Frontiers of Computer Science CN 10-1014/TP
https://doi.org/10.1007/s11704-021-0375-z ISSN 2095-2228
RESEARCH ARTICLE

A location-based fog computing optimization of energy management in smart buildings: DEVS modeling and design of connected objects

Abdelfettah MAATOUG[1,2], Ghalem BELALEM[1], Saïd MAHMOUDI[3]

1 Computer Science Department, Faculty of Exact and Applied Sciences, University Oran 1 Ahmed Ben Bella, Oran, Algeria
2 Science and Technology Department, Faculty of Science and Technology, University of TIARET, Algeria
3 Computer Science Department, Faculty of Engineering, University of MONS, Belgium

Front. Comput. Sci., **Just Accepted Manuscript** • 10.1007/s11704-021-0375-z
http://journal.hep.com.cn on February 22, 2021

Front. Comput. Sci.
https://doi.org/10.1007/s11704-021-0375-z

RESEARCH ARTICLE

A location-based fog computing optimization of energy management in smart buildings: DEVS modeling and design of connected objects

Abdelfettah MAATOUG (✉) [1,2], Ghalem BELALEM[1], Saïd MAHMOUDI[3]

1 Computer Science Department, Faculty of Exact and Applied Sciences, University Oran 1 Ahmed Ben Bella, Oran, Algeria
2 Science and Technology Department, Faculty of Science and Technology, University of TIARET, Algeria
3 Computer Science Department, Faculty of Engineering, University of MONS, Belgium

Abstract Nowadays, smart buildings rely on Internet of things (IoT) technology derived from the cloud and fog computing paradigms to coordinate and collaborate between connected objects. Fog is characterized by low latency with a wider spread and geographically distributed nodes to support mobility, real-time interaction, and location-based services. To provide optimum quality of user life in modern buildings, we rely on a holistic Framework, designed in a way that decreases latency and improves energy saving and services efficiency with different capabilities. Discrete EVent system Specification (DEVS) is a formalism used to describe simulation models in a modular way. In this work, the sub-models of connected objects in the building are accurately and independently designed, and after installing them together, we easily get an integrated model which is subject to the fog computing Framework. Simulation results show that this new approach significantly, improves energy efficiency of buildings and reduces latency. Additionally, with DEVS, we can easily add or remove sub-models to or from the overall model, allowing us to continually improve our designs.

Keywords smart building, energy consumption, IoT, fog computing Framework, DEVS simulation models

1 Introduction

The impact of the Internet of Things (IoT) on the evolution towards next-generation smart environments such as smart homes, smart buildings and cities will largely depend on the effective integration of IoT and fog computing technologies [1,2,3]. Buildings are complex systems and many factors can affect the total energy consumption in different buildings. As noted in [4], Building Energy Management System (BEMS) is one of the available solutions used to reduce building energy consumption. BEMS is a distributed control system within the buildings which monitors and manages lighting, elevator, air conditioning, heating, drainage, and other energy usage [5]. The deployment of BEMS in buildings requires a large number of sensors and actuators on various sites, especially if the control of energy consuming equipment in buildings comes

Received Monthdd, yyyy; accepted Monthdd, yyyy

E-mail: a_maatoug@esi.dz

from their local networks. The goal is to separate the operation of the entire process of this control of the Cloud, and for that, it is necessary to rely on Fog servers which must be closer to the data collection points [6].

This field of research is involved in developing new methodologies that aim to facilitate the integration, deployment and interoperability of sensors and actuators within a smart building's wireless network. In this work, to optimize energy management in smart buildings, we rely on a location-based fog computing Framework. The design of this Framework is presented in [7], and this work is a continuation of it. More precisely, the goal of this work is to model connected objects at a high level of abstraction after defining an environment that allows simulation of the wireless sensor network operating the smart building. Modeling and simulation enable us to simulate different scenarios in order to analyze and assess various parameters and behaviors. Thus, instead of combining real costly functional devices, our designer configures and simulates devices using a modeling and simulation tool. In this project, for the first time, an IoT Fog-based approach has been considered to design, by DEVS modeling, the energy management system used to schedule the energy production and consumption of smarts devices installed in smart buildings. In addition, energy management in smart buildings has been enriched by taking advantage of the location of users to rationalize their energy use.

In order to streamline and simplify this research project and make it possible, we have adopted the following plan, going through three phases which represent the main contributions of this paper:

1) Adopting a Framework for controlling energy consumption within smart buildings in order to implement it in its appropriate environment: Energy consumption is controlled by an intelligent Framework created in [7], which uses a fog computing paradigm in locating users to rationalize the energy consumption in smart buildings. We have also explained the right environment to run this Framework.

2) Design the system that corresponds to the Framework: Following the operation of the above Framework, an IoT-based system was designed as a model to realize the strategies and scenarios included in this Framework according to the multilevel architecture suggested in [4]. In this work, we have proposed to add new components to this architecture, which are in line with technologies used to adjust energy consumption based on user location information.

3) DEVS modeling and simulation: We modeled according to the DEVS formalism, everything that would exist in the environment that matches the aforementioned Framework. Apart from that, through duplicate simulations in different scenarios, we evaluate the performance of the system by running different experiments and applying the necessary changes to improve the efficiency of our contributions.

Modifications to improve performance are possible with DEVS modeling because DEVS divides the global system into subsystems, giving us the freedom to change by removing or adding subsystems to each component system [8], which is what motivates us to choose it. It is also noted here that this decomposition characteristic simplifies the implementation of the systems, whatever their complexity.

This paper is organized as follows. Section 2 reviews the related work. Section 3 presents the building energy management Framework and the DEVS formalism used for modeling. In Section 4, we explain the application scenario that defines the physical and behavioral aspects of an appropriate environment to describe the proposed solutions. In Section 5, we present a layered architecture that conforms to the Framework and DEVS modeling of the resulting systems. Section 6 provides an implementation of the overall model under the JDEVS simulator and discusses the simulation results. We conclude the paper through Section 7. The paper’s logical organization is visualized in Figure 1.

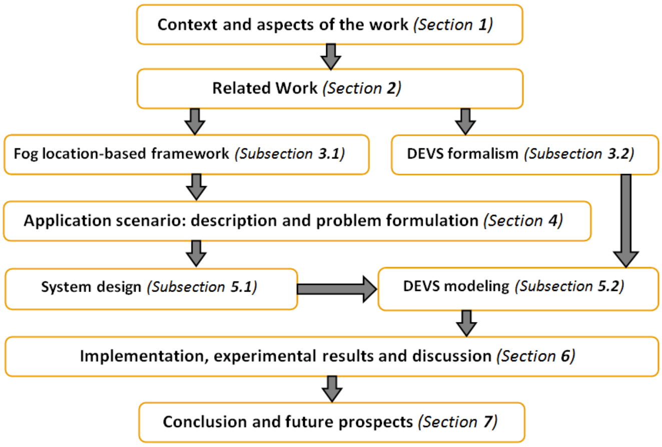

Context and aspects of the work (Section 1)
Related Work (Section 2)
Fog location-based framework (Subsection 3.1)
DEVS formalism (Subsection 3.2)
Application scenario: description and problem formulation (Section 4)
System design (Subsection 5.1)
DEVS modeling (Subsection 5.2)
Implementation, experimental results and discussion (Section 6)
Conclusion and future prospects (Section 7)

Fig. 1 The logical roadmap of this work

2 Related works

Smart building topics, especially energy management, have become a concern for a large number of researchers today. Some of them have proposed modern Frameworks using Cloud-based IoT technology. Others went directly to the design of energy management systems in buildings with modern and intelligent technologies **[9]**. Most of them used the same technologies but with different and distinct offerings. In addition, computing applications based on user location have emerged strongly in saving energy in smart buildings.

The authors of [10] proposed a holistic framework that integrates smart home objects into a Cloud-based solution. They identified a multi-tiered model that aligns with the proposed Framework but no experience has been done for this model. In [11], the authors present a Framework that uses all the computing power as well as the control and monitoring capacities of the Cloud. A facility that aims to be a smart building solution integrated with an IoT platform is delivered. The authors in [12] discuss the design of Cloud-connected system with backend analytic process that performs analysis on energy usage pattern and the wastage. They present the experimental results of real-time monitoring of consumers' energy usage in a commercial space in order to identify the electricity wastage. Most of these solutions are non-exhaustive and are designed to solve partial problems. Furthermore, existing problems in cloud computing are almost all supported in the new fog computing paradigm that is not mentioned in these works.

In this last range, authors of [13] present fog computing as a new platform for home energy management. This platform allows the user to implement energy management with control-as-services, while minimizing the cost of implementation and time to market. The authors implemented and experimented with two prototypes: Home and Micro Grid. However, shifting the processing of high-power central processing nodes to smaller devices on the edge of the network requires careful description and needs to be well detailed. Comparably, the authors of [14] describe how the fog computing paradigm can be used to extend the capability of a data center to take charge of energy management within the building. Further, the work suggested in [15] presents a Fog Computing smart city Architecture Network (FOCAN), a multitier structure in which applications running on things jointly compute, route, and communicate through the smart city environment. The authors have proposed a model to implement their Framework, but it is difficult to know the effectiveness of its implementation on real.

IoT environment offers new opportunities for proactively analyzing human mobility patterns and predicting user's future visit **[16]**. In this context, many researchers have been interested in studying ubiquitous mobile Internet and Smartphone computing applications. The authors of [17] proposed an intelligent Framework that controls the energy of a building depending on the location of its user. In their work, they relied on Smartphone platforms and cloud computing technologies, including the energy proportionality of the building and the user. Authors of [18] represent a user-centric intelligent solution that contributes to the energy sustainability of modern cities. It is a building management platform that has been deployed in a real building, in which a series of tests has been conducted to assess various concerns related to the management of the building infrastructure. The idea is to rationalize the use of energy depending on the occupation of the user. Also, the authors in [19] used the fog computing paradigm in their work where they proposed a system architecture to optimize energy use depending on the user's location information. Fog computing is used to respond to service requests on location.

