DEVS-IoT: performance evaluation of smart home devices network

Majeda Albataineh & Moath Jarrah

ONLINE FIRST

Multimedia Tools and Applications
An International Journal

ISSN 1380-7501

Multimed Tools Appl
DOI 10.1007/s11042-020-09186-w

Springer

Multimedia Tools and Applications
https://doi.org/10.1007/s11042-020-09186-w

DEVS-IoT: performance evaluation of smart home devices network

Majeda Albataineh [1] **· Moath Jarrah** [1]

Received: 20 December 2019 / Revised: 21 April 2020 / Accepted: 4 June 2020
Published online: 21 June 2020

Abstract
Advances in electronics and connectivity have enabled a wide range of applications that can harness data collection for better decision making and an improved lifestyle. The Internet of Things (IoT) provides the communication infrastructure that allows devices with sensing and control capabilities to be connected within a home network. Smart home systems are considered one of the prominent applications in IoT, where it is possible to control home devices to achieve a better usage in terms of cost and comfort. However, smart home networks contain a wide range of devices and finding an optimal schedule for their working hours is an NP-hard problem. Hence, rather than using mathematical optimization to find optimal solutions, this paper proposes a modeling and simulation methods in order to provide good decisions and recommendations for devices' scheduling. Discrete Event System Specification (DEVS) formalism is used to develop a model of a smart home network. The devices are categorized into two groups: monitoring devices and control devices. Monitoring devices include sensors that capture climate, energy, power, performance, and occupant's behavioral data. Control devices send signals remotely for setting and controlling different devices in the smart home network. The behavior in terms of power usage and cost is simulated under different scenarios and settings. The simulation results show that less energy consumption can be achieved if users adopt a behavior where the schedule of three devices is changed every week. As a result, the proposed method can be utilized to make better decisions in setting devices parameters and evaluating the performance of the smart devices network under different conditions, scenarios, and settings.

Keywords DEVS formalism · Smart home network · Scheduling · Modeling and simulation · Poisson process

✉ Moath Jarrah
mjarrah@just.edu.jo

[1] Department of Computer Engineering, Jordan University of Science and Technology, P.O. Box 3030, Irbid 22110, Jordan

1 Introduction

Smart home technology is becoming more attractive to the end users were new devices, sensors, and appliances are capable of communication in the era of Internet of Things (IoT) [27, 35]. The technology enables a real-time interactive response between intelligent devices and users [7, 17]. IoT, control, communication, computer, optimization, and image display technologies are used to connect smart components to the network in order to fulfill the system's requirements and to achieve efficient management and control [4, 29]. However, the technology poses challenges that need to be addressed such as large-scale management, heterogeneity, energy consumption, security, and big data analysis [3, 19, 31, 36].

A study of twenty smart homes in the UK was conducted in the period from 2013 to 2015 [13]. The study produced a dataset that was collected through surveys, interviews, and the working patterns of different sensors. The research in [23] used the dataset in [13] to investigate the heating service in different smart homes with a focus on the heating cost. Seven heating behaviours were studied. Moreover, a comparative study was conducted by the authors of [42] using the datasets to investigate the advantages and disadvantages of the smart home technology. Based on the study, the authors suggest policy recommendations to efficiently use the technology.

People are interested in smart home technology because of its significant impact that may change their lives [8, 40]. Understanding of the technology is an essential factor for users to change their lifestyles to suit the environment. Rather than using mathematical optimization to find optimal solutions such as the work in [18, 21, 22], we focus on simulations to suggest better behaviors and operational settings. This is because the scheduling of devices in large-scale environments is an NP-hard problem [12, 24, 28, 39]. Hence, the dataset in [13] is used in this paper to explore different devices behaviors in order to provide recommendations for smart home users.

This paper uses the Discrete EVent System specification (DEVS) formalism to model and simulate a smart home network system that consists of four rooms with different types of devices and appliances. The framework shows its users the power consumption and the cost under different scenarios and settings. Hence, users can control and change their home devices settings and their behaviors in order to save energy and reduce the overall cost.

The proposed system aims at helping users in changing their devices operational patters. The smart home network system consists of DEVS atomic and coupled models. The modelled house has four rooms and each room has several devices, appliances, and sensors according to the dataset and the parameters in [13]. Each device is modelled as an atomic model that communicates with other devices in the same room or in other rooms using XML messages. Three categories of XML files are used in the simulation of the system. The first category contains XML files that are created by each atomic model which corresponds to a device, an appliance, or a sensor. These files contain reported data based on the devices behaviors and power consumption. For example, a temperature sensor atomic model records the temperature values, the sensing time, the sensor ID, and sensor settings that include the model, time for the next sensing value, the working start day, in addition to other related parameters. The second category of XML files covers the communication and controlling messages. These files are used by the atomic models to receive messages, send messages, and/or send control signals to other atomic models. The third category contains a set of files for each room (RoomDataCollection) XML file. These files store all recorded data of each room and will be used by the smart meter atomic model which is connected to a room controller. In addition, there is one log file for each device that contains all the events that cause changing in the operational time or the settings of the device. Donald e.Knuth Poisson Algorithm is used in this paper to generate random numbers to be used in changing the working hours for each day [26]. Simulations are performed to compare a fixed-time table (according to the dataset) and a system that changes its working hours based on Poisson distribution. The results show that changing the devices working hours periodically can achieve better total power consumption than a fixed working pattern. Hence, recommendations are provided to the users in order to assist them in following the simulation trends that are revealed.

The rest of the paper is organized as follows. Section 2 provides a related work. Section 3 briefly describes the smart home technology and the Discrete EVent System specification (DEVS) formalism. Also, it discusses the dynamic configuration feature of DEVS components. Section 4 presents the smart home network system structure within DEVS environment. Section 5 describes the dataset of the devices and their setting. In addition, it explains the methodology of the modelling and simulation of the proposed system. Section 6 presents the simulation results and provides a discussion and recommendation regarding the revealed trends. Section 7 concludes the paper.

2 Related work

Modeling and simulations help system designers in studying the given system under different settings and operational behaviors. For example, the work in [20] proposed a model that represents the power sources of a smart grid. The authors simulated the model using solar and wind profiles in order to meet the load requirements. Also, storage devices were modelled in order to store the excess in the generated energy which can be needed in periods of load peaks. Moreover, the Building Energy Management System (BEMS) was proposed in [30], which is a DEVS-based approach that decompose a system into subsystems. Adaptive and predictive mechanisms can be used in BEMS in order to control the energy consumption. Also, DEVS formalism was used in order to study the comfort level in smart homes in [1] using fuzzy methods. The work in [33] used the hidden Markov model to manage smart homes and reduce power consumption. Three states were identified for the devices which are active, away, and sleep.

An IoT architecture was presented for IoT smart home network systems in [19]. The architecture addressed the problem monitoring and remote controlling of smart home devices using a standard protocol. On another hand, the researchers of [32] discussed the challenges in a smart home city framework and the socioeconomic opportunities in a city area. The work was based on a European Union project that provides a testbed and implementation of smart city services [38]. Another study was conducted to demonstrate an in-depth use of smart home technologies for 10 households [37]. The result of the study reported four concerns that need to be addressed. First, the smart home technology can be socially and technically disruptive. Second, for the households to be able to use the technology, adaptation and familiarity are required, which in some cases can limit their usage. Third, learning to use the technology is time-consuming and can be a challenge for many users. Finally, there is a lack of evidence that energy can be saved using smart home technologies.

In addition, energy saving is an important aspect that would encourage the householders to adopt the smart home technology. The study in [14] focused on introducing smart home technology in Singapore and how to use it for saving energy and reducing power consumption.

Four main areas were introduced in the study. First, a review of Singapore power consumption. Second, Singapore policies and programs that are related to the energy saving. Third, people behaviors can result in saving energy. Finally, the smart home energy saving perspective for householders. In addition, the work in [6] shows that services that are offered by smart home technology are not limited to device management and energy consumption. It can offer a variety of comprehensive range services including security, appliances control and management, living assistance, and remote monitoring.

Smart homes, smart grid, and IoT technologies are being developed in order to create the smart cities concept. Using IoT technology and applying it in different services in smart environments can create intelligent perspectives of these services [5]. Three main groups of nine themes were introduced in [5]. The first group views smart home functionality, instruments, and socio-technical. The second group describes the users and the use of a smart home, prospective users, decisions, and interactions. The third discusses the possible challenges in achieving smart home technology in terms of software, hardware, design, and domestication.

3 Background of smart homes and DEVS formalism

3.1 Smart home technology

A smart home is a system where devices are automatically controlled to satisfy the end user and the surrounding environment [29]. Smart technology, computers, network, and other technologies are used to control devices, sensors, and appliances in the house, such as controlling the temperature, house lighting system, security system, entertainment system, and other tasks. Smart home devices and appliances are integrated and controlled as a single machine either remotely or locally using suitable protocols. Moreover, the devices can communicate with each other through a home controller in order to be managed effectively. Three essential elements or components exist in the technology which are [29]:

1. Single Functionality Devices: these are devices that operate to accomplish a task and can be controlled automatically or remotely. Examples include lighting, energy, temperature, soil quality, security, entertainment, different types of sensors, meters, detectors, and switches.
2. Centralized Control Unit: represents a device such as a mobile phone, control panel, tablet, or any device that can control, manage, and receive signals from functionality devices using an inner communication protocol. The control unit communicates with other devices using a networking method such as Bluetooth, infrared, and Wi-Fi.
3. Protocols: these cover the standard communication and interaction methods and components. They aim to solve heterogeneity and interoperability problems where different types of devices need to send and receive messages with other devices and with control units. In addition, different smart homes can communicate and exchange relevant data with central servers. Figure 1 shows a sketch of four smart homes.

