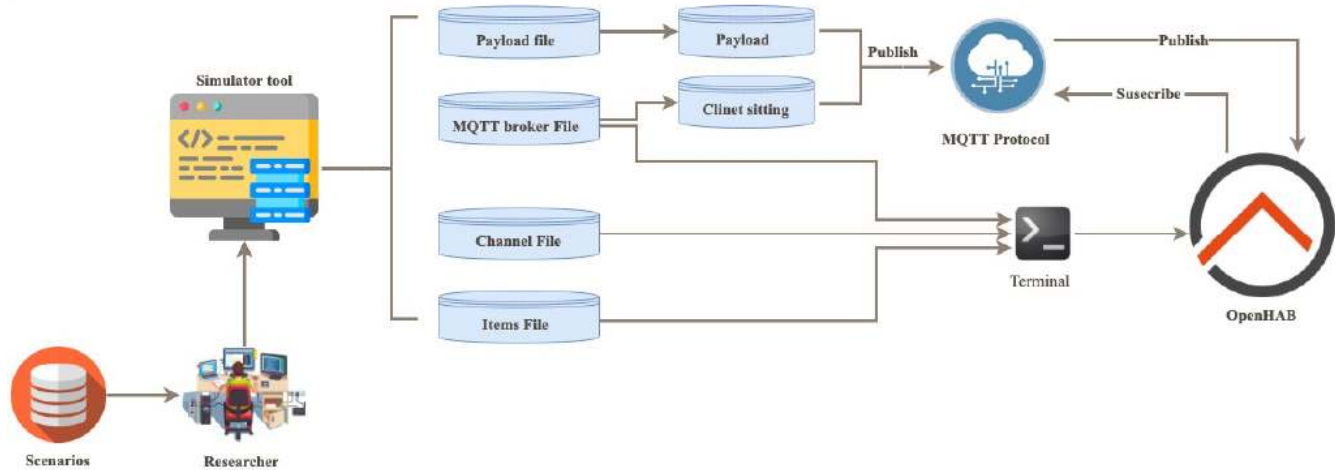


Figure 3: Overview of the Proposed Approach

are essentially the associated gadgets. At the point when a gadget (a customer) needs to send information to the broker, we consider this activity a “publish,” and when they want to receive data, it is known as a “subscribe.” in our case, as the shown Figure 4 the OpenHAB client connects to receive the message while the tool is the client to publish the message.

For the transformation of data from the simulation tool to the OpenHAB to happen, it needs to configure two files:

- Broker: This bridge represents an MQTT Broker connection, configured and managed by this binding.
- Supported Channels

The extension of broker connection bridges with a channel: This channel is activated when a worth is distributed to the designed MQTT on the broker connection. The event payload will be the received MQTT topic and its worth. Finally, the execution of the transfer process is dependent on two of the configuration parameters, which are: Client sitting and Payload. Each will be created through the simulation tool by the researcher. MQTT works as a broker, connecting its

simulator data (Client) to its client, OpenHAB, and configuring connections to brokers via OpenHAB Things.

Scenario: A series of human events at smart home that the researcher wants to emulate. Each human event involves a set of activities on the smart home. each action has a specific state, date, and time.

Simulation Tool

The GUI interface which allows the researcher to do the simulation function of the given Scenario. Furthermore, it provides the user with the ability to create all related files that are required to display the human activities (IoT devices) of the scenarios on the OpenHAB interface. Below, The GUI interface components will be explained along with an explanation of the main function assigned for each.

(1) Simulate Human Event Service: Simulates the real environment, like the real home. It consists of smaller environments like floors that contain rooms such as a living room, bathroom, dining room, etc. Furthermore, it simulates devices of the home such as the air conditioner, TV, fan and so on. There are many kinds of IoT devices and sensors which, in indeed, can be presented as humidity, light, temperatures, etc. The user can simulate the scenario for the home by controlling the home devices status, determining the date, time, and how many days the action will be repeated. As shown in Figure 5, the process of simulating given scenario occurs through steps which eventually arrive at the configuration of the payload file, that the connector phase dependant on.

The most important features in this section will be detailed as follows: As shown in Figure 5, assume that the user wants to simulate two events for two devices; light and TV. The light being located on the first floor of the house inside

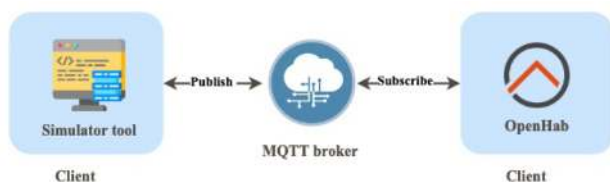


Figure 4: MQTT is based on clients and a server. MQTT server is called MQTT broker, the clients are connected devices. (publish/subscribe) operations to send and receive the data respectively.