Otherwise, the authors in [20] modeled and simulated, in DEVS, a smart home network system consisting of four rooms with different types of devices. The system shows to its users the power consumption and the cost under different scenarios. Hence, users can control and change their home devices settings and their behaviors in order to save energy and reduce cost. However, it would have been better if DEVS formalism were used to represent the proposed system in order to prove its effectiveness. Indeed, the authors incorporated it into the work of their system itself, which made it lose its primary role in improving performance.

In summary, the references mentioned above reveal the critical importance of using IoT technology in these solutions, recognizing that energy management in smart buildings, must be specifically extended to the IoT based on the new paradigm of fog computing. Currently, all these great efforts have not been able to reach all what these new technologies can achieve in smart buildings. This is due to the lack of modeling and simulation that must be used in research to verify the validity of what has been proposed and experiment with the aim of improving it. Moreover, design efforts must be based on formal and dynamic models to support simulation and formal verification [21].

3 Backgrounds

In this section, we briefly present the building energy management Framework used in our work environment and the DEVS formalism used in the design and modeling of the resulting systems.

3.1 Fog location-based Framework

In the work [7], a comprehensive Framework for automatic energy control of smart buildings was introduced based on the location of the users. This Framework has been systematically designed on the basis of the following main components (See Figure 2):

- Wireless Sensor Network for Building Automation System (WSN-BAS);
- Fog Computing Platform;
- Service Oriented Middleware (SOM);
- And GPS[1] location of user.

This Fog location based Framework considers two types of WSN-BAS: House WSN-BAS, and Office WSN-BAS. These two installations have the same role: control the equipment that consumes energy in these two buildings from its local WSNs away from the Cloud, i.e. by separating the entire process of this control from the Cloud respecting the doctrine of the new fog computing paradigm.

The fog computing platform consists of the following units: WSN-BAS fog node, sensor node, local user and the Cloud [22]. The WSN-BAS fog node contains platform services or platform infrastructure components. The essential services that the platform must provide to cover its task of managing energy in smart buildings are: Authentication access, HVAC/lighting control, Resource management, and User-Location Services (ULSs). The platform infrastructure components (the supports, the essential basis of the platform, its maintenance or its operation) are: Virtual Machine (VM) Scheduling, Control by Smartphone, and WSN-BAS Management.

To facilitate the development of applications and services provided by the fog computing platform, the Framework uses Service-Oriented Middleware (SOM).

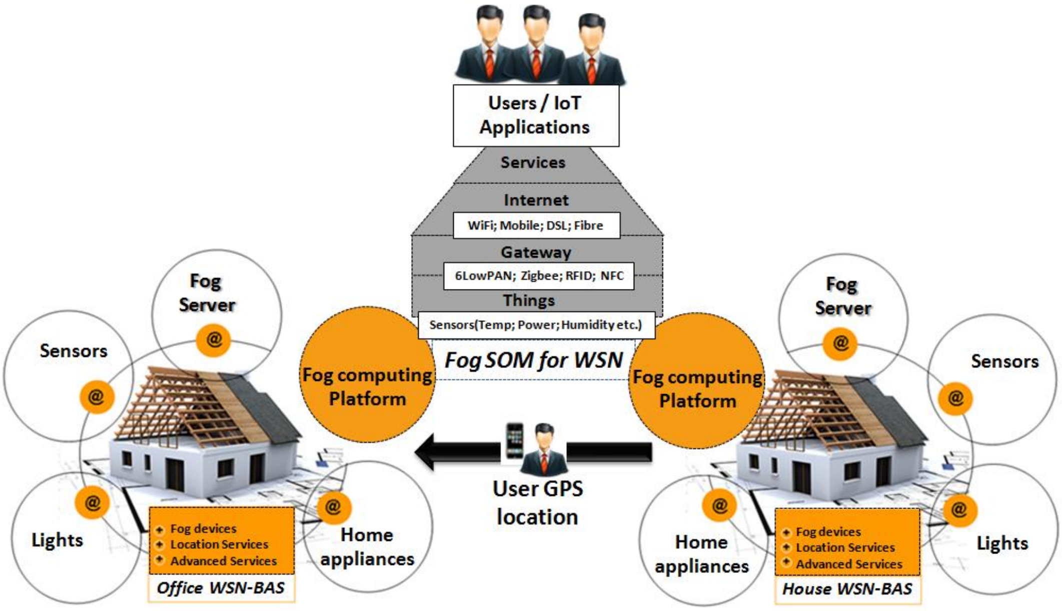


Fig. 2 Fog location-based Framework [7]

[1] GPS: Global Positioning System

This SOM comprises three layers: *Clustering layer* that gathers the data detected at each fog node, and manages the lifecycle of the data sources in the WSN-BAS, *Resources managing layer* that filters collected data to describe information needed to build services for optimizing energy management in the building, and *Producing layer*, which provides a unified and homogeneous view for the standardization of the filtered data, and the resulting data is presented as a construct of services that can be used directly as an IoT application.

GPS location, one of the services provided by the fog computing platform can indicate the communication distance between the user and the building (house or office) [19]. Now, to ensure that the user is inside or outside the building, it is necessary to test if this distance exceeds a well-defined threshold, and depending on the location of the user, we can rationalize the use of energy in both buildings by adjusting their energy consumption policy according to the user' location.

3.2 The DEVS formalism

Discrete Event System Specification (DEVS) is a popular formalism for modeling complex dynamic systems using a discrete-event abstraction [23]. This formalism is based on the theory of systems; it allows a formal representation of models susceptible to achieve mathematical manipulations comparable to the differential equations for continuous systems. It is an abstract universal formalism independent of the implementation.

DEVS formalism is based on the definition of two types of models: *atomic models* and *coupled models*.

The atomic models are the basic components; they describe the basic behavior of the system.

Formally, an Atomic Model (AM) is specified by the following structure [24]:

$$AM = (X, S, Y, \delta_{ext}, \delta_{int}, \lambda, ta)$$

Where:

- X is the set of input events;
- S is a set of states;
- δ_{int}: $S \rightarrow S$ is the internal transition function caused by the occurrence of internal events;
- Y is the set of output events;

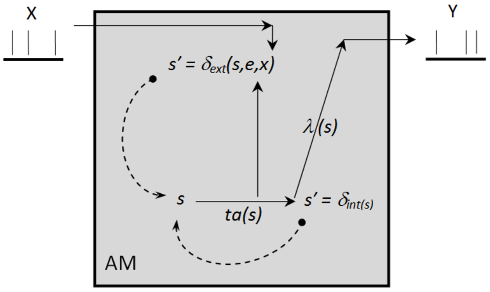


Fig. 3 Description of a DEVS atomic model

- δ_{ext}: $Q \times S \rightarrow S$ is the external transition function caused by the occurrence of external events, where:
  - $Q= \{(s,e)/s \in S, 0 \leq e \leq t_a(s)\}$ is the total state set;
  - e is the time elapsed since last transition;
- λ**:** $S \rightarrow Y$ is the output function;
- ta**:** $S \rightarrow R^{+}_{0,\infty}$ is the *time advance* function.

The interpretation of these elements is illustrated in Figure 3.

The input event set, X, defines all possible inputs that can occur on the input ports; the output set, Y, *includes* all possible outputs that the atomic model can send.

External inputs received on the input ports invoke a model's external transition function δ_{ext}, which determines how the model changes its state based on the inputs and model's current state $s' = \delta_{ext}(s,e,x)$. The model remains in a state for an amount of time that is determined by the time advance function $ta(s)$. When this time has expired, the output function λ is invoked, which sends output value λ(s) on the output ports. Following the invocation of the output function, the internal transition function δ_{int}, is invoked to transit the model to a new state $s' = \delta_{int(s)}$.

In this article, we use the graphical notation of [23], as shown in Figure 4. A state is represented by a circle containing the name of the state, the operations on the variables and the life time of the state *(ta)*. The model is in its initial state *"Si"*.

Figure 4a represents an external transition. An input event is specified by using *'?'*. The continuous line

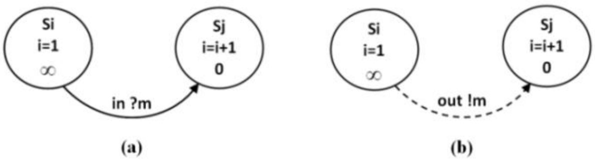


Fig. 4 Graphical notations of an external transition (a), and an internal (b).

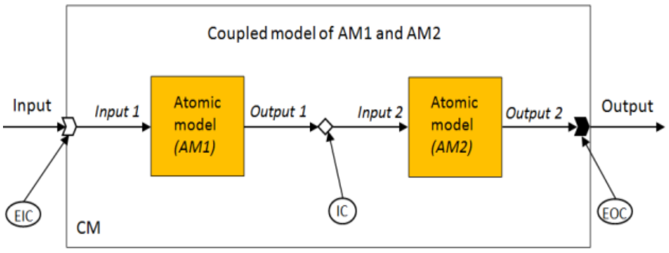


Fig. 5 Description of a DEVS coupled model

represents a state transition specified by the external transition function. When the model receives event *"m"* on its port *"in"* it goes into state *"Sj"*. Figure 4b denotes an internal transition. An output event is specified using *‘!’*. The dotted line represents a state transition specified by the internal transition function. The model evolves in state *"Sj"*, emitting event *"m"*.

To more describe complex systems, Atomic models may be coupled in the DEVS formalism to form what is called a coupled model.

Coupled models are defined by a set of sub-atomic models and/or coupled models to represent the internal structure of the system through coupling between models. It has the following three coupling relationships (see Figure 5):

1) An Internal coupling relation *(IC)*, for the coupling between the sub model ports constituting the coupled model.

2) An External Inputs Coupling relation *(EIC)*, for the coupling between the input ports of the coupled model and the input ports of the sub-models.

3) An External Outputs Coupling relation *(EOC)*, for the coupling between the output ports of the coupled model and output ports of the sub-models.

Formally, a Coupled Model (CM) is specified by a structure [24]:

$$CM = (X, Y, D, \{M_d \ / \ d \ \in D\}, EIC, EOC, IC)$$

Where:

- X is a set of possible inputs of the coupled model;
- Y is a set of possible outputs of the coupled model;
- D is the set of names associated to the components of the coupled model;
- M_d is the set of the coupled model components;
- EIC is the set of External Input Coupling;
- EOC is the set of External Output Coupling;
- IC is the set of Internal Couplings.