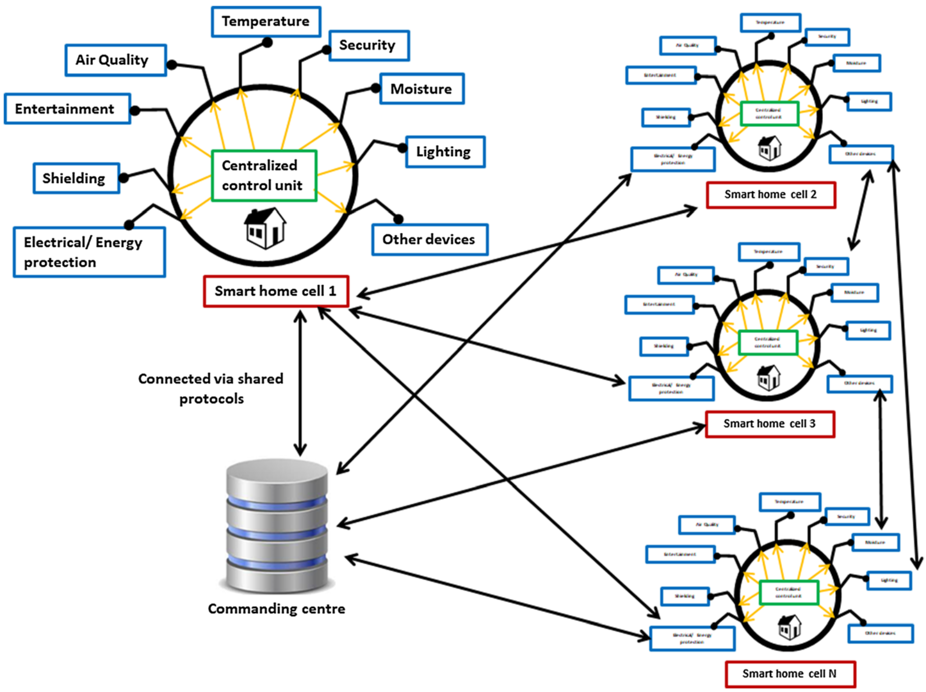

Fig. 1 A system of smart home cells and a commanding center [29]

3.2 DEVS modeling and simulation formalism

DEVS is a timed event system, powerful and flexible modular formalism for modeling, analyzing, and simulating discrete or continuous event systems. It was invented in 1976 by Bernard P. Zeigler. Using DEVS allows system designers to define systems behaviors using states, input, and output events. It also defines the entire system structure [16, 41, 43].

DEVS classifies a system behavior using two levels (atomic and coupled). The atomic level describes the behavior of discrete event systems in term of transitions between sequential states and the reaction of the model when receiving external inputs [43]. The coupled level defines a system as a network of atomic components. Coupled and atomic models can be combined to produce a hierarchical structure.

A basic or atomic model in DEVS formalism consists of the following [2, 10]:

- Input ports are used to receive external inputs. The inputs represent external events.
- Output ports are used to send messages or events to other models.
- States are the different phases or states that a model can have. A state is represented by its name (phase) and a sigma parameter.
- The time advance function is responsible of providing the time that causes a state transition.
- The internal transition function causes a transition to the next phase when the current state's time is elapsed.
- The external transition function causes a transition to a new phase as a result of receiving an external event on the input ports. Three parameters are used to determine the next phase

which are: the current state, the port where the input has occurred, and the values of the input.

- The confluent transition function is executed under the cases where the internal and external functions are causing a transition simultaneously. Hence, the confluent function resolves any conflict in the next phase/state.
- The output function is used to generate messages that are to be sent through the output ports. This function is executed right before an internal transition occurs. Fig. 2 shows the eight components of an atomic model.

3.2.1 Atomic model formalism

The DEVS atomic model formalism is a structure that describes the different aspects of discrete event behaviors of a system which is given as below:

$$M = < X, Y, S, \delta_{ext}, \delta_{int}, \lambda, ta >$$

Where, the time base is a continues real number ($\mathbf{T} = \mathbb{R}$). **X** is the set of inputs ($\mathbf{X} = X_1, X_2 \ldots, X_m$). **Y** is the set of outputs ($\mathbf{Y} = Y_1, Y_2, \ldots, Y_L$). **S** is the state set and represents the different states of the system ($\mathbf{S} = S_1, S_2, \ldots, S_N$). $\boldsymbol{\delta_{ext}}$ is the external function. When an input X_i is received on the input ports, it might cause a transition from the current state. The result from an external event depends on three main elements which are: the current state of the model, the input, and the elapsed time. Hence, $\boldsymbol{\delta_{ext}} = Q \times \mathbf{X} \rightarrow \mathbf{S}$, where Q is the total state set ($Q = \{(S_i, \mathbf{e}) \mid S_i \in \mathbf{S}, 0 \leq \mathbf{e} \leq \mathbf{ta}(S_i)\}$). **e** is the elapsed time and takes a value in the range from 0 to $\mathbf{ta}(S_i)$. **ta**(.) is the time advance function that keeps track of the remaining time before an internal transition causes the model to go into the next state. $\boldsymbol{\delta_{int}}$ is the internal transition function that describes the transitions from one state to the next sequential state after the expiration of $\mathbf{ta}(S_i)$ for the current state ($\boldsymbol{\delta_{int}}$: $\mathbf{S} \rightarrow \mathbf{S}$). **λ** is the output function that generates output events (i.e. sends messages to other models). **λ** is invoked only when an internal transition is occurred. For more details, please refer to [43].

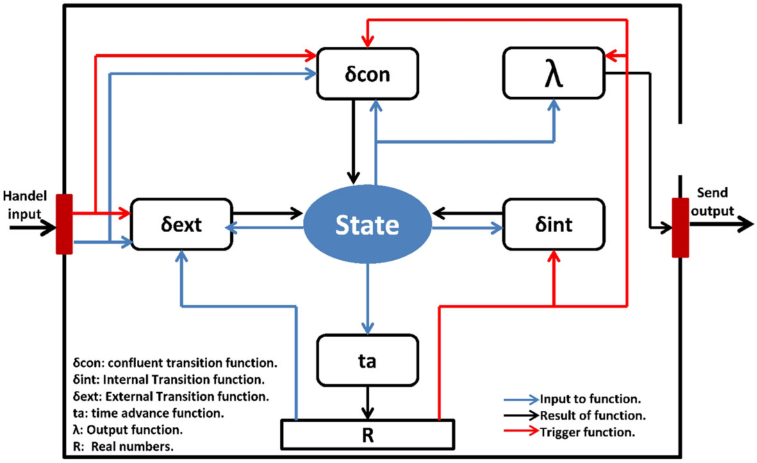

Fig. 2 DEVS elements [43]

3.2.2 Coupled model formalism and hierarchical model construction

A coupled model is a connection between several atomic models to form a new model and can be a component in a larger model structure as shown in Fig. 3. It contains the following elements [43]:

1. A set of basic models (atomic models).
2. A set of input and output ports to receive and send external events, respectively.
3. Coupling specifications which can be either an external input, an external output, or an internal coupling. The external input coupling is a coupled input that is directly connected to one of the atomic/coupled models within a higher coupled model. The external output is an atomic/coupled output within a higher model that is directly connected to one of the coupled model output ports. The internal couplings refer to internal connections between the different models within a higher coupling model. The three coupling types are shown in the top right of Fig. 3.

In summary, there are two types of DEVS components. First, the atomic model which represents the basic model in DEVS. Second, the coupled model which represents a set of atomic models (basic models) that have internal and external coupling specifications. Atomic and coupled models together form a larger (hierarchical) structure as shown in Fig. 3.

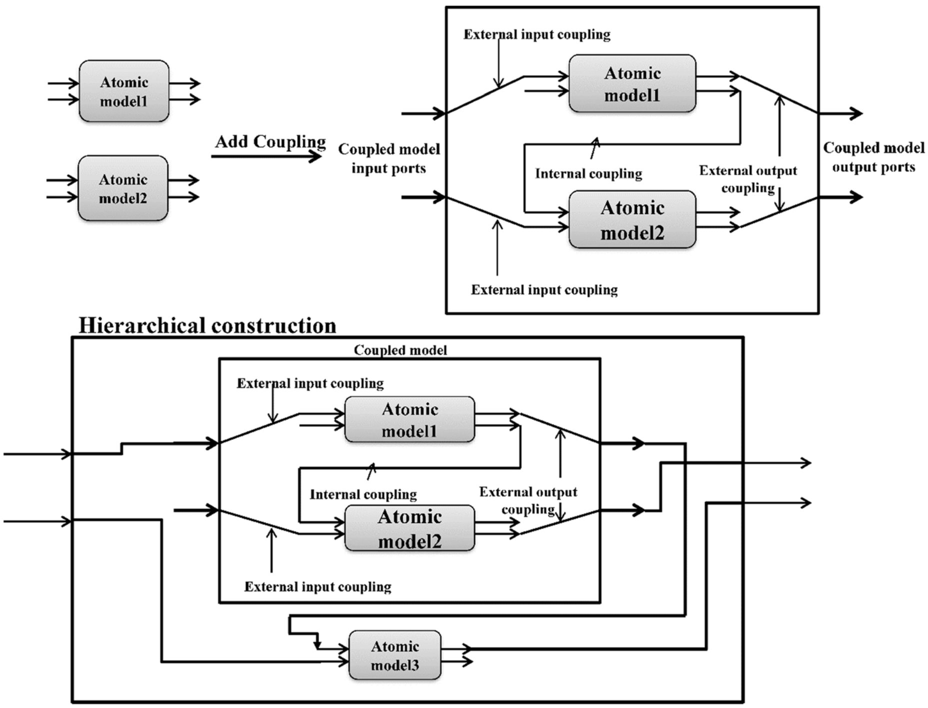

Fig. 3 Components coupling [43]

3.2.3 Dynamic reconfiguration in DEVS

Dynamic Reconfiguration is the system ability to dynamically change it is structure and interface in response of different situations, as well as providing an environment for modeling and simulation of the component-based complex systems with high capacity and flexibility [15]. For examples, an engineer needs to dynamically add or remove nodes in a distributed computing system according to the system workload. In such complex system, a dynamic reconfiguration modeling with high capacity is needed such as DEVS environment.