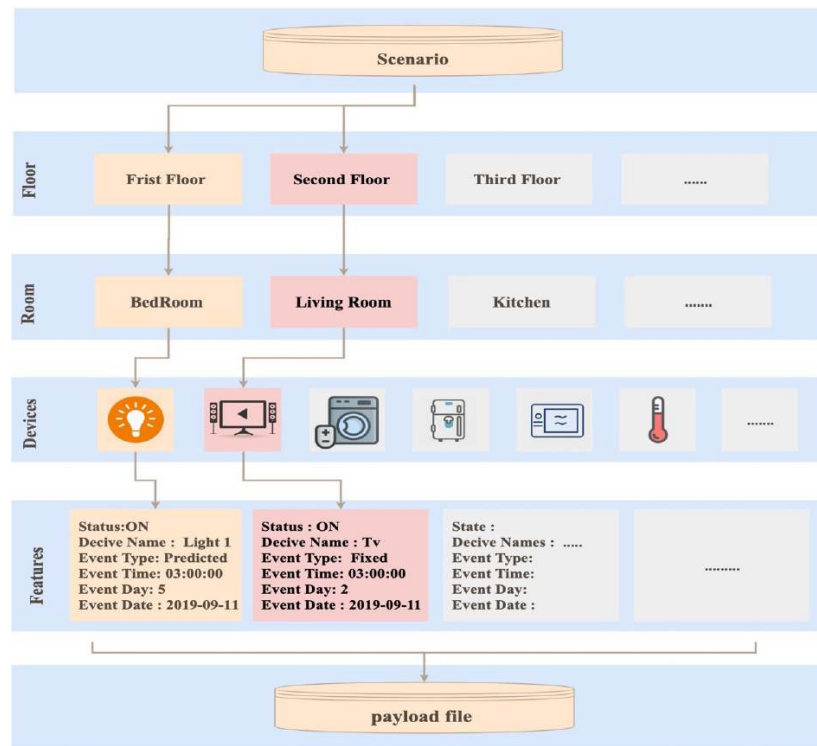


Figure 5: Simulate Human event service Container Hierarchy. It shows Features of Smart Devices that Located in a Room and floors – starting from the given scenario to the Payload file creation.

the Bedroom and the TV on the same floor, but in the Living room. Given the first device which is: The researcher assigned the status of the device as "ON", and gave it the name "Light 1". Now, they look for the event details. The researcher wishes to publish the event on to OpenHAB every three hours for five consecutive days starting from 11-09-2019. Because of that, the event type is labelled as "periodic". As for the second event; the TV, the status is "ON" and the device name is "TV", for the event details. The researcher plans on publishing this event on to OpenHAB at 03:00 pm for two consecutive days starting from 11-09-2019. Because of that, the event type is assigned as "Fixed". The TV certainly will turn "ON" at 3:00 pm on 11-12 Sep 2019.

As a result, the benefits of this section are the ability to control and manage the events that are simulated with the possibility of determining many features such as name and location, furthermore, the ability to repeat events for specific dates and times easily without the need to reformulate the event again. Consequently, getting the Payload file, enables the publishing of those events on OpenHAB using the MQTT protocol.

(2) MQTT Broker Bridge Service (server): One of the objectives of this tool is to gain the ability to publish simulation data (that was built previously on the simulation section) on to the OpenHAB platform using the MQTT, therefore, there are some steps which are required to set up the communication to OpenHAB allowing the retrieval of data. One of the most important steps is the creation of a broker file within OpenHAB. Through this GUI, there is an opportunity to configure the file where there are subsequent files built depending on this data. This tool allows (server) Configure Bridge Configuration files with required parameters such as MQTT ID, host name, port Username and Password. As illustrated in Figure 6, that configuration file needs to set up the MQTT binding on OpenHAB. Linking the MQTT broker with OpenHAB is done by adding the (.things) file on \$OPENHAB/CONF/THINGS through the terminal.