4 Application scenario

Applications achieved in this scenario are facilitated by wireless sensors deployed in the building's atmosphere to measure temperature, humidity, or different gas levels. In this case, the WSN-BAS ensures the exchange of information between all connected objects including sensors of a stage, and their readings can be combined to form reliable measurements. Sensors and actuators will use distributed decision-making and operation in coordination with indoor Fog servers to quickly interact with data. The WSN-BAS maximizes its utility by adjusting its operational target according to the system status and the user location. Two scheduling algorithms for the optimization of energy consumption in buildings by determining the location of the building user have been designed, namely User Location Algorithm and Energy Saving Plan Algorithm. Each algorithm serves a distinguished task and the WSN-BAS switches from one scheduling strategy to another depending on the distributed decision of the fog server. By applying Fog computing in this scenario, smart buildings can maintain their internal structure and environments to save energy and even other resources [25].

4.1 Scenario description

We now present the scenario that summarizes all the notable scenes of our environment. This scenario consists of two buildings, house and office and the user who occupies them daily. Each building has its own policies and requirements to consider when controlling energy consumption. This control is performed using a system based on local WSN (WSN-BAS), i.e. away from the Cloud. The intermediary Fog layer consists of geographically dispersed Fog servers that are deployed in local buildings for mobile users, i.e. in their houses or offices.

By providing local services by the Fog Computing Platform installed in the Fog Server which is sent to submit to deployment sites, the Fog server installed inside the building can quickly pre-load localized content. Through this too, mobile users can enjoy high-speed local connections without having to search

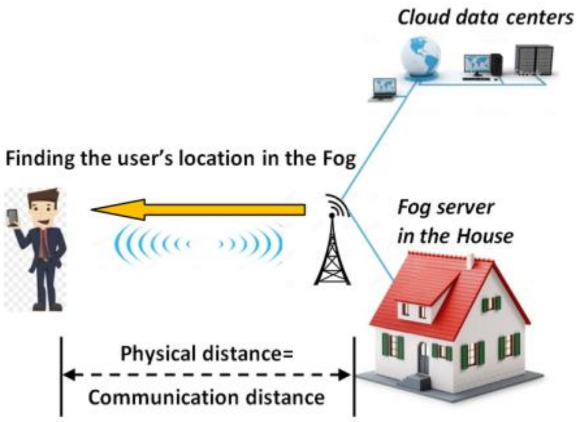


Fig. 6 How to locate the user in our approach

the Cloud. The goal of this method is that once the user leaves the building, the building management system switches directly into vacation mode, and if not, the system will maintain a satisfactory comfort; which aims a rational use of energy.

To follow our environment scenario, we have adopted three types of policy for the management of energy consumption, one for the case of a user installed in his house, one for the case he is in his office and one for the case that he is outside. The idea is to always locate his Smartphone and with the fastest speed. The support of his Smartphone comes out of the house and moves to the office. Location changes trigger servers to automatically adjust energy saving policies for both buildings. The basic function is to allow the server to detect changes of user's location and trigger changes to the energy policy by turning on/off the electrical devices in those buildings.

As shown in Figure 6, with the Fog server located in the house, it will be easy and fast to find the distance between the user and the house or the office just with a simple communication with the user. The physical distance will be the distance of this communication. The Fog server can indicate the communication distance using the GPS location service [26,27], and based on this distance, the user's location is determined for both buildings, whether home or office.

4.2 Problem formulation

In the following, we have created the general algorithms related to the automatic adjustment of energy consumption in the two buildings house and office, by locating the user and of course after a correct authentication to the system. The correct authentication is the guarantee that our Framework is independent in all stages of its operation, individually and secluded [28], as it is never allowed to log in twice before logging out. Therefore, the first algorithm will be the correct authentication of the system in order to access the energy consumption adjustment application, the second algorithm to locate the user and the third algorithm to modify the energy saving policy at house and office according to the user's location.

Algorithm 1 includes procedures for authentication and authorization of access to the data stored in the information system. The system user authentication is realized on the basis of the user name and password entered. This algorithm is characterized in that in case of unsuccessful authorization of a user, the user session is closed. Meanwhile, the user access is blocked and the user is notified accordingly. There are other types of algorithms allowing users to be authenticated efficiently while they are in specific locations in an environment and will be rejected if they change their location [29,30], but in this work we are satisfied with normal authentication.

Algorithm 1 Access to the Location Service (LS)

```
1 Login to the system: A user name and a password are required;
2 Checking for a blocked access: if the check is successful then the system goes to 6 else the system goes to 3;
3 If (Authentication = false) then the system performs the authentication procedure: if successful authentication then the system goes to 4 else the system goes to 6;
4 Set Authentication ← true; an access to the LS resources procedure is executed;
5 To exit from the system: the system goes to 8 or else the system goes to 3;
6 An unsuccessful attempt for logging-on to the LS is recorded: the system goes to 7;
7 The access to the LS is denied: An error message is output; The system goes to 8;
8 Set Authentication ← false; the session is closed.
```

Regarding the location of the user, the process takes place only after proper authentication to access the location services (see Algorithm 2). These services can be obtained from the Fog and if not available, they will be obtained from the Cloud. After accessing the user's location, it is presented to the party of the system that requests its use in its procedures.

Algorithm 2 User Location (UL)

```
/*Initialization                                  */
Input: Acces_Authorized, Authentication
OutPut: LS, UL
while  Access_Authorized =  false do
    if Authentication then
        Access_Authorized ⟵ true;
    else
        Access_Authorized ⟵ false;
    if Acces_Authorized then              /*LS query */
        if LS is not available then     /*cloud query */
            obtain LS from Cloud data centers;
        else                              /*Fog query*/
            obtain LS from Fog server;
    if LS is available then
        /* returns  the user's location*/
        return UL;
```

Algorithm 3 modifies the power saving mode in buildings, house and office, according to user requirements, the amount of available energy and metrological predictions. This can be done by comparing the distance between the user and each of the two buildings and after regulating energy use modes according to meteorological predictions. If the distance between the user and the house is less than a predetermined distance and the power is available, this puts the power consumption mode plenty for the house and economic for the office and the same if the user is near his office. If the user is outside the scope of both buildings, their status will certainly be the economic mode. Assuming that the user cannot occupy the house or office for more than 8 hours, automatic adjustment of energy consumption policies using these algorithms will turn off some home appliances for 16 hours in the buildings, in the house or office. In this way, energy use in buildings will be reduced very rationally. We were also able, in the case of a large number of buildings, to determine the location of the user if he was inside or outside any one of them, but in this work we were satisfied with two buildings, depending on what the application scenario imposes on us.

Moreover, speed in implementation is the only guarantor to reduce errors resulting from time interference during the execution of procedures, and this feature is provided by applying the fog computing paradigm when locating the user [31].

Algorithm 3 Energy saving plan

```
Input: UL, D, $Min_{exist\_pw}$
OutPut: $Plan_{House}$, $Plan_{Office}$
while UL is available do
    compute $House_{cons\_pw}$, $Office_{cons\_pw}$;
    regulate House_mode, Office_mode;
    compute $D_{User-House}$,$D_{User-Office}$;
    if ($D_{User-House} \leq D$) and ($House_{cons\_pw} \geq Min_{exist\_pw}$)
    then                /* user is in the house */
        House_mode ⟵ Plenty; $Plan_{House} = 1$;
        Office_mode ⟵ Economic; $Plan_{Office} = 0$;
    else
        if ($D_{User-Office} \leq D$) and ($Office_{cons\_pw} \geq Min_{exist\_pw}$)
        then            /* user is in the Office*/
            Office_mode ⟵ Plenty; $Plan_{Office} = 1$;
            House_mode ⟵ Economic; $Plan_{House} = 0$;
        else    /* user is out                    */
            House_mode ⟵ Economic; $Plan_{House} = 0$;
            Office_mode ⟵ Economic; $Plan_{Office} = 0$;
return $Plan_{House}$, $Plan_{Office}$;
```

5 System design and DEVS modeling

The simulation model was built according to the proposed scenario for the application of the Fog-based location Framework suggested in the work [7] for automated control of energy in smart buildings.

Considering the application of this scenario to the environment of two buildings, house and office, and their user, the coupled DEVS model of the resulting systems will be composed of five main sub models, each gather a set of objects (atomic models) and corresponds to a main actor in the scenario (house, office, house EMS, Office EMS and user location). In other words, each sub-model delimits a couplet model bringing

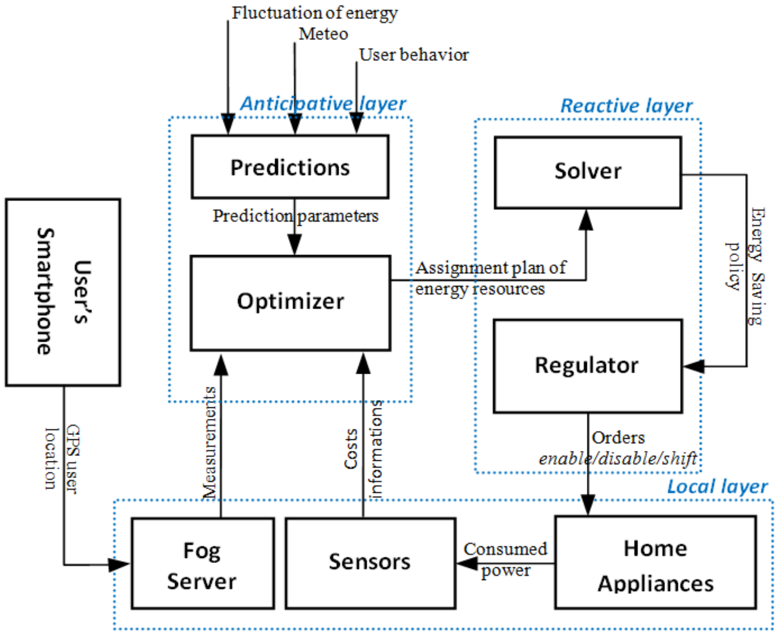


Fig. 7 Multilevel model for the building EMS (House or Office EMS)

together a set of atomic models contributing in a role of main actor in the application scenario. The following section describes the structure of each sub model and its expected behavior in the application scenario.