The dynamic reconfiguration of component-based modeling and simulation involve many advantages such as [44, 45]:

- Adds flexibility for the development of different systems.
- Provides practical ways to model complex systems to reflect structural and behavioral changes.
- In large systems, it can load sub-system components for simulation.

The reconfiguration feature provides different forms of component's manipulation for component-based system designers which are [9]: adding and removing of a component to the system during the simulation, adding or removing a coupling/connection between different components during the simulation, updating a component to produce a new version with new interface and behavior, and moving a component from one location to another.

The variable structure of a system refers to a dynamic change on the system's inner components and/or the relationships between the components [11].

The operational boundaries of a components-based systems that result in a structural change are:

- *addModel()*: is used by a model to add other component models for its parent model.
- *removeModel()*: is used by a model to remove itself and its parents.
- *addCoupling()*: is used to add coupling/connections.
- *removeCoupling()*: is used to remove coupling/connections.

In addition, DEVS provides reconfiguration features to dynamically add and remove input and output ports using the methods: *addInport()*, *addOutport()*, *removeInport()*, *removeOutport()*.

The following is an example of a temperature sensor system using the dynamic reconfiguration feature. The system has eight temperature sensor models that are connected to an atomic *Receiver* model. The *Receiver* collects temperature data and send them to the *DataRecord* atomic model to store them. When the *Receiver* receives a value greater than 40 °C, it sends the value to the *Display* atomic model which in turns displays it on the output port of the system as shown in Fig. 4.

If the size of data queue at the *Receiver* model becomes greater than 50 temperature values, the system automatically generates a new *Receiver* atomic model, as well as a new *DataRecord* model. The original *Receiver* removes the coupling of the last four sensors and four of its input ports. Then it adds a coupling of last four sensors to the new *Receiver* after removing four input ports from the new *Receiver*. The goal is to distribute the load between two *Receiver* models. In addition, a new input port is added to the *Display* model and a new coupling between the new *Receiver* output port to the *Display* model. Finally, a new *DataRecord* is added with a coupling with the new *Receiver* model as shown in Fig. 5.

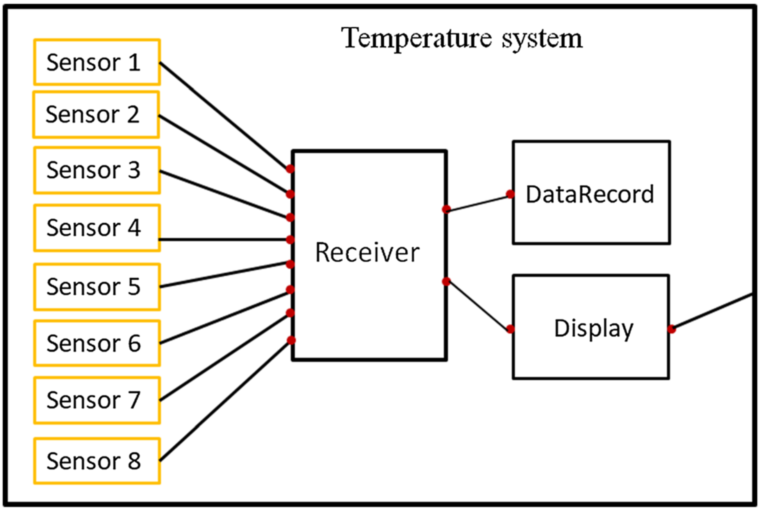

Fig. 4 A temperature system

4 Smart home network system model

The smart home network that is modeled and simulated consists of four rooms as shown in Fig. 6. The rooms are: kitchen, bedroom, living room, and a bathroom with laundry services. The devices in each room will be discussed in Section 5. The devices power ratings and working hours are set according to the dataset [13, 23].

4.1 Atomic models

Five atomic models are defined in the control devices category. Detailed DEVS specifications of the first model is given. The specifications of the other four models follow the same

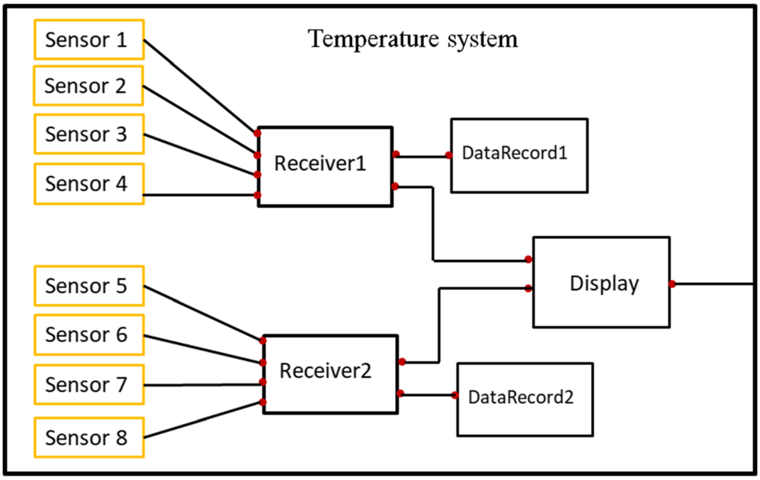

Fig. 5 A variable structure and interface of a temperature system

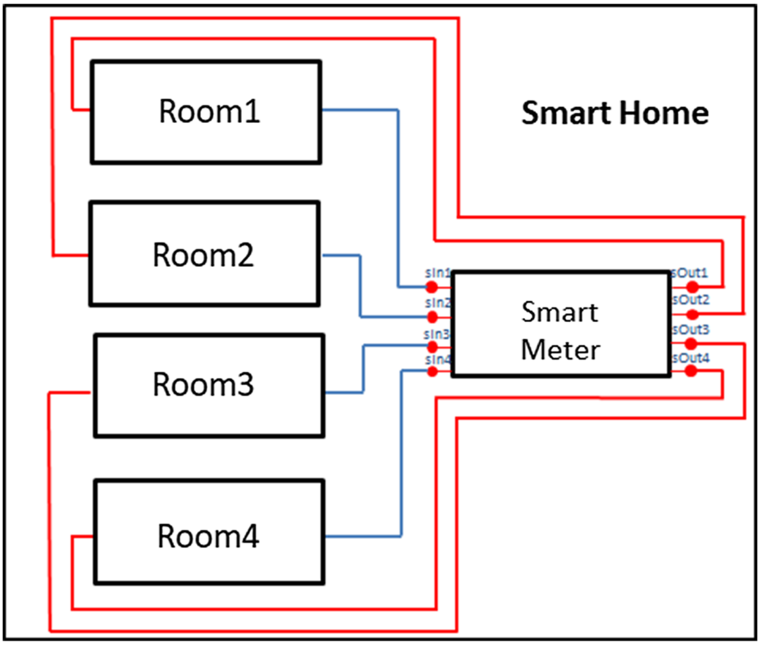

Fig. 6 A 4-room smart home model

approach. Due to space limitation, we provide a description of the other four models without detailed DEVS specifications. The models are:

1- *Identifier*: it receives the ID, data, and control messages from the devices monitoring devices in a room. If the received message is an Id message, the *Identifier* atomic model sends it the *Controller* atomic model. If the received message is a data message, it will be sent to the *Recorder* model. Data messages are information that are recorded by a device such as the room temperature. If the received message is a control message, it will be sent to *Signal Controller* atomic model. Control messages can be ON or OFF signals. The *Identifier* model keeps a record of the connected devices in a room. It supervises any dynamic reconfiguration of the system in cases of adding, removing, and replacing room devices. The *Identifier* has four to six input ports and four output ports for sending and receiving data.
2- *Recorder*: it receives data messages from the *Identifier* and stores the data in the collection XML file which contains the device ID, the name of the room, the reported data, and the reporting time. The *Recorder* receives data from different devices.
3- *Signal Controller*: it receives the ID messages from control devices and sends ON/OFF control messages accordingly to the target devices. The *Identifier* determines the devices that need to be turned ON or OFF based on the daily schedule. The ON/OFF messages are then sent to the *Signal Controller* which in turns communicate with the target devices through four to seven dynamically established connections (output ports).

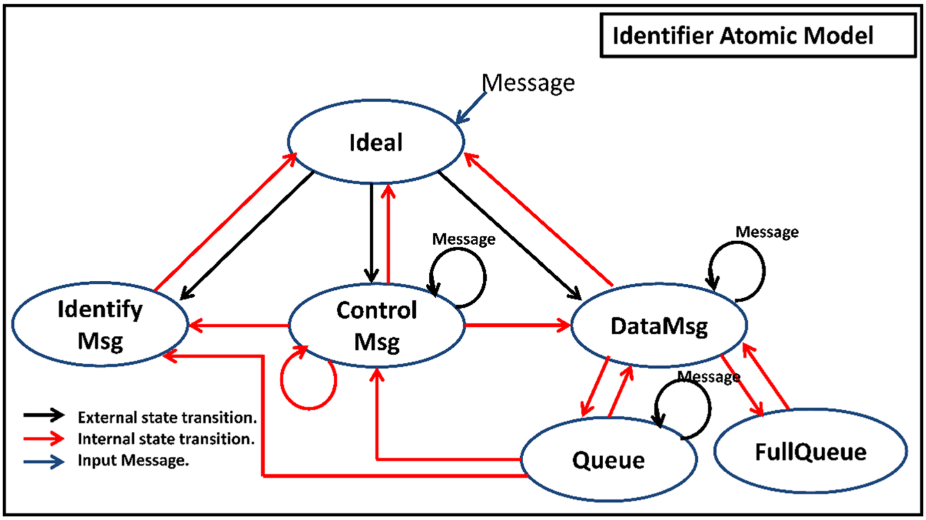

Fig. 7 Identifier atomic model transitions

4- *Controller*: it dynamically creates extra *Recorder* and *Identifier* atomic models when the size of the queue exceeds a threshold. The original *Identifier* continuously checks the queue size and when the size becomes greater than 25, the model sends a full-queue message to the *Controller*. The *Controller* creates a second *Identifier* model and divides the connected devices between the original and the new model. Then it connects the new *Identifier* model with the new *Recorder* model. By distributing the workload, the delay in the message queue is reduced. The model does not have output ports and receives the *Identifier's* messages through an input port.