(3) Channel Configuration Service: The next step after establishing the broker file is to create channel files that allows display devices within the OpenHAB interface. The value of the channel configuration file is its ability to extend the MQTT broker connection bridges, therefore, some data is based on the data previously given in the broker file as the MQTT ID and the broker host, whilst, location and

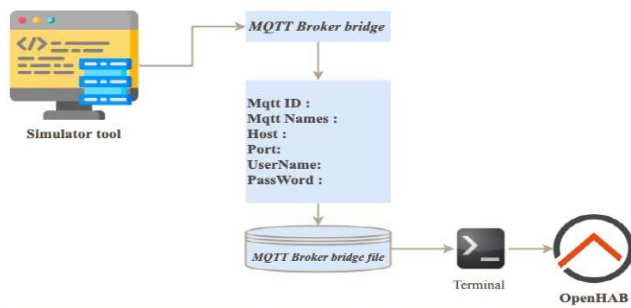


Figure 6: MQTT Broker bridge service. It shows the configuration Features of the bridge file – Finishing with the step of setup into OpenHAB.

Topic parameters are based on payload information. Other required parameters can be set easily using a clear method for researchers, such as (Things ID, Things Name and the Channel Type). In the end, as shown in the Figure 7, the channel file configured (.things), is ready to be stored in \$OPENHAB/CONF/THINGS.

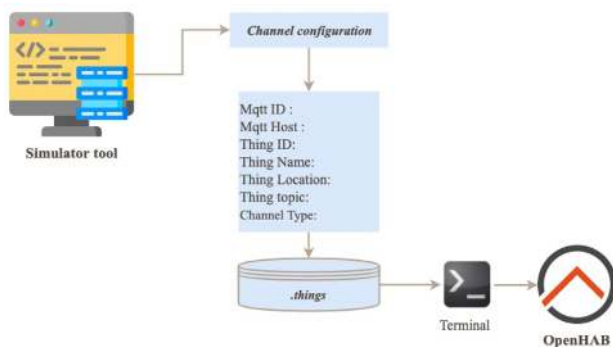


Figure 7: Channel configuration service: It shows the configuration Features of the Channel file – Finishing with the step of setup into OpenHAB.

(4) Items File Service: The final step is to configure OpenHAB to display smart devices through its interface. This involves the creation of item files to allow its exhibit of the status of simulation devices more clearly. As the users of OpenHAB usually do with the reality IoT devices on a real smart home. Items can be linked to channels that manually define Things inside OpenHAB UI. Once an item is associated with a channel, the status of a device is shown in OpenHAB UI.

As in the channel file, some parameters of the items depend on the previous data assigned to the bridge and channel files. The configuration file parameters are: MQTT Name, Thing Name, Item Name, Item Label, Item Type. Determining the category of the items shows the item icon. Finally, As shown in Figure 8, integrating the items with OpenHAB

using the (.items) file \$OPENHAB/CONF/ITEMS through the terminal.

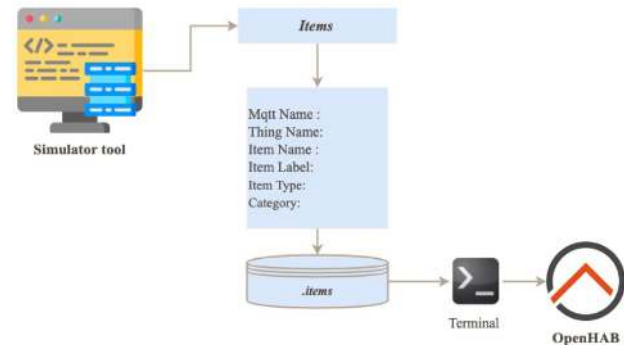


Figure 8: Items configuration service: It shows the configuration Features of the items file – Finishing with the step of setup into OpenHAB.

(5) Connector Execution: The main purpose of creating this tool is to allow the researcher to directly publish a regenerated payload file based on the sequence of events identified in the simulation of human events to the OpenHAB platform via MQTT protocol. As shown in Figure 9, Python code was written to perform this task. The basic files for running this function bind the simulation data into the OpenHAB platform as the sitting client File and payload file. According to the sitting client parameters, MQTT Host, Port, username and password will be generated directly dependent on MQTT Broker bridge values and the Payload file will be generated based on the data of the Simulate Human events file.

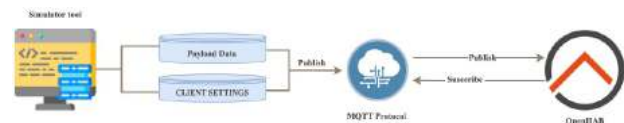


Figure 9: Connector execution: It shows the publish operations of Payload file and client sitting file – through MQTT protocol into OpenHAB.