5.1 System architecture

We have designed the model below (see Figure 7) for energy management in smart building respecting the multilevel architecture which has been described in the work presented by [4]. We explain it again but briefly and focus on new modifications in it and additions. This structure carries several levels of modeling that correspond to several levels of command to divide the optimization problem into sub-problems in order to be more reactive to uncertainties. The information exchanged between these layers is either measurements, consigns, instructions or emergency messages. These different management levels correspond to hierarchical resolution algorithms according to different time scales. An anticipative solution is calculated at the highest level. This is ensured by what is called the optimizer which takes care of preparing in advance a plan for consumption and production of energy for a longer time horizon; this means a longer period for sampling the initially uncertain predictions related to weather and user behavior regarding movement and power consumption, which can be obtained via his Smartphone. In order to be adopted with the scale of the reactive layer, the solution obtained is then refined in a lower level of reactive management that fits the solution already calculated taking into account more detailed information for a smaller sampling period. This additional information are the actual energy consumption measurements of home appliances which has been automatically captured by sensors, and the user's location calculated by the Fog server.

The solver helps the reactive layer to carry out the assignment plan of energy resources taking into account energy constraints, and user comfort in real time. Its operation complements that of the anticipative layer which sends it the specified instructions and this latter adjusts them to the actual execution conditions according to the user's location. Even if the energy resource is unavailable or limited, a regulator will intervene by deactivating the consumption of certain equipment and by balancing energy consumption and production. This will choose the energy consumption policy before the reactive layer transfers the instructions to the local layer.

The latter solution should converge on enabling, disabling or shifting instructions for the function of building home appliances for a local adjustment of the building's energy consumption policy. In this work, our goal is not only to model these control layers but also to validate their operation and the entire energy management mechanism; of course this will be after designing a simulation model. The sampling period for these layers is approximately one hour for the anticipative layer, one minute for the reactive layer and in real time for the local layer.

5.2 DEVS modeling

In the following, the coupled models describe the composition of the objects that contribute to the optimization of energy consumption in the building. In our proposed environment, we note that they are identical in both buildings (house or office). The difference lies only in the home appliances inside these two buildings. Also, we have to notice that DEVS environment is a modular tool [4], where a user can extend the number of any of the components that are presented here, such as buildings or their contents. Therefore, the user can easily add and/or delete components to model and simulate different structures.

The model we propose divides each BEMS (House EMS or Office) into eight subsystems: *Predictions*, *Optimizer*, *Solver*, *Regulator*, *Fog Server*, *Sensor*s, *Home appliance*, and *User's Smartphone*.

We give to a BEMS the following structure of a DEVS coupled model:

$BEMS = (X, Y, D, \{M_d \;/\; d \in D\}, EIC, EOC, IC)$

Where:

- $X = \{(p, v) / p \in IPorts, v \in X_p\}$ are the inputs values;
- $Y = \{(p, v) / p \in OPorts, v \in Y_p\}$ are the outputs values;
- D is the set of names of the atomic or coupled models that compose the BEMS:
- *D = { User's Smartphone, Predictions, Optimizer, Solver, Regulator, Home appliance, Sensor, Fog Server };*
- M_D is the set of atomic or coupled the models that compose the BEMS:
- $M_{User's\ Smartphone}$ is the model of User's Smartphone subsystem;
- $M_{Predictions}$ is the model of Predictions subsystem;
- $M_{Optimizer}$ is the model of Optimizer subsystem, and so on for other subsystems;
- $EIC = \{((BEMS, a), (d, b)) / a \in IPorts_s, b \in IPorts_d\}$
- $EOC = \{((d, b), (BEMS, a)) / a \in OPorts_{BEMS}, b \in OPorts_d\}$
- $IC = \{((i, a), (j, b)) / i, j \in D, i \neq j, a \in OPorts_i, b \in IPorts_j\}$

EIC defines ports of EMS model that are connected to ports of component models that receive external events; *EOC* defines ports of EMS model that are connected to ports of component models that emit events and *IC* are the internal coupling between component models (Figure 13 in the next section clearly shows all these variables).

Due to the large number of atomic models in the global system and the similarity between the method for describing atomic models, in this paper, we will suffice to describe only one model from each system layer. Specifically, we will describe the lighting model for the local layer, the optimizer model for the anticipative layer and the solver for the reactive layer.

Let's take the example of a simple lamp to describe lighting model for both building: house or office. It is an element of the local layer. It is one of the existing home appliances in the building to provide lighting in the building. It can only be switched on or off. Figure 8 presents the DEVS atomic model of a simple lamp with four states: Off, On-try, On, and Off-try.

The DEVS formal description of the 'simple lamp' model is: $Lamp = (X, S, Y, \delta_{ext}, \delta_{int}, \lambda, ta)$

Where:

- *X = {0, 1}*, represents the input values;
- *Y = {ON, OFF}*, represents the output values;
- State variables: *S = {(phase, ta)}*, where:
  - Phase = {On, On-try, Off-try, Off};

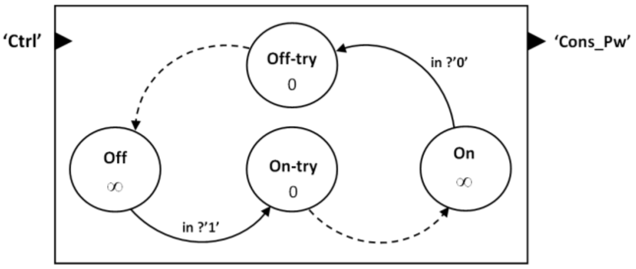


Fig. 8 DEVS atomic model of a simple lamp

- $\delta_{ext}(OFF,\ In?‘1’) = (On\text{-}try,\ 0)$, The external event in?‘1’ switch the lighting on;
- $\delta_{ext}(ON,\ In?‘0’) = (Off\text{-}try,\ 0)$, The external event in?‘0’switch the lighting off;
- $\delta_{int}\ (On\text{-}try) = (On)$, $\delta_{int}\ (Off\text{-}try) = (Off)$;
- $\lambda\ (On\text{-}try) = ON$, $\lambda\ (Off\text{-}try) = OFF$;
- $ta\ (On) = \infty$, $ta\ (Off) = \infty$, $ta\ (On\text{-}try) = 0$, $ta\ (Off\text{-}try) = 0$.

Now we go to the anticipative layer, the optimizer that provides the master plan for optimizing energy consumption by implementing procedures that represent a mathematical model to solve the general problem of user localization. In another meaning, the optimizer gives an energy resources allocation plan, depending on the user's location, to the reactive layer for actual implementation.

Figure 9 presents the DEVS atomic model of the optimizer with seven states: In_{House}, $\text{In}_{\text{Office}}$, $\text{Out}_{\text{Buildings}}$, House_To_Out, Office_To_Out, Towards_House and Towards_Office .

There are two inputs for the optimizer: ‘UL_{House}’ and ‘UL_{Office}’. The variable UL_{House} (the user's location in relation to the house) takes two values, ‘0’ or ‘1’; the value ‘0’ means going to the house, and the value ‘1’ means going to the outside. The same goes for the $\text{UL}_{\text{Office}}$ variable. This model can be changed to the state of ‘In_{House}’, ‘$\text{In}_{\text{Office}}$’, ‘$\text{Out}_{\text{Building}}$’.

The formal description of the ‘optimizer’ model in DEVS is: $Optimizer = (X, S, Y, \delta_{ext}, \delta_{int}, \lambda, ta)$

Where:

- $X = \{UL_{house},\ UL_{office}\}$ present the input event variables, where:
  - $\text{UL}_{\text{house}} = \{0,1\}$ This port indicates the user's location relative to the house:

  The external event ULHouse?‘0’ indicates that the user is heading house; the external event UL_{House}?‘1’ indicates that the user is leaving house.

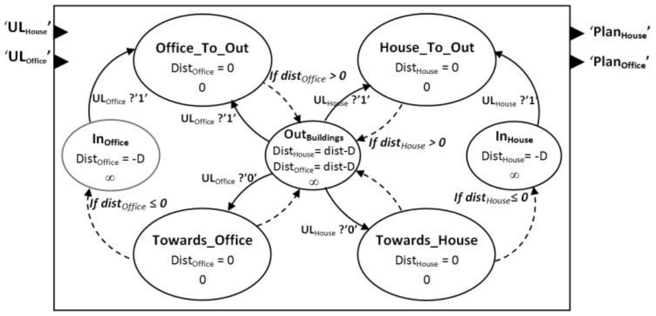

‘UL_{House}’
‘$\text{UL}_{\text{Office}}$’
‘$\text{Plan}_{\text{House}}$’
‘$\text{Plan}_{\text{Office}}$’
Office_To_Out
$\text{Dist}_{\text{Office}} = 0$
0
House_To_Out
$\text{Dist}_{\text{House}} = 0$
0
$\text{In}_{\text{Office}}$
$\text{Dist}_{\text{Office}} = \text{-D}$
∞
$\text{Out}_{\text{Buildings}}$
$\text{Dist}_{\text{House}} = \text{dist-D}$
$\text{Dist}_{\text{Office}} = \text{dist-D}$
∞
In_{House}
$\text{Dist}_{\text{House}} = \text{-D}$
∞
Towards_Office
$\text{Dist}_{\text{Office}} = 0$
0
Towards_House
$\text{Dist}_{\text{House}} = 0$
0
$\text{UL}_{\text{Office}}$?’1’
$\text{UL}_{\text{Office}}$?’1’
If $dist_{Office} > 0$
UL_{House} ?’1’
UL_{House} ?’1’
If $dist_{House} > 0$
$\text{UL}_{\text{Office}}$?’0’
If $dist_{Office} \leq 0$
UL_{House} ?’0’
If $dist_{House} \leq 0$