5- *Smart Meter*: it represents the monitoring device of the system. The model displays the relevant data and information about all home devices. Charts and tables were used in the implementation to present the information in a convenient way. There is one input and one output ports for each room. Once a device is connected in a room, the corresponding *Identifier* informs the *Smart Meter* of the device's ID and name. Upon receiving the message, the *Smart Meter* retrieves the device's data from an XML file for presentation. The presented data include working hours, power consumption, and the cost. The user can choose the simulation time based on a daily, monthly, or yearly basis. The charts and tables are generated while the simulation in an interactive fashion. Also, if a user changes the working hours of a device, the information is reflected directly on the *Smart Meter*'s generated results.

Figure 7 shows the state transitions of the *Identifier* atomic model and Listing 1 shows a formal DEVS description of the model. Each state has a sigma value (σ) that represents the time the system stays in that state. For example, the *Identifier* model stays in state *'DataMsg'* for a time unit of 1. Formal descriptions of other models can be derived in a similar way as shown in Listing 1.

Listing 1 Formal DEVS description of the Identifier atomic model

DEVS = (X, Y, S, δ_{ext}, δ_{int}, δ_{con}, λ, ta)
Initialization phase = Idle, σ = ∞, record, queuemsg = value on port, queuefull, queue.
InPorts = {"gIn", "gIn1", "gIn2", "gIn3", "gIn4", "gIn5", "gIn6"}, where
InPort ⊠ $G_{in,}$ X_{Gin} = {1, 2, 3, 4, 5, 6, 7}
X= {(p, v) | p ⊠ InPorts, v ⊠ {X_{Gin}}}
OutPorts = {"gOut"," gOut2"," gOut3"}
Y = {(p, v) | p ⊠ OutPorts, v ⊠ Y_{Gout}}
S= {"Idle"," DataMsg"," IdentifyMsg"," Queue"," fullQueue"} x $R^{+}_{0,\infty}$ x R
δ_{ext} (phase, σ, e, (p, v)):
δ_{ext} ("Idle", ∞, e, (p, v)) =
If (record.contains ("control")) ---->("IdentifyMsg", 0).
δ_{ext} ("Idle", ∞, e, (p, v)) =
If (record.Contains ("Data MSG:")) ---->("DataMsg", 1).
δ_{ext} ("DataMsg", 1,e,(p,v))= ("DataMsg", 1-e, queue.add(queuemsg)).
δ_{ext} ("Queue", 1,e,(p,v))= ("Queue", 1-e, queue.add(queuemsg)).
δ_{int} (phase, σ):
δ_{int} ("IdentifyMsg", 0) = ("Idle", ∞).
δ_{int} ("fullQueue", 0) = ("DataMsg", 1).
δ_{int} ("IdentifyMsg", 0) = ("Idle", ∞).
δ_{int} ("DataMsg", 1) =
if (queuefull == 1) ------> ("fullQueue",0).
Else if (queue not empty) -----> ("Queue",1).
Else ----------> ("Idle", ∞).
δ_{int} ("Queue", 1) =
if(record.contains("control"))------> ("IdentifyMsg",0).
Else if (record.contains("Data msg:")-------->("DataMsg",1).
δ_{con} (phase, σ, e, (p, v)): δ_{ext} (phase, σ, e, (p, v)), δ_{int} (phase, σ).
λ (phase,σ) :
λ (" DataMsg", 1) = record.
λ (" IdentifyMsg", 0) = record.
λ (" fullQueue", 0) = "Queue is full".
ta (phase,σ):
ta ("Idle", ∞) =∞.
ta("DataMsg",1) =1.
ta("IdentifyMsg",0) =0.
ta("fullQueue",0) =0.
ta("Queue",1) =1.

The second category of atomic models are the monitoring devices that send identification messages to the *Identifier* model. Based on a monitoring device's working schedule, the recording time and the data are stored in an XML file. In addition, a Poisson distribution is used to trigger independent events that cause changes to the schedule. Two modifications on the schedule can be inserted which are: 1) new working hours during a specific day and 2) the number of days until the consequent change. A log file is used to store the new modifications. The *Smart Meter* sends controlling signals to monitoring devices, and the *Identifier* receives messages from the devices. Three model types were determined based on the functionalities that they perform which are:

1- ***Climate: refers to climate sensors such as wind speed and direction, temperature, total and average horizontal solar irradiance, relative humidity, total rainfall, and average barometric pressure. Each device model has a unique XML file for data storing.***

2- ***Appliance: refers to different types of appliances such as washers and dishwashers, lamps, heaters, video players and recorders, computers, printers, toasters, routers, coffee makers, alarms, DAB radio, dryers, refrigerators, televisions, projectors and monitors. The model stores the device's electrical power usage in an XML file.***
3- *Meter*: refers to gas and electric power meters. The model measures the gas volume or the electrical power usage every 30 min and stores the data in an XML file.

The various atomic models are coupled together in a room as shown in Fig. 8.

5 Simulation methodology

5.1 Dataset and smart devices

The dataset that was used in conducting the simulation are based on the REFIT project [13, 23]. The project was a collaboration between three Universities in the United Kingdom in the period of May 2012 to October 2015. Twenty smart homes participated in the study where the total number of rooms was 389, the number of radiators were 252, the number of appliances in the 20 homes was 618, the number of the fixed heater was 19, the number of attached sensors to rooms was 1567, and the number of variable records provided by devices and sensors was 2457. The dataset includes parameters that were used in our simulation such as the devices/appliances initial settings and the working schedule. Below we present the different devices and their parameters, attributes, and data records. Thirty different devices are used in this paper. The devices can be categorized into two: monitoring devices (sensors) or home appliances. The data are recorder in XML format files. Table 1 shows a snippet of one of the XML files with a description.

Table 2 presents a list of the devices and appliances that were used in the simulation. Each device is assigned four values which are: the energy consumption per hour (kWh), the cost per hour, the initial setting of the working hours per day, and the device's hours limitation per day.

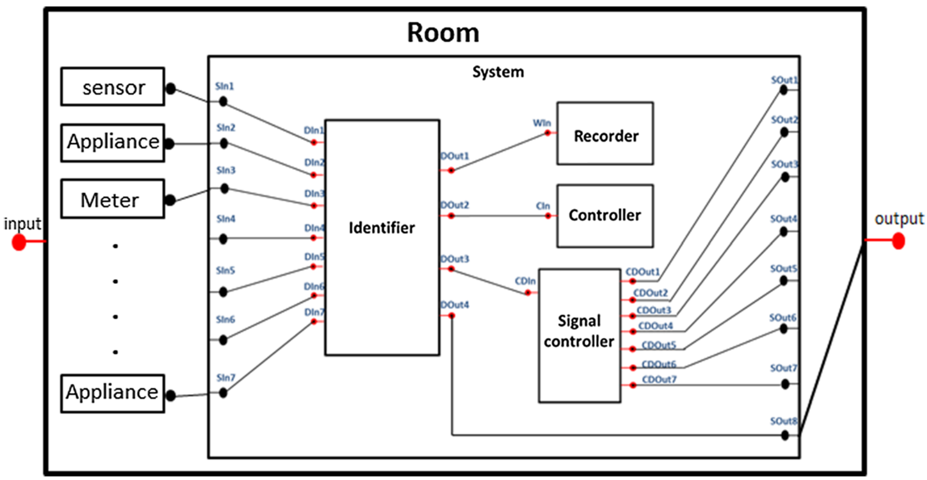

Fig. 8 Coupling of atomic models within a room

The dataset of [13] has been used to determine these values according to the USA average utility rate per kWh.

5.2 Poisson and uniform distributions within the simulation

The dataset provides a schedule of working hours and the rate of reported data for each smart home device. However, in order to build a system that simulates different user behaviors, Poisson Algorithm, as shown in Listing 2, is used to make changes on the pre-defined schedule (fixed-time table). Also, a Uniform random number is generated to represent the new working hours during a specific day.

The Poisson algorithm generates a discrete distribution time value. The value is used to specify the time when a device must change its working hour values. In this system, the algorithm parameter *Lambda* is assigned a value of 20. During the simulations, this value results in reasonable random numbers that fall in the range of one day (24 h). We have used different values for *Lambda*, and 20 was a suitable value that returns numbers in the range between 8 to 28. If the Poisson algorithm returns a value that is at least one day from the start of the system simulation time, or one day from the previous change that was applied on a device, then the value is rejected, and the algorithm is invoked again.

The Uniform random numbers, that are generated in the simulation at each new Poisson value, have specific values that satisfy the functionality of a device and/or the user's requirement. For examples, the vacuum cleaner should not work continuously for 10 h per day. Also, security cameras and refrigerators should not work for only 2 h per day. Hence, the Uniform random numbers are set according to the specifications of the devices.

A scenario that is simulated has the following description. A kitchen has an air filter that has a pre-defined schedule of 24 working hours per day (WHperD). A Poisson value is generated (T1 = 360 min) which causes a change in the schedule after 360 min are elapsed. Also, a Uniform value is generated (16 h) which results in changing the air filter working hours to 16 h. Moreover, a second Poisson value (T2 = 648 min) causes another change in the working hours (e.g. 13 h). Poisson and Uniform distributions can be used at any time while the simulation is running.