REFERENCES

- [1] Hande Alemdar, Halil Ertan, Ozlem Durmaz Incel, and Cem Ersoy. 2013. ARAS human activity datasets in multiple homes with multiple residents. In *Proceedings of the 2013 7th International Conference on Pervasive Computing Technologies for Healthcare and Workshops, PervasiveHealth 2013*. <https://doi.org/10.4108/icst.pervasivehealth.2013.252120>
- [2] Nasser Alshammari, Talal Alshammari, Mohamed Sedky, Justin Champion, and Carolin Bauer. 2017. OpenSHS: Open smart home simulator. *Sensors (Switzerland)* (2017). <https://doi.org/10.3390/s17051003>
- [3] Arni Ariani, Stephen J. Redmond, David Chang, and Nigel H. Lovell. 2013. Simulation of a smart home environment. In *Proc. of 2013 3rd Int. Conf. on Instrumentation, Communications, Information Technol., and*

The image displays the user interface of the Simulation Tool, which is divided into five main sections, each with a title and a set of input fields and buttons.

- SIMULATE HUMAN EVENTS:** This section has a title bar with the text "By this section, you can create a custom file." Below the title bar, there are input fields for "Name", "Event Name", "Event Date", "Event Time", "Event Type", "Event State", and "Event Data". There are also buttons for "Generate Event File", "Download", and "Copy".
- MQTTBROKER BRIDGE:** This section has a title bar with the text "By this section, you can create MQTT Broker file." Below the title bar, there are input fields for "ID", "Name", "Host", "Port", "Username", and "Password". There are also buttons for "Generate MQTT Broker File", "Download", and "Copy".
- CHANNELS:** This section has a title bar with the text "By this section, you can set up Channels file." Below the title bar, there are input fields for "Channel ID", "Channel Name", "Channel Type", "Channel State", "Channel Data", and "Channel Type". There are also buttons for "Generate Channel File", "Download", and "Copy".
- ITEMS:** This section has a title bar with the text "By this section, you can set up Items file." Below the title bar, there are input fields for "Item Name", "Item Type", "Item State", "Item Data", "Item Type", "Item State", and "Item Data". There are also buttons for "Generate Item File", "Download", and "Copy".
- CONNECTER:** This section has a title bar with the text "By this section, you can generate the connectivity file to connect, already." Below the title bar, there are input fields for "Connect to OpenHab Through Python Script" and "Connect to OpenHab Through Python Script". There are also buttons for "Generate Connectivity File", "Download", and "Copy".