Fig. 9 DEVS atomic model of an optimizer

  - $\text{UL}_{\text{Office}} = \{0,1\}$ This port indicates the user's location relative to the office:

  The external event $\text{UL}_{\text{Office}}$?‘0’ indicates that the user is heading office; the external event $\text{UL}_{\text{Office}}$?‘1’ indicates that the user is leaving office.
- $Y = \{IN_{HOUSE},\ OUT_{BUILDINGS},\ IN_{OFFICE}\}$;
- State variables:

  $S = \{(phase,\ Dist_{House}\ or\ Dist_{\ Office},\ ta)\}$, where:
  - Phase = $\{\text{In}_{\text{House}}, \text{In}_{\text{Office}}, \text{Out}_{\text{Buildings}}$, House_To_Out, Office_To_Out, Towards_House, Towards_Office };
  - ‘$\text{Dist}_{\text{House}}$’ or ‘$\text{Dist}_{\text{Office}}$’, is a variable that represents the current distance of the user from the object building (house or office), ranging from -D to dist, where ($\text{Dist}_{\text{House}} \leq 0$) presents ‘$\text{In}_{\text{House}}$’, ($\text{Dist}_{\text{Office}} \leq 0$) presents ‘$\text{In}_{\text{Office}}$’, ($\text{Dist}_{\text{House}} > 0$) and ($\text{Dist}_{\text{Office}} > 0$) presents ‘$\text{Out}_{\text{Buildings}}$’, $\text{Dist}_{\text{House}} \in [\text{-D}, \text{dist}]$, $\text{Dist}_{\text{Office}} \in [\text{-D}, \text{dist}]$;
- $Y = \lambda\ (Dist)$, where λ is the output function of this atomic model;
- $\delta_{ext}(In_{House},\ UL_{House}?‘1’) = (House\_To\_Out,\ Dist_{House}=0,\ 0)$ *and* $\delta_{ext}(Out_{Building},\ UL_{House}‘1’) = (House\_To\_Out,\ Dist_{House}=0,\ 0)$, $\delta_{ext}(Out_{Building},\ UL_{House}‘0’) = (Towards\_House,\ Dist_{House}=0,\ 0)$, $\delta_{ext}(Out_{Building},\ UL_{Office}‘1’) = (Office\_To\_Out,\ Dist_{Office}=0,\ 0)$, $\delta_{ext}(Out_{Building},\ UL_{Office}?‘0’) = (Towards\_Office,\ Dist_{Office}=0,\ 0)$, and $\delta_{ext}(In_{Office},\ UL_{Office}‘1’) = (Office\_To\_Out,\ Dist_{Office}=0,\ 0)$;
- $\delta_{int}\ (Towards\_House) = In_{House}$ if $\text{Dist}_{\text{House}} \leq 0$, $\delta_{int}\ (Towards\_Office) = In_{Office}$ if $\text{Dist}_{\text{Office}} \leq 0$, $\delta_{int}\ (Towards\_House) = Out_{Buildings}$ if $\text{Dist}_{\text{House}} > 0$, $\delta_{int}\ (Towards\_Office) = Out_{Buildings}$ if $\text{Dist}_{\text{Office}} > 0$, $\delta_{int}\ (House\_To\_Out) = Out_{Buildings}$ if $\text{Dist}_{\text{House}} > 0$, $\delta_{int}\ (Office\_To\_Out) = Out_{Buildings}$ if $\text{Dist}_{\text{Office}} > 0$;
- $\lambda(Towards\_House) = IN_{HOUSE}$ if $\text{Dist}_{\text{House}} \leq 0$, $\lambda(Towards\_House\ ) = OUT_{BUILDINGS}$ if $\text{Dist}_{\text{House}} > 0$, $\lambda(Towards\_Office) = IN_{OFFICE}$ if $\text{Dist}_{\text{Office}} \leq 0$, $\lambda(Towards\_Office) = OUT_{BUILDINGS}$ if $\text{Dist}_{\text{Office}} > 0$, $\lambda(House\_To\_Out) = OUT_{BUILDINGS}$ if $\text{Dist}_{\text{House}} > 0$, $\lambda(Office\_To\_Out) = OUT_{BUILDINGS}$ if $\text{Dist}_{\text{Office}} > 0$;
- $ta(House\_To\_Out) = 0$, $ta(Office\_To\_Out) = 0$, $ta(Towards\_House) = 0$, $ta(Towards\_Office) = 0$, $ta(In_{House}) = \infty$, $ta(In_{Office}) = \infty$ and $ta(Out_{Buildings}) = \infty$.

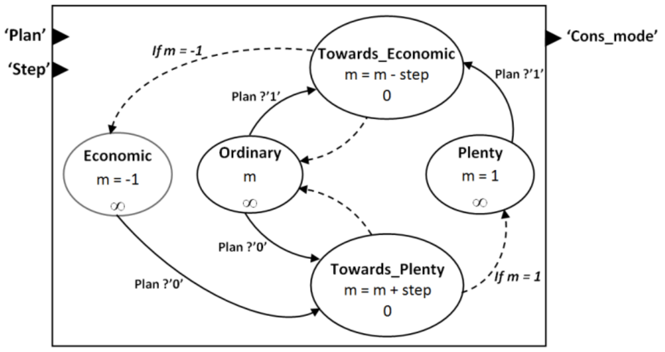


Fig. 10 DEVS atomic model of a solver

We finally move to the reactive layer, the solver that is an important player in the location based optimization of the building energy management. This element plays the role of actually adjusting the mode of energy use in the building according to the allocation plan received from the optimizer.

Figure 10 presents the DEVS atomic model of the solver with five states: ordinary, towards_economic, economic, towards_plenty and plenty. There are two inputs for the solver: ‘plan’ and ‘step’. The variable plan (trigger to plenty / trigger to economic) takes two values, ‘0’ or ‘1’; the value ‘0’ means trigger to plenty, and the value ‘1’ means trigger to economic. This model can be changed to the state of ‘plenty’, ‘economic’, ‘ordinary’.

The DEVS formal description of the ‘solver’ model is:

$$Solver = (X, S, Y, \delta_{ext}, \delta_{int}, \lambda, ta) \text{ Where:}$$

- *X = {Plan, Step}* present the input event variables, where:
  - Plan = {0,1} this port gives the order to trigger the policy of energy use :

  The external event plan?‘1’ indicates adjust towards plenty mode; the external event plan?‘0’ indicates adjust towards economic mode. ‘Step’ is an absolute value of adjustment, Step ∈ [-1, 1].
- *Y = {ECONOMIC, ORDINARY, PLENTY};*
- State variables: *S = {(phase, m, ta)},* where:
  - Phase = {plenty, ordinary, economic, towards_plenty, towards_economic};
  - ‘m’ is a variable that represents the current solver mode, ranging from -1 to 1, where -1 presents ‘economic mode’, 0 presents ‘ordinary mode’ and 1 presents ‘plenty mode’, m∈[-1, 1];
- *Y = λ (m),* where λ is the output function of this atomic model;
- δ_{ext} *(economic, plan?‘0’) = (towards_plenty, m = m + step, 0),* δ_{ext} *(ordinary, plan = ‘1’) = (towards_economic, m = m - step, 0),* δ_{ext}*(ordinary, plan?‘0’) = (towards_plenty, m = m + step, 0)* and δ_{ext} *(plenty, plan = ‘1’) = (towards_economic, m = m - step, 0);*
- δ_{int} *(towards_plenty) = ordinary* and δ_{int} *(towards_economic) = ordinary* if -1 <m<1, δ_{int} *(towards_economic) = economic* if m ≤-1, δ_{int} *(towards_plenty) = plenty* if m ≥1;
- *λ(towards_economic) = ECONOMIC* if m = -1, *λ(towards_economic) = ORDINARY* if -1 <m≤0, *λ(towards_plenty) = ORDINARY* if 0 ≤m<1 and *λ(towards_plenty)= PLENTY* if m = 1;
- *ta(towards_plenty) = 0, ta(towards_economic) = 0, ta(ordinary) = ∞, ta(economic) = ∞ and ta(plenty) = ∞.*

5.3 DEVS simulation model

Figure 11 presents a coupled DEVS model of an environment formed by a user which moves every day from house to office and coming back after finishing his daily work. This environment corresponds to implementing the proposed Framework presented in [7]. Three coupled models: *House EMS model, Office EMS model,* and *User location model,* have been separately modeled forming the simulation problem corresponding to this environment.

The simulation begins when the user leaves his house. This is confirmed when the user crosses a well-defined threshold. The user location service sends immediately a mode decision message to the EMS in the house building, which then makes the necessary changes to the energy saving policy by turning on/off the home hold appliances for that building. If the user enters his office's threshold, the user's location service also sends a mode decision message to the EMS in the office that also makes the necessary changes of energy saving policy in the office building. The section below describes in detail this energy saving strategy by changing energy use modes (economic, ordinary, and plenty), depending on the user’s location for each building, house or office.

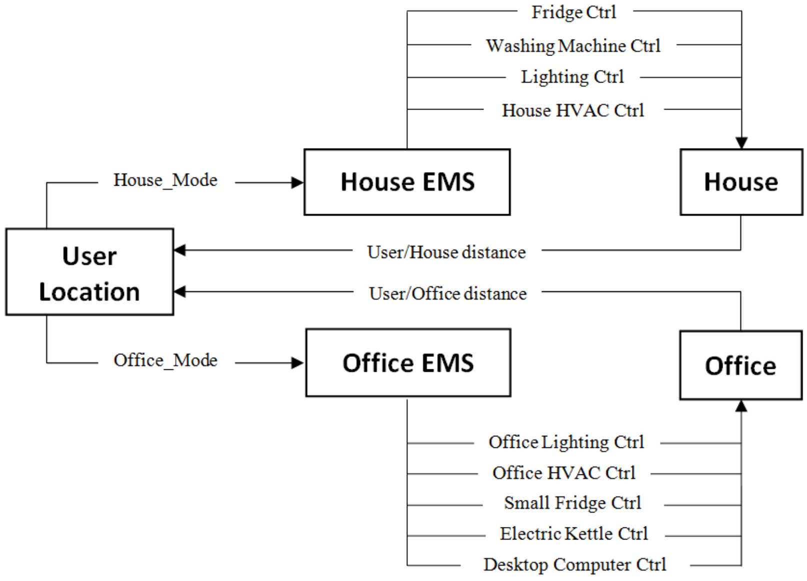


Fig. 11 DEVS simulation model of the energy management in house and office

6 Implementation

To evaluate the performance of the proposed approach, we implemented and simulated the Fog computing environment using JDEVS Simulator [32]. JDEVS is a tool whish implements the DEVS models in Java. JDEVS is composed of five independent modules shown in Figure 12 (squares correspond to the modules, diamonds correspond to the actors and the circles to the data exchange formats): a simulation engine, a graphic modeling interface, a storage module in libraries, a module for GIS connections and experimental frameworks for visualization and simulation.