Listing 2 Donald Knuth Poisson algorithm [26]

```
L ←exp (-λ), K←0, P←1
Do
K←K+1
Generate a uniform random number U in [0,1]
P←P x U
While (P > L)
Report the value K+1
```

5.3 Simulation files

In the smart home network system simulation, four XML format files and one text file are used. The system designer configures the devices data to be simulated using two of the XML files. Also, she/he can pause the simulation and modify the files. The other files are generated

Table 1 Attributes of smart home devices and their data records

Appliance Attributes	Example	Data Record
A device's attributes are at the top of its corresponding XML file. A monitoring device: 1. StartDataTime 2. SensorId. 3. Manufacturer. 4. VariableType. 5. Units. 6. Model. A control device: 1. zoneControl. 2. onOffControl. 3. model. 4. manufacturer. 5. manualOverride 6. ID 7. fuelType. 8. Timer. 9. StartDataTime.	<Appliance ID = “Appliance12” startDateTime = “2013–10-01 T00:00:00Z” applianceType = “Kettle” plugIdRef = “Plug32” spaceIdRef = “Space6”/> < Sensor id = “Sensor1451” manufacturer = “CurrentCost” model = “Individual appliance monitor (IAM)” > <TimeSeriesVariable ID = “TimeSeriesVariable2361” variableType = “Electrical power” units = “W”/>	An example of a data record in XML < Record Time = “78.0” > <Data>1.7264942199847868 </Data> </Record>

Table 2 Operating values of appliances

Device name	Energy use (kWh)	Cost per hour ($)	Initial working hours	Hours limitation
Television	0.78	0.1	5	1–24
Refrigerator	1.8	0.23	24	24
Clothes Washer	0.51	0.07	2	1–10
Dishwasher	0.33	0.043	2	1–4
Microwave Oven	1.5	0.18	1	1–2
Water Heater	4.5	0.54	4	1–12
Toaster	1.1	0.13	1	1–2
Lamp	0.3	0.04	12	1–24
Hair Dryer	0.71	0.09	1	1–3
DVD player	0.017	0.01	4	1–8
Ceiling Fan	0.07	0.01	6	1–24
Clothes Dryer	2.79	0.33	2	1–8
Coffee Maker	1	0.12	1	1–3
Computer	0.075	0.01	8	1–24
Garage Door Opener	0.4	0.05	1	1–2
Computer Notebook	0.05	0.01	8	1–24
Deep Fryer	1	0.12	1	1–2
PC Monitor	0.043	0.005	8	1–24
Air Filter	0.0039	0.003	12	1–24
Iron	2.2	0.28	1	1–2
Printer	0.013	0.01	1	1–2
Receiver	0.056	0.01	5	1–24
Recharge Tools	0.052	0.01	4	1–10
Router	0.144	0.02	24	1–24
Room Heater	31.68	4.12	12	1–24
Vacuum	0.523	0.071	1	1–2
Stereo System	0.066	0.01	1	1–2
videogame System	0.144	0.02	3	1–12
Gas Meter	0.004	0.001	24	24
Air Temperature sensor	0.004	0.001	24	1–24

by the system components during the simulation for reporting data, messages, and other useful information. Below is a description of these files.

1. DevicesData: is an XML file that is prepared by the system designer. The file contains the data of the smart home devices. The simulation of this paper uses a file that contains information about 30 devices. Each device has ten attributes other than the device name and ID. The attributes are the device wattage, cost per hour, working hours, energy consumption per hour, energy consumption, and estimated cost. The file is used by the room *Identifier* atomic model.
2. ControlDevicesSchedule: is an XML file that is prepared by the system designer to set the schedule of sending control signals (e.g. ON/OFF) to the devices. The file is used by the *Smart Meter* and C*ontroller* atomic models.
3. DataRecords: it is an XML file that is generated by the devices during the simulation. It keeps all the data that the devices sense or collect. The total number of different

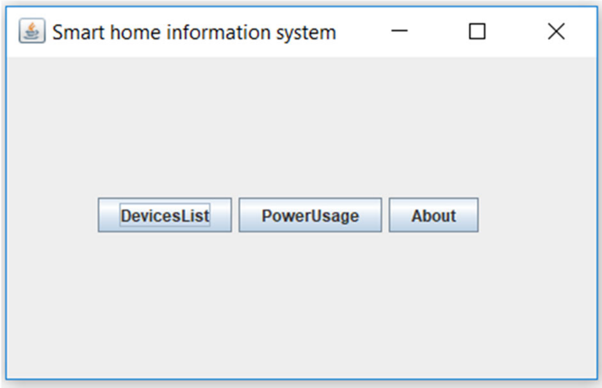

Fig. 9 Smart home information panel

fields in this file are four: the device name, the device ID, the time of recorded data, and the data values.

4. LogFiles: is a text file that is generated during the simulation and stores the events that trigger changes to the pre-defined schedule as a result of the Poisson distribution. The file has four fields which are: the time of the change, the new value, the changed

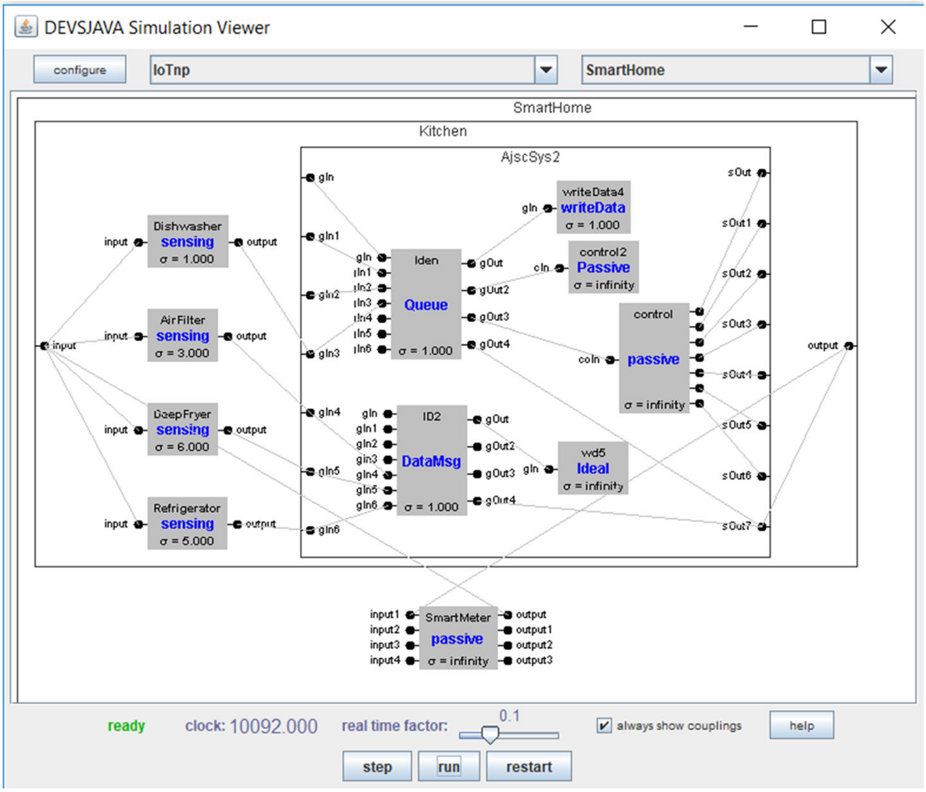

Fig. 10 The kitchen model in DEVS environment

Table 3 Devices in the kitchen

Device Name	Wattage (W)	Hours per day	Energy per day (Wh)	Cost per day ($)	Energy per week (Wh)	Cost per week ($)	Weekly in DEVS units
Dishwasher	330	8	2.64	0.34	18.48	2.38	3360
Air filter	50	24	1.2	0.16	8.4	1.12	10,080
Deep Fryer	1000	1	1	0.13	7	0.91	420
Refrigerator	255	24	1.8	0.23	12.6	1.64	10,080
Microwave	1500	3	4.5	0.59	31.5	4.10	1260
Toaster	1100	1	1.1	0.14	7.7	1	420
Coffee Maker	1000	2	2	0.26	14	1.82	840
Total	5235	63	14.24	1.85	99.68	12.97	10,080

value, and the time for the next change. This file is used by all devices in the smart home and modified during the simulation.

5. DataCollection: is an XML file that is used by the room *Recorder* atomic model to save information about the devices in a room. It contains the device ID, name, data, data recording time, and the operating hours of a device.

6 Simulation and evaluation

Simulations were performed to provide insights to homeowners that can help in managing the energy consumption. The first set of simulations evaluates the system performance using a pre-defined schedule for one room and for the 4-room smart home network. The pre-defined

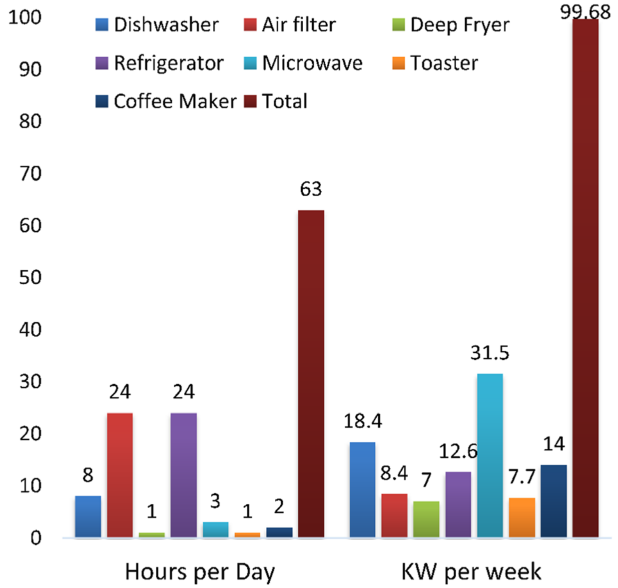

Fig. 11 Daily operating hours and weekly power consumption

Table 4 Devices in the bedroom

Smart Home Room 2: Bedroom							
Device Name	Wattage (W)	Hours per day	Energy per day (Wh)	Cost per day ($)	Energy per week (Wh)	Cost per week ($)	Weekly in DEVS units
Television	156	5	0.78	0.1	5.46	0.7	2100
Lamp	300	12	3.6	0.47	25.2	3.29	5040
Computer	75	8	0.6	0.08	4.2	0.56	3360
PC Monitor	42	8	0.34	0.04	2.38	0.28	3360
Printer	13	1	0.013	0.01	0.091	0.07	420
Recharge Tools	13	4	0.052	0.02	0.364	0.14	1680

schedule is based on the dataset in [13, 23]. The second set of simulations shows the results when the Poisson distribution is used to change the schedule and working hours. The results show that some scenario behaviors cause energy saving when compared with the fixed schedule; and hence, we recommend that users change their behaviors in accordance with the conditions that are revealed by the simulations.