Figure 10: User Interface of the Simulation Tool

- Biomedical Engineering: Science and Technol. for Improvement of Health, Safety, and Environ., ICICI-BME 2013.* <https://doi.org/10.1109/ICICI-BME.2013.6698459>
- [4] Ibrahim Armac and Daniel Retkowitz. 2007. Simulation of smart environments. In *2007 IEEE International Conference on Pervasive Services, ICPS*. <https://doi.org/10.1109/PERSER.2007.4283934>
 - [5] Kevin Bouchard, Amir Ajroud, Bruno Bouchard, and Abdenour Bouzouane. 2010. SIMACT: A 3D open source smart home simulator for activity recognition. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-642-13577-4_47
 - [6] Alex Brice. 2018. Best of open source smart home: Home Assistant vs OpenHAB. <https://smarthome.university/your-smart-home-platform-home-assistant-vs-openhab/>
 - [7] Mario Buchmayr, Werner Kurschl, and Josef Küng. 2011. A simulator for generating and visualizing sensor data for ambient intelligence environments. In *Procedia Computer Science*. <https://doi.org/10.1016/j.procs.2011.07.014>
 - [8] Center for Advanced Studies in Adaptive Systems (CASAS). [n. d.]. Welcome to CASAS. <http://casas.wsu.edu/datasets/>
 - [9] Sophie Chabridon, Amel Bouzeghoub, Anis Ahmed-Nacer, Pierrick Marie, and Thierry Desprats. 2017. Unified modeling of quality of context and quality of situation for context-aware applications in the internet of things. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-319-57837-8_30
 - [10] D J Cook, A S Crandall, B L Thomas, and N C Krishnan. 2013. CASAS: A Smart Home in a Box. *Computer* 46, 7 (2013), 62–69. <https://doi.org/10.1109/MC.2012.328>
 - [11] Jessamyn Dahmen, Brian L. Thomas, Diane J. Cook, and Xiaobo Wang. 2017. Activity learning as a foundation for security monitoring in smart homes. *Sensors (Switzerland)* (2017). <https://doi.org/10.3390/s17040737>
 - [12] Wenbo Ding and Hongxin Hu. 2018. On the safety of IoT device physical interaction control. In *Proceedings of the ACM Conference on Computer and Communications Security*. <https://doi.org/10.1145/3243734.3243865>
 - [13] Sumi Helal, Jae Woong Lee, Shantonu Hossain, Eunju Kim, Hani Ha-gras, and Diane Cook. 2011. Persim - Simulator for human activities in pervasive spaces. In *Proceedings - 2011 7th International Conference on Intelligent Environments, IE 2011*. <https://doi.org/10.1109/IE.2011.34>
 - [14] Home-assistant.io. 2019. Home Assistant vs. Hass.io - Home Assistant. <https://www.home-assistant.io/faq/ha-vs-hassio/>
 - [15] Zahra Forootan Jahromi, Amir Rajabzadeh, and Ali Reza Manashty. 2011. *A Multi-Purpose Scenario-based Simulator for Smart House Environments*. Technical Report. arXiv:1105.2902 <http://arxiv.org/abs/1105.2902>
 - [16] Jae Woong Lee, Seoungjae Cho, Sirui Liu, Kyungeun Cho, and Sumi Helal. 2015. Persim 3D: Context-Driven Simulation and Modeling of Human Activities in Smart Spaces. *IEEE Transactions on Automation Science and Engineering* (2015). <https://doi.org/10.1109/TASE.2015.2467353>
 - [17] J. Lundström, J. Synnott, E. Järpe, and C. D. Nugent. 2015. Smart home simulation using avatar control and probabilistic sampling. In *2015 IEEE International Conference on Pervasive Computing and Communication Workshops, PerCom Workshops 2015*. <https://doi.org/10.1109/PERCOMW.2015.7134059>
 - [18] Bokyoung Park, Hyeongyu Min, Green Bang, and Ilju Ko. 2015. The user activity reasoning model in a virtual living space simulator. *International Journal of Software Engineering and its Applications* (2015). <https://doi.org/10.14257/ijseia.2015.9.6.06>
 - [19] Jennifer Renoux and Franziska Klügl. 2019. Simulating daily activities in a smart home for data generation. In *Proceedings - Winter Simulation*

- Conference. <https://doi.org/10.1109/WSC.2018.8632226>
- [20] Rosslin John Robles and Tai Hoon Kim. 2010. Context aware systems, methods and trends in smart home technology. In *Communications in Computer and Information Science*. https://doi.org/10.1007/978-3-642-16444-6_20
- [21] Margaret Rouse, Sharon Shea, and Vendela Redriksson. 2018. What is smart home or building (home automation or domotics)? - Definition from WhatIs.com. <https://internetofthingsagenda.techtarget.com/definition/smart-home-or-building>
- [22] Veralia Gabriela Sanchez, Carlos F. Pfeiffer, and Nils Olav Skeie. 2017. A review of smart house analysis methods for assisting older people living alone. <https://doi.org/10.3390/jsan6030011>
- [23] SPEC INDIA. 2009. Advantages and Disadvantages of XML - SPEC INDIA. <https://www.spec-india.com/blog/advantages-and-disadvantages-of-xml>
- [24] J. Synnott, L. Chen, C. D. Nugent, and G. Moore. 2014. The creation of simulated activity datasets using a graphical intelligent environment simulation tool. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2014*. <https://doi.org/10.1109/EMBC.2014.6944536>
- [25] Jonathan Synnott, Chris Nugent, and Paul Jeffers. 2015. Simulation of smart home activity datasets. *Sensors (Switzerland)* (2015). <https://doi.org/10.3390/s150614162>
- [26] Emmanuel Munguia Tapia, Stephen S. Intille, and Kent Larson. 2004. Activity recognition in the home using simple and ubiquitous sensors. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2004). https://doi.org/10.1007/978-3-540-24646-6_10
- [27] Matthias Templ, Bernhard Meindl, Alexander Kowarik, and Olivier Dupriez. 2017. Simulation of synthetic complex data: The R package simPop. *Journal of Statistical Software* (2017). <https://doi.org/10.18637/jss.v079.i10>