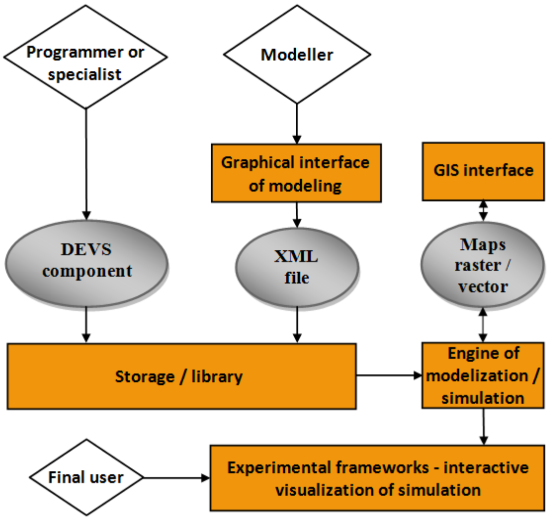

Programmer or specialist
Modeller
Graphical interface of modeling
GIS interface
DEVS component
XML file
Maps raster / vector
Storage / library
Engine of modelization / simulation
Final user
Experimental frameworks - interactive visualization of simulation

Fig. 12 View of JDEVS software modules

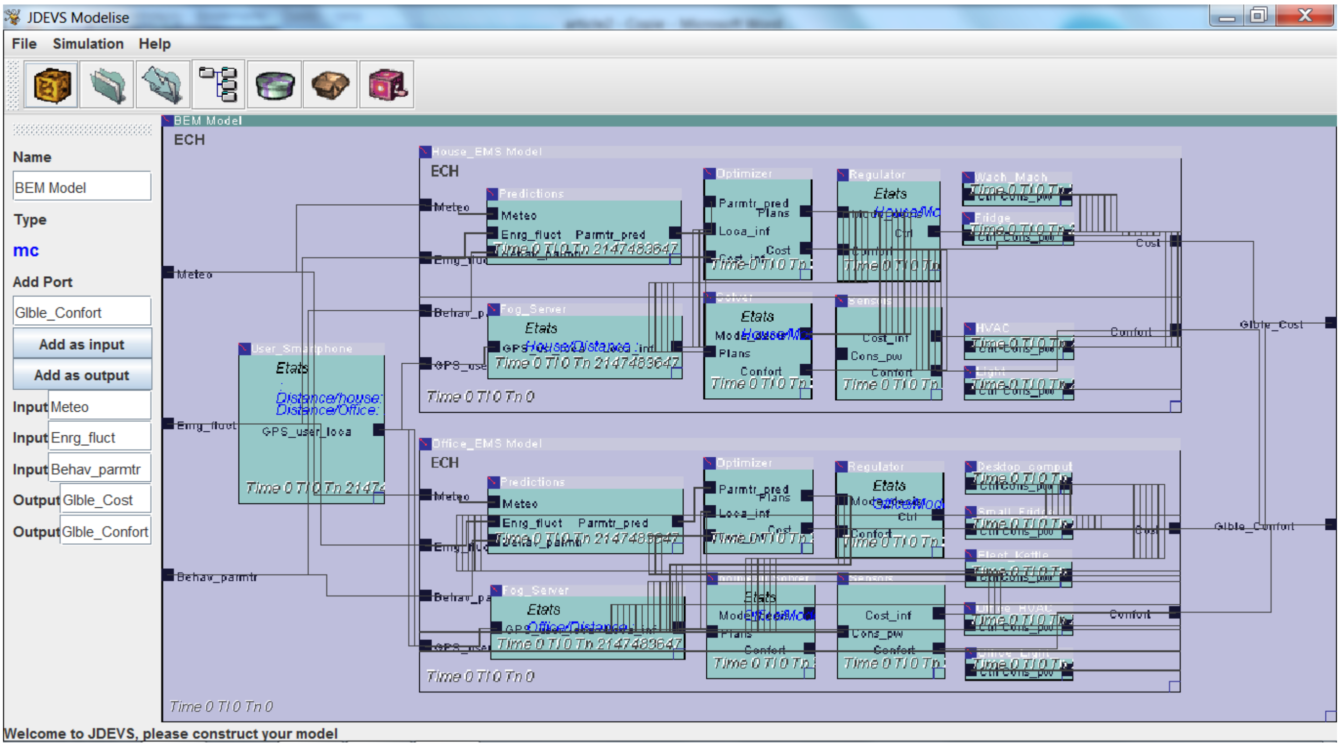
Each of these modules uses the class architecture defined in the framework which allows these modules to work together and to be able to be updated independently.

The simulation engine is an implementation of the classic-DEVS methodology with ports. The graphic modeling interface allows users to graphically build their models. Atomic DEVS models are automatically transformed into JAVA instructions. The storage module in libraries presents the models by domain and sub domains as a tree. Experimental Frameworks are the simulation user interfaces of the models stored in libraries. Experiments on diagram models can be performed directly from the graphical modeling interface. Several methodologies for coupling to GIS are used. The JDEVS inventors chose a "loose" type coupling. For this they use the Geotools library [33] which allows storing and reading Arcview, ASCII, and GML[2] file formats.

6.1 Simulation of the proposed model using the JDEVS tool

In DEVS, an experimental framework is used to perform validation tests. If the behaviour of the model and its system counterpart are within the limits of acceptable tolerance, the model is valid [23]. To this end, we have

[2] GML: Geographic Markup Language



Name: BEM Model

Type: mc

Add Port: Glble_Confort

Add as input

Add as output

Input Meteo

Input Enrg_fluct

Input Behav_parmtr

Output Glble_Cost

Output Glble_Confort

Welcome to JDEVS, please construct your model

Fig. 13 Representation of the global model using the JDEVS tool

modeled all the sub-systems present in the environment described above according to the formalism DEVS, and below we place these models under the JDEVS Simulation Tool to conduct experiments on them and study its efficiency and performance.

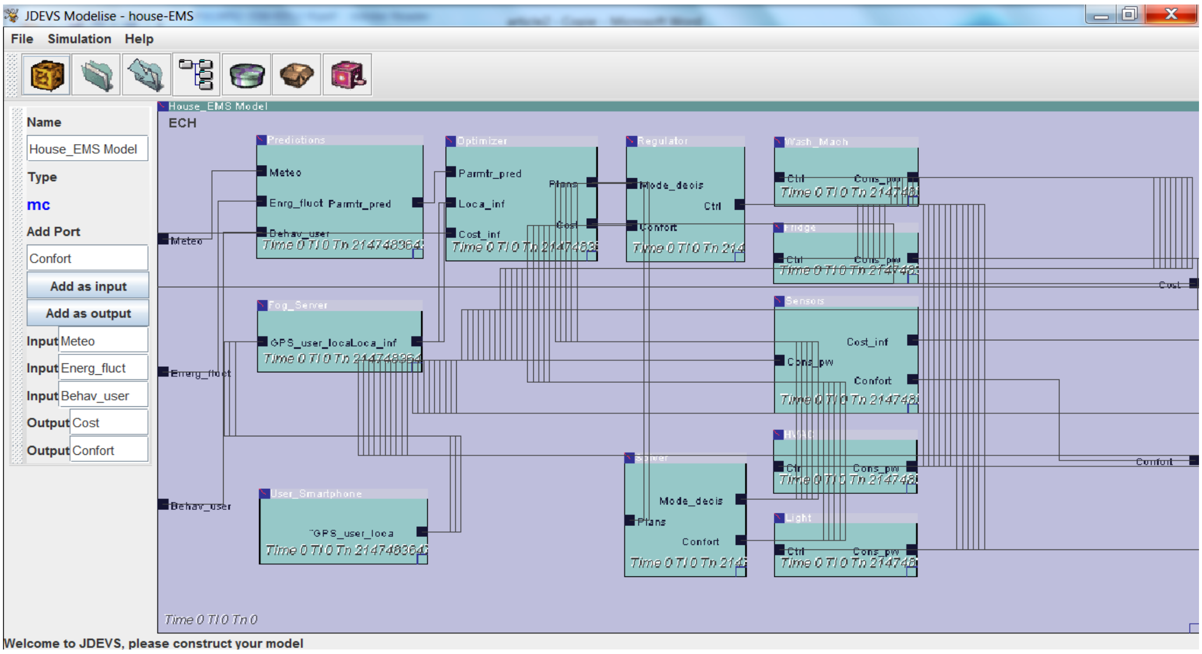
The model consists of two coupled models and a common atomic model between them: The *house EMS coupled model*, the *office EMS coupled model*; and the *Smartphone* atomic model (Figure 13). This is a generic model that represents different types of devices and other in the house EMS, the office EMS and the Smartphone atomic sub-system. The behavior and implementation of these components are similar but the functionality is different.

Figure 14 shows the house EMS coupled model implemented on JDEVS environment. This model consists of several atomic models. The anticipative layer contains *Predictions* and *Optimizer* atomic models, the reactive layer contains *Solver*, and *Regulator*, and the local layer contains *Fog Server*, *Sensors*, and home appliances atomic models represented by: *Fridge*, *Wash_Mach*, *Light* and *HVAC*.