Figure 9 shows the graphical user interface of the designed system, where a user can get information about the devices and the power usage. The devices information includes the device ID, type, model, and manufacturer. Also, its default start time and unit. The information is stored at the header of each device's XML file. The first button (DevisesList) a list of the devices and their attributes. The second button presents a bar chart that shows the total power consumption and cost. The third button (About) shows the structure and settings of the home network.

Table 5 Devices in the living room

Smart Home Room 3: Living Room							
Device Name	Wattage (W)	Hours per day	Energy per day (Wh)	Cost per day ($)	Energy per week (Wh)	Cost per week ($)	Weekly in DEVS units
Television	156	5	0.78	0.1	5.46	0.7	2100
Lamp	300	24	7.2	0.94	50.4	6.58	10,080
DVD Player	17	4	0.068	0.01	0.476	0.07	1680
Ceiling Fan	35	12	0.42	0.65	2.94	4.55	5040
Computer Notebook	25	24	0.6	0.08	4.2	0.56	10,080
Receiver	28	5	0.14	0.02	0.98	0.14	2100
Router	6	24	0.144	0.02	1.008	0.14	10,080
Room Heater	1320	24	31.68	4.12	221.76	28.84	10,080
stereo System	33	5	0.165	0.025	1.155	0.175	2100
Video game System	36	3	0.144	0.02	1.008	0.14	1260

Table 6 Devices in the bathroom and laundry room

Smart Home Room 4: Bathroom and Laundry							
Device Name	Wattage (W)	Hours per day	Energy per day (Wh)	Cost per day ($)	Energy per week (Wh)	Cost per week ($)	Weekly in DEVS units
Clothes Washer	255	6	1.02	0.14	7.14	0.98	2520
Water Heater	4500	12	54	7.02	378	49.14	5040
Lamp	300	12	3.6	0.47	25.2	3.29	5040
Hair Dryer	710	2	1.42	0.18	9.94	1.26	840
Clothes Dryer	2790	4	11.16	1.46	78.12	10.22	1680

6.1 Fixed settings

The first fixed-setting simulation is conducted for the kitchen room. The time of the simulation is one week (10,080 DEVS time unit). The kitchen has seven devices which are: air filter, dishwasher, deep fryer, microwave, toaster, refrigerator, and coffee maker. Figure 10 shows a snippet from the DEVS environment. The devices are connected through the *Smart Meter*. Table 3 shows the devices' schedule.

Figures 11 shows that the weekly power consumption is 99.68 KW. The total working hours of the devices in one day is 63 h. Without variable day-to-day pricing, the estimated annual cost equals to the cost of one day multiplied by 365 days (1.85 × 365 = 675.25$), and the estimated annual power consumption equals to 14.24 × 365 = 5197.6 KW.

The second fixed-settings simulation is conducted for the four rooms. The devices in the kitchen room is the same as before. Tables 4, 5, and 6 show the devices that are used in the other rooms and their corresponding data in terms of the wattage, working hours, daily cost day, and weekly cost. The simulation is performed for one week (10,080 DEVS time unit) as well.

In addition, we have modeled and used five more devices that might exist in smart home network systems. The devices are garage door opener, iron, vacuum cleaner, gas meter, and air temperature sensor. These devices are referred to as "*other devices*". Table 7 shows the devices and their corresponding data. The vacuum cleaner is used in three rooms (kitchen, living room, and bedroom) to reflect the fact that a vacuum can be moved from one room to another.

Table 7 Other devices in the smart home system

Other Devices							
Device Name	Wattage (W)	Hours per day	Energy per day (Wh)	Cost per day ($)	Energy per week (Wh)	Cost per week ($)	Weekly in DEVS units
Garage Door Opener	400	1	0.4	0.05	2.8	0.35	420
Iron	100	2	2.2	0.28	15.4	1.96	840
Vacuum	542	1	0.542	0.071	3.8	0.497	420
Gas Meter	0.095	24	0.004	0.003	0.028	0.021	10,080
Air Temperature sensor	0.095	24	0.004	0.003	0.028	0.021	10,080

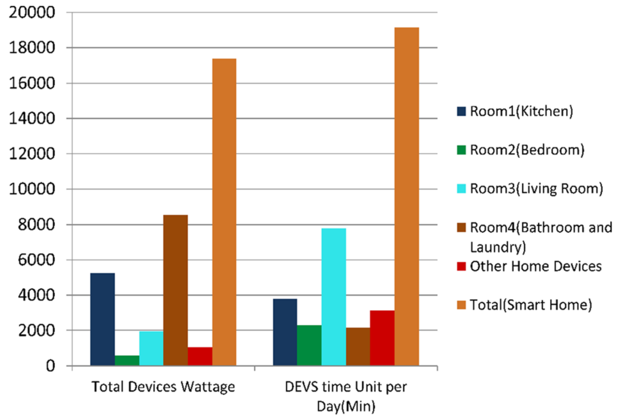

Fig. 12 Total DEVS time unit and devices wattage of four rooms in the smart home system

System modelers and designers can choose to use this feature for any device. The dynamic reconfiguration feature in DEVS environment is important for changing the system structure by adding and removing couplings while the simulation is running.

The total cost and power consumption of these devices are 2.85$ and 22.056KW per week, respectively. The yearly estimates are 148.56$ and 1150.1KW, respectively.

Figure 12 shows the amount of power that is consumed by each room in one week. The right part in the figure shows the total devices' operating time in each room (in DEVS time units). Using the fixed schedule and settings, the estimated yearly cost and power consumption of the 4-room smart home system with other devices are 6654.74$ and 49,461.9KW, respectively. As shown in the figure, the cost of the bathroom and laundry have the highest cost of 3383.6$ per year. The bedroom has the least cost. Providing this information using simulation can help users in changing their behaviors and the working hours of the devices that consume the highest power.

6.2 Variable settings

The result of this section focuses on the kitchen room while changing devices schedule and operating hours. The kitchen contains an air filter, a coffee maker, a refrigerator, and a dishwasher. The results shown are for one week. However, a user can run the DEVS environment for any time period. Table 8 presents a sample of the modifications on each device's value and their corresponding DEVS time units.

We have used 51 Poisson events to trigger changes in devices' schedule. Figure 13 shows the effect on changing working hours, power consumption, and the cost for one week. We investigated the changes that occurred for each device. For example, the air filter had working hours of 24 in the time period t = 0 to t = 264. At t = 264, the Poisson algorithm triggered a change that caused the working hours to become 11 h. Consequently, the power consumption and cost are reduced. Some of the triggers increase the power consumption and some others decrease it. At the end of the simulation, the total cost and power consumption were 14.4$ and 116.3KW, respectively. Subsection 6.1 reported that the power consumption and cost are

Table 8 Sample of the Poisson algorithm effect on the kitchen devices

Time (t)	Hours per Day				Power per Day				Total power KW at (t)	Total cost $ at (t)	Total hours at (t)
	Air Filter	Coffee Maker	Dishwasher	Refrigerator	Air Filter	Coffee Maker	Dishwasher	Refrigerator			
0–264	24	2	8	24	1.2	2	2.64	1.8	7.64	0.1471	58
264–432	11	2	8	24	0.55	2	2.64	1.8	6.495	0.147	45
432–456	11	2	16	24	0.55	2	5.28	1.8	9.63	0.129	53
456–480	11	4	16	24	0.55	4	5.28	1.8	11.63	0.176	55
480–792	11	4	16	23	0.55	4	5.28	1.73	11.7	0.176	54
792–867	13	4	16	23	0.65	4	5.28	1.73	10.08	0.212	56
867–912	13	4	16	2	0.65	4	5.28	0.15	5.8	0.2104	35
912–1060	13	4	3	2	0.65	4	1	0.15	23.8	0.2145	22
1060–1344	13	22	3	2	0.65	22	1	0.15	23.3	0.22	40
1344–1372	3	22	3	2	0.15	22	1	0.15	24.13	0.1925	30
1372–1488	3	22	3	13	0.15	22	1	0.98	20.13	0.11	41
9960–10,080	20	6	16	14	1	6	5.28	1.05	13.33	0.139	56

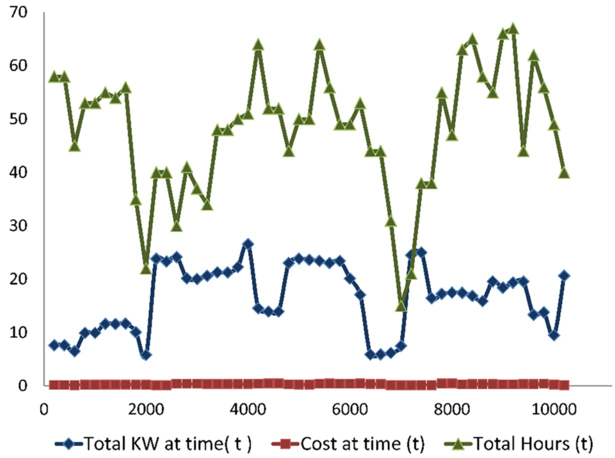

Fig. 13 Power, cost, and working hours using Poisson distribution

12.97$ and 99.6KW, respectively. Hence, the simulation results shown in Fig. 13 indicate an increase in the power and cost.

For the 4-room smart home, the Poisson parameter is set to 20 (λ = 20). The number of devices in this scenario is 13 devices, three devices in each of the three rooms the kitchen, the bedroom, and the bathroom, and four devices in the living room. The 10,080 DEVS time units which corresponds to one week. The number of Poisson events that trigger changing on the schedule is 170. Table 9 shows a sample of the working hours schedule for each device in the system. The Poisson algorithm generates a time value for the next change. When that time arrives, the corresponding device atomic model assigns a uniform random number that represent a new value for the working hours. It should not exceed the device's limitation hours (Table 2). The new value can be higher or lower than the previous value. This affects the power consumption and cost.

Figure 14 shows the working hours, power, and cost of the system. The simulation starts with a pre-defined schedule. At time t = 648, the dishwasher and the water heater change their working-hour values. At different simulation periods, the power consumption and cost are

Table 9 Sample of the working hours per day as a result of the Poisson algorithm

Time	0–648	648–700	700–827	827–892	892–987	987–1104	…	9933–10,080
Air Temperature sensor	24	24	24	24	24	16	…	23
Clothes Washer	6	6	4	5	2	6	…	3
Air Filter	24	24	24	24	12	12	…	13
Iron	1	1	1	1	1	2	…	1
PC Monitor	8	8	8	23	23	17	…	23
Deep Fryer	1	1	1	2	2	2	…	1
DVD Player	4	4	4	4	4	10	…	16
Computer	8	8	8	8	12	12	…	20
Lamp	12	12	12	12	12	22	…	4
Water Heater	24	21	21	21	21	11	…	18
Dishwasher	3	2	2	2	2	1	…	1
Refrigerator	24	24	24	24	24	24	…	24
Video game system	6	6	6	3	3	3	…	1

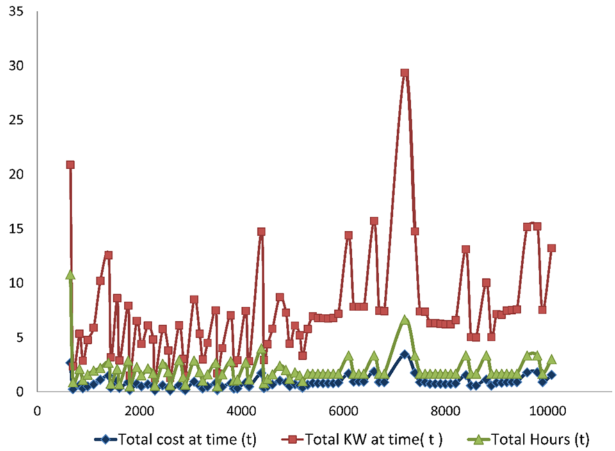

Fig. 14 The power consumption and cost for the 4-room smart home

different than the pre-defined schedule. The values can be higher or lower. Overall, the power consumption and cost for one week 593.14KW and 68.36$, respectively.

We aim to use the Poisson algorithm to trigger changes to reveal scenarios where homeowners can use to achieve better system performance that the fixed-settings scenario. Our goal is to run different simulations with different Lambda values to lower the power consumption and cost. This can provide users with insights on behaviors that are recommended and to avoid some settings that might degrade the overall performance. Also, many simulations were conducted where a schedule of only one device is changed at a time. These simulations help in identifying the best and worst-case scenarios depending on the chosen Poisson values. Table 10 shows the working hours for the smart home devices at both the best and worst-case scenarios.

The following set of simulation varies the value of the Poisson algorithm parameter Lambda (λ). The goal is to explore the behavior of the smart devices network at different values of Lambda. The values that are used for λ are: 20, 40, 50, 80, 100, and 120. The simulations are for the case of the kitchen. Figures 15, 16, and 17 show the power consumption

Table 10 Working hours for the best-case and worst-case scenarios

Devices	Best-case scenario Hours per day	Worst-case scenario Hours per day
Air Temperature sensor	24	24
Clothes Washer	6	10
Air Filter	24	24
Iron	1	2
PC Monitor	10	20
Deep Fryer	1	2
DVD Player	11	17
Computer	10	16
Lamp	24	24
Water Heater	9	24
Dishwasher	2	4
Refrigerator	24	24
Video game system	12	24

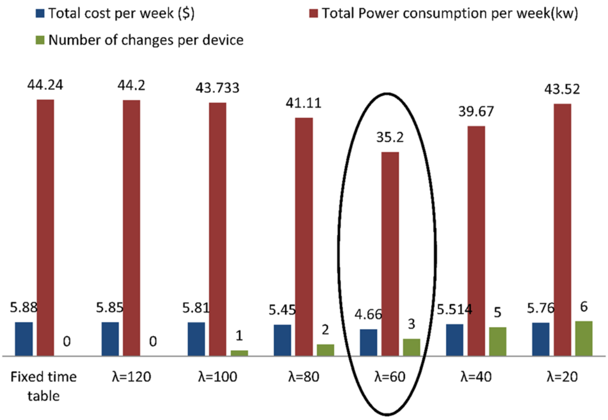

Fig. 15 Power consumption and cost for one week

and cost for the smart kitchen for one week, one month, and one year, respectively. The figures show a comparison between a fixed schedule and different Lambda values ($\lambda = 20$, 40, 60, 80, 100, and 120). The results are averaged for multiple simulation runs. When $\lambda = 120$, which is a large value, the power and cost values are very close to the fixed schedule because very small number of changes occur to the devices' working hours. The lowest value of power consumption is achieved when $\lambda = 60$. This value causes an average of 3 changes per device in a week time. Hence, users are advised to change their devices schedule accordingly.

The simulation of the kitchen room (with four devices) was conducted 1000 times in order to compute the confidence interval. For the power consumption, the standard deviation (σ) of the 1000 simulation equals 1.66496 and the mean (μ) equals 36.5262KW. Table 11 shows the confidence intervals using the t-distribution.

6.3 Summary and evaluation

Simulation results have revealed important insights and information for smart home users. The simulations showed that if devices change their schedules 3 times per week, savings in power consumption can be achieved with high confidence. Hence, users are recommended to follow the revealed behavior. Table 12 shows simulation results for the 4-room smart home. As

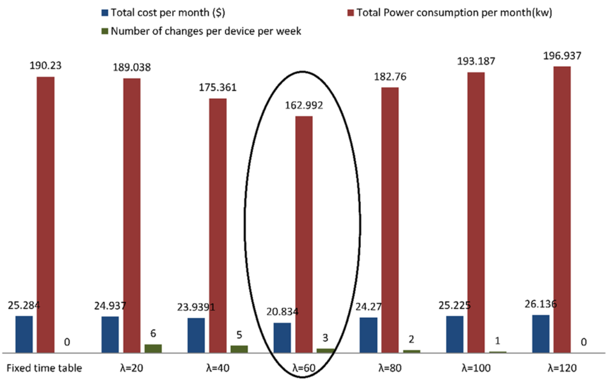

Fig. 16 Power consumption and cost for one month

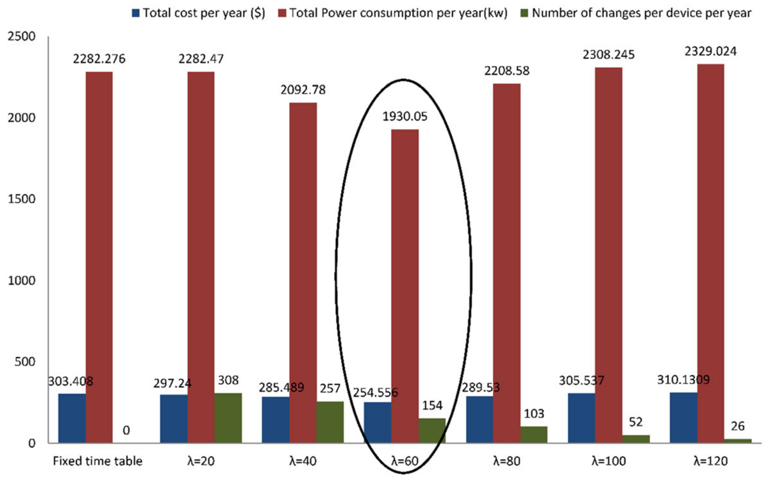

Fig. 17 Power consumption and cost for one year

shown in the table, the total amount of power and cost for one week are lower than using a fixed schedule.

The simulations provide users with a handy tool to experiment with different system parameters (for example, Lambda). Based on the simulation results, users can tailor their behaviors and system's settings to suit their needs and predict the new behavior's cost. They can then tune their devices' usage to achieve better performance. Also, they can investigate the operation of an individual device, a separate room, or an entire house.

However, the proposed approach did not model data security and confidentiality protocol that occur between smart meters and home devices [34]. Hence, the devices and appliances models should capture the behavior of authentication that occur between senders and receivers. Moreover, the developed devices' models assume that when a device is working, the amount of consumed power by the device does not change. This assumption might not be correct because some sensing devices can behave differently while monitoring a surrounding element. For example, a surveillance camera might produce more recordings and frames when the monitored environment has objects that are moving than when it is static [25]. Hence, the consumed power can be different during the working hours based on the devices' activity levels.

Table 11 Confidence Intervals using t-distribution

Confidence Interval (C.I.)	Power Consumption (KW)
95%	36.5262 ± 0.1032
98%	36.5262 ± 0.1225
99%	36.5262 ± 0.13562
99.9%	36.5262 ± 0.17325

Table 12 Power and cost values for the 4-room smart home system network

Scenario	Total power consumption (Wh)	Total Cost ($)	Simulation time
Fixed schedule	675.47kw	103.54$	One week
Modified schedule based on Poisson distribution	593.14kw	68.45$	One week

7 Conclusion and future work

Smart home network systems are considered one of the most prominent and market-appealing technologies with great potential and commercial viability. Six models were developed to capture the behavior of smart home devices which are: Identifier, Recorder, Signal controller, Smart Meter, Controller, and Devices. The system's components and devices were modeled using the Discrete Event System Specification (DEVS) formalism environment. Dynamic component-based structure is used in building the smart devices network. Different data files were generated by the devices during the simulation to provide the users with the capability for testing and evaluation. The performance of the system is evaluated in terms of the power consumption.