The office EMS coupled model is the same as the house EMS coupled model. The only difference is in the home appliances. The anticipative layer contains *Predictions* and *Optimizer* atomic models, the reactive layer contains *Solver* and *Regulator*, and the local layer contains *Fog Server*, *Sensors* and home appliances atomic models represented by: *Small_Fridge*, *Desktop_Computer*, *Office_Light*, *Office_HVAC* and *Elect_Kettle*.

Initially, when the system users connect a device to the system, the device sends identification messages that include the name and ID to an identifier model. If authentication is correct, the system starts to work. Also, a working time schedule is attached for each layer of the mechanism of each EMS. The anticipative layer works on a time horizon of the order of an hour, the reactive layer of the order of one minute and the local layer works in real time.

In general, the main objective of validation is to check if a system works as expected, in different conditions and scenarios of simulation. The validation aims to make the model useful in the sense that it responds to good problem. Our problem was detailed and formulated in section 4. In the next section, we study the performance of this approach via the different simulation scenarios.



JDEVS Modelise - house-EMS

Name: House_EMS Model

Type: mc

Add Port: Confort

Input: Meteo

Input: Energ_fluct

Input: Behav_user

Output: Cost

Output: Confort

Welcome to JDEVS, please construct your model

Fig. 14 Representation of the house EMS coupled model using the JDEVS tool

6.2 Experimental results and discussion

In order to clarify the potential of this approach through an effective and accurate comparative study, we performed the following steps:

Before starting the simulations, we had to explain the three modes of energy consumption: *Ordinary*, *Economic* and *Plenty* modes, showing the amounts of energy consumed by all electrical devices in each mode for each of the two buildings, house and office. We show in Tables 1 and 2, respectively, the results of estimating daily energy consumption for house and office, explaining these three modes. Then, we change the user's location assuming that he spends 08 hours working, 08 hours inside the house and 08 hours outside the two buildings. We apply dynamic control and policy changes to the house and office through simulations and we record the results of energy consumption within 24 hours. We call this energy consumption mode ‟smart mode" and we compare it with the three virtual modes: plenty, ordinary and economic. Figures 15 and 16 show this comparison

Table 1 Daily House energy consumption estimation

	Lighting	Fridge	Washing machine	Other appliances	HVAC
Plenty mode (***)	8000 wh	2100 wh	2000 wh	3000 wh	16500 wh
Ordinary mode (**)	4000 wh	2100 wh	1000 wh	2400 wh	10518 wh
Economic mode (*)	2400 wh	2100 wh	600 wh	1500 wh	8410 wh
Total		(***): 31.60 kWh ;		(**): 20.02 kWh;	(*): 15.01 kWh.

Table 2 Daily Office energy consumption estimation

	Lighting	Small fridge	Electric kettle	Desktop computer	HVAC
Plenty mode (***)	1360 wh	800 wh	700 wh	3360 wh	4850 wh
Ordinary mode (**)	941 wh	800 wh	450 wh	1720 wh	3256 wh
Economic mode (*)	487 wh	800 wh	210 wh	672 wh	2798 wh
Total	(***): 11.07 kWh ;		(**): 07.17 kWh;	(*): 4.97 kWh.	

accurately. From what these two ffigures show, we noticed that after applying the rational adjustment of energy consumption to each of the two buildings, we just consume the amount of energy equivalent to what it is in the economic mode preserving all that the plenty mode provides of comfort. This way we have reduced energy consumption to the maximum, and without prejudice to the welfare requirements of users.

In order to prove that the user's location-based optimization is the cause of this achievement, we monitor the events through which the user changes his position in relation to both the house and office buildings. In other words, we will only record the events that lead to a difference in the building's user distance compared to the pre-determined distance for the boundaries of the two buildings (in our procedures, we considered that the specific distance to the area of each building equal to 05 meters by default). For this, we perform simulations of the following events (see Table 3 and Table 4).

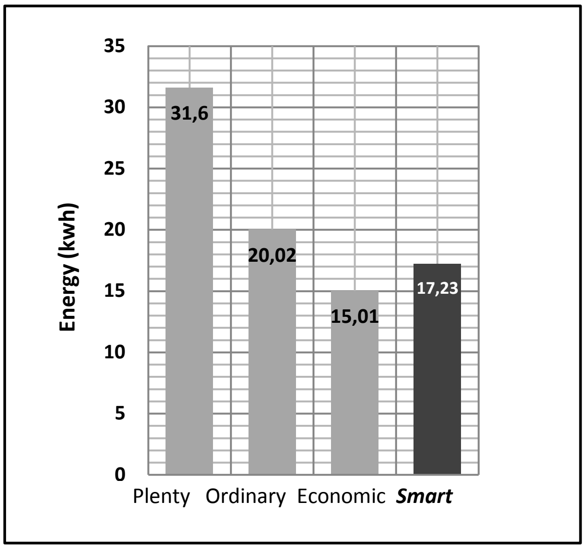


Fig. 15 Comparison of the energy consumption of the house with the three modes' energy estimation.

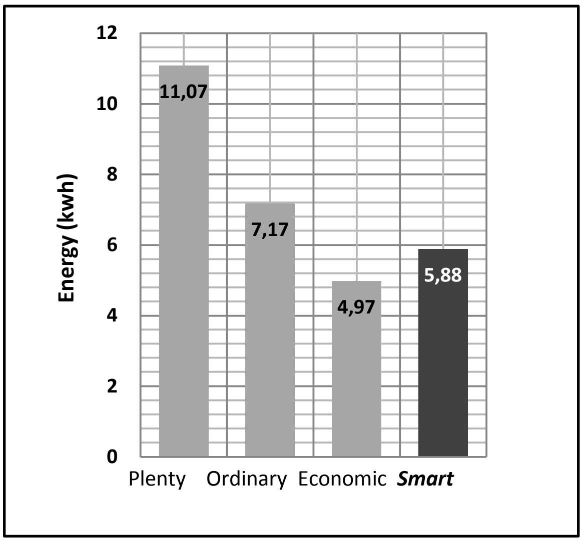


Fig. 16 Comparison of the energy consumption of office with the three modes' energy estimation.

Now let us observe, for example, the energy consumption of lighting in all situations of the user's presence, whether inside or outside the house area or the office. The amount of energy consumed by a simple lamp in the house or the office can be picked up from Table 3 or 4. Through this experiment, we explain how the system controls the lighting of the building by

Table 3 House simulation data

Date	Distance House/user	Lighting	Fridge	Washing machine	HVAC
June 18, 2019 12:06:49 (time 0)	12,75	863	374	0	614
June 18, 2019 12:09:31 (time 1)	0,00	871	402	6	698

Table 4 Office simulation data

Date	Distance Office/user	Lighting	Small fridge	Electric kettle	Desktop computer	HVAC
June 18, 2019 11:57:32 (time 0)	3,87	680	345	256	1269	2419
June 18, 2019 12:00:11 (time 1)	26,47	683	362	256	1287	2468

monitoring the user in two distances, one of them is less than the minimum distance specified for the building area which is equal to 5 meters, and the second is greater than that distance. In this work, we consider that the lighting system consists of a group of simple lamps and their control is in a group way, i.e. there are only two cases for this system: turn on the lighting or turn off the lighting.

As explained in the previous example of the atomic model of lighting (see in Section 5.2), a simple lamp can only be switched on or off and its atomic model consists of four states: Off, On-try, On, and Off-try. The initial phase of the model 'lamp' is 'off'. At time zero, the distance between the user and his office exceeded the minimum (Distance <5). During this moment the minimum distance is lowered, the system immediately turns on the lighting in the building, i.e. when the model 'lamp' receives the "in" event on its "1" port, it goes to the 'on' phase. This transition in cases was made by the external transition function. At time one, the distance between the user and his office is higher than the minimum value (Distance >5). During this moment, the system immediately turns off the lighting in the office building, i.e. when the lamp model receives the "in" event on its "0" port, it goes to the 'off' phase. This transition in the case was also made by the external transition function.

Now, we compare, in Figure 17 and Figure 18, the 24-hour simulation results of the total energy consumed of building lighting devices (house or office) with the default results of energy consumption by the same lighting devices for two buildings that do not use this location-based optimization of energy use. We note that the actual energy consumption of the system after applying this optimization is much lower than in buildings that do not apply it; And this difference begins to increase precisely in the minutes when the user is within the boundaries of either of the two buildings, until we get a large portion of the energy gain in the end of the day (27,11 % for the house and 27.78 % for the office).

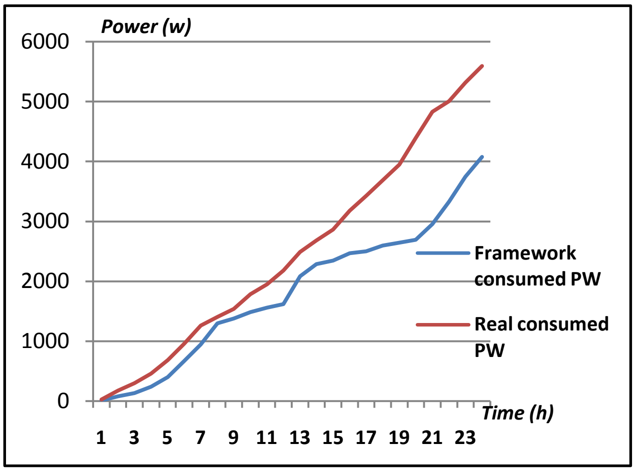


Fig. 17 Daily house's lighting energy consumption

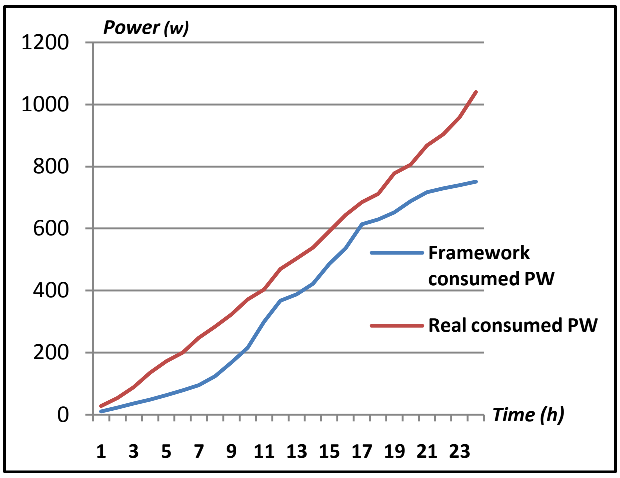


Fig. 18 Daily office's lighting energy consumption

In conclusion, in order to confirm these results and deepen our analysis, a third simulation was carried out over a whole week. The results are shown in two figures. Compared to real consumption, Figures 19 and 20 show total optimized energy consumption for the house and office respectively, for an entire week. Through the curves shown in the figures, it appears after the comparison that we gained about 18.76 % in total energy consumption per week for the house