Extensive simulations and different scenarios were investigated in order to reveal insights and useful information for the users. The simulations compared a fixed schedule with a Poisson-based dynamic schedule to mimic human usage of devices. The Poisson algorithm is used because events are independent. The Poisson distribution generates random times that represent specific times for a schedule change. Also, a uniform random number is used to generate a new working schedule for each device. Users of the proposed system can use it to study and investigate the power consumption of a device, a room, or a house.

Regarding the future work, we aim to provide scenarios where different devices in the smart home network have different working hours based on the time of the year. For example, a device operates for longer hours in a specific season than in other seasons. Adding this feature to the devices schedule can result in a higher accuracy in evaluating the system performance. In addition, we aim to add more devices and rooms in order to simulate larger buildings. Moreover, we aim to add other variables of interest such as the comfort level of smart home occupants. In order to achieve the future work objectives, the distributed DEVS simulation framework (i.e. DEVS- Service-Oriented Architecture (SOA)) will be deployed on a distributed computing environment to divide the simulation tasks among multiple nodes. In DEVS-SOA, the different computing nodes cooperate by simulating a subset of the devices' models. Then, the nodes communicate by sending messages to other nodes such as their recorded data and controlling signals. This will add scalability to the proposed modeling and simulation approach.

References

1. Aid L, Zaoui L, Mostefaoui SAM (2016) Using DEVS for modeling and simulation of ambient objects in intelligent buildings. J Ambient Intell Humaniz Comput 7(4):579–592

2. Akhtar N, Niazi MA, Mustafa F, Hussain A (2011) A discrete event system specification (DEVS)-based model of consanguinity. J Theor Biol 285(1):103–112
3. Alansari Z, Anuar N, Kamsin A, Soomro S, Riyaz Belgaum M, Miraz D, et al (2018) Challenges of internet of things and big data integration. Emerging Technologies in Computing 47–55
4. Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M (2015) Internet of things: a survey on enabling technologies, protocols, and applications. IEEE Commun Surv Tutor 17(4):2347–2376
5. Balta-Ozkan N, Davidson R, Bicket M, Whitmarsh L (2013) The development of smart homes market in the UK. Energy 60:361–372
6. Bhati A, Hansen M, Chan CM (2017) Energy conservation through smart homes in a smart city: a lesson for Singapore households. Energy Policy 104:230–239
7. Bilal M (2017) A review of internet of things architecture, technologies and analysis smartphone-based attacks against 3D printers. arXiv:1708.04560 [cs.NI]
8. Chan M, Campo E, Estève D, Fourniols J-Y (2009) Smart homes – current features and future perspectives. Maturitas 64(2):90–97
9. Xuejun Chen (2002) Dependence management for dynamic reconfiguration of component-based distributed systems. In Proceedings 17th IEEE International Conference on Automated Software Engineering. 279–284
10. Chow ACH, Zeigler B (1994) Parallel DEVS: a parallel, hierarchical, modular modeling formalism. Winter Simulation Conference DOI: https://doi.org/10.1109/WSC.1994.717419
11. Elzas MS, Zeigler BP, Oren TI (1989) Concepts for distributed knowledge maintenance in variable structure models. In: Modelling and simulation methodology: knowledge system paradigms. 45-54
12. Ferdinando F, William Y, Enrico P (2017) A Multiagent system approach to scheduling devices in smart homes. In proceedings of the. In: 16th conference on autonomous agents and MultiAgent systems (AAMAS '17), pp 981–989
13. Firth S, Kane T, Dimitriou V, Hassan T, Fouchal F, Coleman M, Webb L (2019) REFIT smart home dataset. https://repository.lboro.ac.uk/articles/REFIT_Smart_Home_dataset/2070091/1. Accessed 20 December 2019
14. Hargreaves T, Wilson C, Hauxwell-Baldwin R (2018) Learning to live in a smart home. Build Res Inf 46(1):127–139
15. Hu X, Zeigler BP, Mittal S (2003) Dynamic reconfiguration in DEVS component-based modeling and simulation. SIMULATION
16. Hu X, Hu X, Zeigler BP, Mittal S (2005) Variable structure in DEVS component-based modeling and simulation. SIMULATION 81(2):91–102
17. Iqbal A, Ullah F, Anwar H, Kwak KS, Imran M, Jamal W, Rahman A (2018) Interoperable internet-of-things platform for smart home system using web-of-objects and cloud. Sustain Cities Soc 38:636–646
18. Jaradat M, Jarrah M, Jararweh Y, Al-Ayyoub M, Bousselham A (2014) Integration of renewable energy in demand-side management for home appliances. International renewable and sustainable energy conference (IRSEC). 571-576
19. Jaradat M, Jarrah M, Bousselham A, Jararweh Y, Al-Ayyoub M (2015) The internet of energy: smart sensor networks and big data management for smart grid. Procedia Comput Sci 56:592–597
20. Jarrah M (2016) Modeling and simulation of renewable energy sources in smart grid using DEVS formalism. Procedia Comput Sci 83:642–647
21. Jarrah M and Al-Shrida F (2017) A multi-objective evolutionary solution to improve the quality of life in smart cities. The 14th international conference on smart cities: improving quality of life using ICT & IoT (HONET-ICT). 36-39
22. Jarrah M, Jaradat M, Jararweh Y, Al-Ayyoub M, Bousselham A (2015) A hierarchical optimization model for energy data flow in smart grid power systems. Inf Syst 53:190–200
23. Kane T, Firth SK, Hassan TM, Dimitriou V (2017) Heating behaviour in English homes: an assessment of indirect calculation methods. Energy Build 148:89–105
24. Karaboghossian T, Zito M (2018) Easy knapsacks and the complexity of energy allocation problems in the smart grid. Optim Lett 12(7):1553–1568
25. Khan M, Tanveer H, Mohammad T, Giovanna S, Victor HCD. (2019) Cost-effective video summarization using deep CNN with hierarchical weighted fusion for IoT surveillance networks. J IEEE Internet of Things.
26. Knuth D (1969) The art of computer programming. Vol. 2. Addison Wesley Longman Publishing Co
27. Kortuem G, Kawsar F, Sundramoorthy V, Fitton D (2010) Smart objects as building blocks for the internet of things. IEEE Internet Comput 14(1):44–51
28. Lee E, Bahn H (2013) Electricity usage scheduling in smart building environments using smart devices. ScientificWorldJournal 2013:1–11
29. Li M, Gu W, Chen W, He Y, Wu Y, Zhang Y (2018) Smart home: architecture, technologies and systems. Procedia Comput Sci 131:393–400

30. Maatoug A, Belalem G, Mostefaoui K (2014) Modeling and simulation of energy management system for smart city with the formalism DEVS: towards reducing the energy consumption. Int J Comput Appl 90(18): 38–43
31. Mehdi G, Roshchin M (2015) Electricity consumption constraints for smart-home automation: an overview of models and applications. Energy Procedia 83:60–68
32. Mehdi L, Ouallou Y, Mohamed O, Hayar A (2018) New smart home's energy management system design and implementation for frugal smart cities. In the 2018 international conference on selected topics in Mobile and wireless networking (MoWNeT). 149–153
33. Mubdir B, Al-Hindawi A, Hadi N (2016) Design of smart home energy management system for saving energy. Eur Sci J ESJ 12(33):521
34. Nasir NH, Shabir AP, Javaid AS, Fadi A, Khan M (2019) Secure data transmission framework for confidentiality in IoTs. Ad Hoc Netw 95:101989
35. Piyare R, Tazil M (2011) Bluetooth based home automation system using cell phone. In 2011 IEEE 15th international symposium on consumer electronics (ISCE). 192–195
36. Ray AK, Roshan R (2017) Issues, challenges and application of big data in smart home. IJCA Proc Int Conf Comput Syst Math Sci 14–17
37. Tendeloo YV, Vangheluwe H (2018) Discrete event system specification modeling and simulation. Winter Simulation Conference (WSC) 162–176.
38. Theodoridis E, Mylonas G, Chatzigiannakis I (2013) Developing an IoT smart city framework. IISA. doi: https://doi.org/10.1109/IISA.2013.6623710
39. Ullman JD (1975) NP-complete scheduling problems. J Comput Syst Sci 10(3):384–393
40. Van Woensel L, Archer G, Panades-Estruch L, Vrscaj D (2019) Ten technologies which could change our lives: potential impacts and policy implications. http://www.europarl.europa.eu/EPRS/EPRS_IDAN_527417_ten_trends_to_change_your_life.pdf. Accessed 20 December 2019
41. Wainer GA, Goldstein R, Khan A (2018) Introduction to the discrete event system specification formalism and its application for modeling and simulating cyper-physical systems. Winter Simulation Conference (WSC) 177–191
42. Wilson C, Hargreaves T, Hauxwell-Baldwin R (2017) Benefits and risks of smart home technologies. Energy Policy 103:72–83
43. Zeigler BP (2003) DEVS today: recent advances in discrete event-based information technology. In the11th IEEE/ACM international symposium on modeling, analysis and simulation of computer telecommunications systems, MASCOTS. 148–161
44. Zeigler BP, Louri A (1993) A simulation environment for intelligent machine architectures. J parallel Distrib. Comput 18(1):77–88
45. Zeigler BP, Reynolds RG (1985) Towards a theory of adaptive computer architectures. In the 5th international conference on distributed computing systems. Computer Society Press 19:468–475

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.