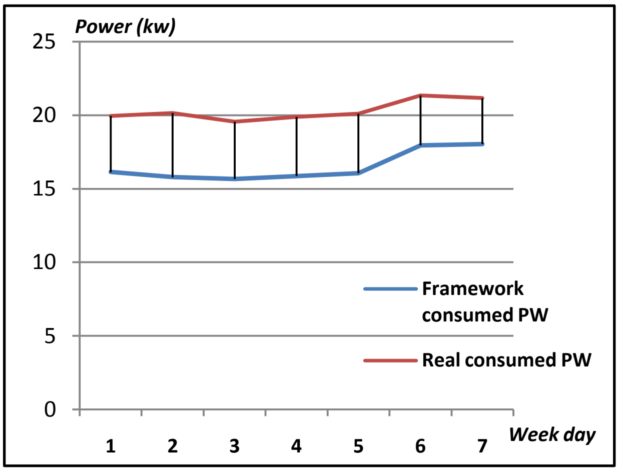


Fig. 19 Weekly house energy gain

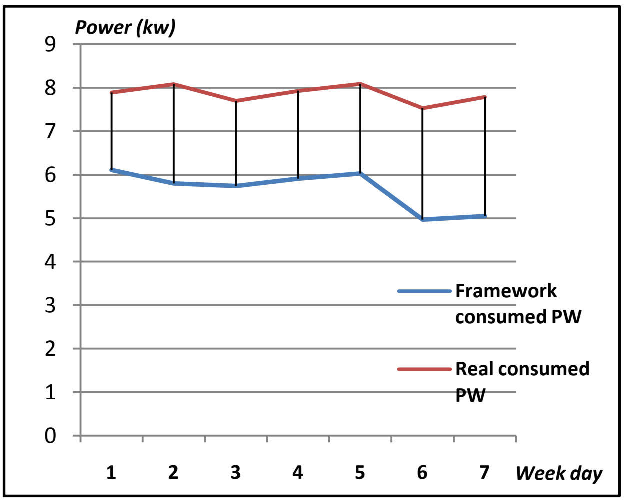


Fig. 20 Weekly office energy gain

after applying the optimization approach based on user location and about 27.99 % for the office. This means that with this new approach, general users will enjoy a luxurious lifestyle without any special care or hassle, and they will pay only what economy mode users pay. We also note and confirm by observing the speed of implementation that the secret to the quality of performance of this smart Framework lies in the use of fog computing paradigm in its design**.**

6.3. Analytical comparison

The aim of this brief elucidation is to compare the proposed solutions with the latest similar work in terms of the Framework or structure used, application scenario, design, method of treatment, and the results obtained in particular the energy and complexity of the algorithms. Since this work is the first to model and simulate, under DEVS, a complete and intelligent Framework to optimize the use of energy in a suitable sample environment (To the best of our knowledge), we compare it with the closest and most similar proposed solutions. For this purpose, we adapted the works suggested in [20].

Our work is mainly based on a holistic and intelligent Framework that relies on IoT technology, inspired by the fog computing paradigm, to locate the user for two buildings, house and office in order to rationalize energy use in them. Conversely, the work suggested in [20] is not dependent on any Framework, system or structure. Given this, we can judge that our work is more organized and adopts a correct methodology in how to achieve its goals, especially to obtain luxury for the benefit of users at the lowest cost and in a systematic and rational way, and all this is done very quickly through the adoption of a fog computing paradigm to locate the user.

Regarding the application scenario, we have proposed an environment which is an initial sample representing a user moving from his home to his office to do his work, the objective is to create a mini-environment out of a larger environment that contains the movement of many users of the various buildings that form at least an area of a smart city, and this was intentional. In the proposed work at [20], the application scenario was just a four-room home, which in our opinion is difficult to generalize even to a small part of the smart city components.

The design should be based on dynamic and formal models to support formal simulations for formal verification [21]. In DEVS, design begins from the atomic models, and after pinning them together we obtain the coupled models, which are held together until we obtain the complete model. In [20] we did not observe this method of fixing the sub-models, which added some ambiguity to the design method. Thus, in the event that the model representing the real environment is ambiguous, we cannot convincingly verify the effectiveness of the proposed solution.

The two works used the DEVS formalism as a modeling and simulation tool, although in a very different way. But what is remarkable is that the work [20] has used it for other purposes out of the ordinary, which in our opinion is an adventure. Unlike us, instead of using mathematical optimization to find optimal solutions, this work proposes a modeling and simulation methods in order to provide good decisions and recommendations for devices' scheduling.

Now we turn to the results obtained. In Table 5, we summarize the simulation results of the weekly home energy consumption for each of our works to compare the percentage of gains that were achieved after each work applied its suggested solutions. By comparing the percentages obtained by optimizing energy use in both works within a week (21,34 % for our work, 12,19 % for the work [20]), it clearly appears that the reduction

Table 5 Total weekly Energy consumption in both works

The proposed work	Scenario	Total energy consumption (kw)	Energy gained (%)	Simulation time
[20]	Fixed schedule	675.47kw	12.19%	One week
	Modified schedule	593.14kw		
Our work	Framework not applied	197.13kw	21.34%	One week
	Framework applied	155.07kw		

in energy use resulting from our work outweighs the optimizing resulting from the work suggested in [20] by nearly half (about 43%). Also with regard to the complexity of the algorithms, by following the fog computing Framework, we obtain a faster speed to implement it thanks to the local calculation guaranteed by the fog servers installed inside the buildings (Away from the Cloud). Moreover, this Framework provides virtual storage in case it is actually installed [7].

In summary, and compared to the work [20], we conclude by saying that we were able in a smart and robust way due to its clarity, to optimize the use of energy in a more suitable environment to facilitate the realization of smart energy for a smart city.

7 Conclusion and future prospects

The building sector constitutes a major potential source of energy savings, particularly through the rationalization of energy use, which has attracted the interest of many researchers to contribute to its development through valuable proposals [34].

In this paper, we presented a new method allowing to optimize energy management in smart buildings that depends on locating the user in the rationalization of energy use. The work offers contributions in two areas:

1. in the architectural design of the system derived from the energy management Framework proposed in the work [7].
2. in the use of DEVS formalism for assessing new methods and tools to modeling and simulating this resulting system.

To design the system compatible with this Framework, we relied on the environment made up of a user of two buildings, his home and his office, always accompanied by his mobile phone. We view this environment as a main portal to create larger and more complex environments, such as an entire city, for which we aspire to find solutions to make it smarter. Regarding modeling and simulation, we justify our choice with the modular nature of the formalism DEVS exploited in this study to form sub-models that allow field experts to independently develop simulation techniques and then integrate their work until they arrive at the overall model. This division of aggregated systems facilitates their modeling and simplifies their complexity.

In the future, based on this work, we plan to create a Framework similar to this, in terms of the entire structure, which can manage the energy consumption in an entire city, regardless of scenes number. Therefore, we consider this work as a starting sample for a small environment that can be generalized into a larger Framework that can represent a wider environment such as a smart city.

Acknowledgements

We sincerely thank the General Directorate of the Scientific Research and Technological Development (DGRSTD) of the Ministry of Higher Education and Scientific Research of Algeria, for all the resources provided for the accomplishment of this project.

References

1. Bellavista P, Berrocal J, Corradi A, Das S K, Foschini L, Zanni A. A survey on fog computing for the Internet of Things. Pervasive Mob. Comput. 2019; 52(6): 71–99. https://doi:10.1016/j.pmcj.2018.12.007.

2. Jalali F, Khodadustan S, Gray C, Hinton K, Suits F. Greening IoT with Fog: A Survey. In: 2017 IEEE International Conference on Edge Computing (EDGE), Honolulu, HI, (2017), 25–31.
https://doi: 10.1109/IEEE.EDGE.2017.13.

3. Atlam H F, Walters R J, Wills G B. Fog computing and the internet of things: a review. Big Data and Cognitive Computing, 2018; 2(10) :1–18.
https://doi.org/10.3390/bdcc2020010.

4. Maatoug A, Belalem G. Conception and validation of smart building energy management system BEMS using the discrete event system specification DEVS. Journal of Communications Software and Systems (JCOMSS) 2014; 10(2): 107–113.
https://doi: 10.24138/jcomss.v10i2.131.

5. Lau B P L, Marakkalage S H, Zhou Y, Hassan N U, Yuen C, Zhang M, Tan U X. A survey of data fusion in smart city applications. Information Fusion Dec. 2019; 52: 357–374.
https://doi.org/10.1016/j.inffus.2019.05.004.

6. Zao J K, Gan TT, You C K, Cheng E C, Wang Y T, Mullen T, Jung T P, 2014. Augmented brain computer interaction based on fog computing and linked data. In: Proceedings of the International Conference on Intelligent Environments (IE), Shanghai, China, (2014), 374–377.
https://doi: 10.1109/IE.2014.54.

7. Maatoug A, Belalem G, Mahmoudi S. Fog computing framework for location-based energy management in smart buildings. Multiagent and Grid Systems 2019; 15(1): 39–56.
https://doi: 10.3233/MGS-190301.

8. Etemad M, Aazam M, St-Hilaire M. Using DEVS for modeling and simulating a Fog Computing environment. In: 2017 International Conference on Computing, Networking and Communications (ICNC), Santa Clara, CA, USA, (Jan 2017), 849–854.
https://doi: 10.1109/ICCNC.2017.7876242.

9. Himeur Y, Alsalemi A, Al-Kababji A, Bensaali F, Amira A. Data fusion strategies for energy efficiency inbuildings: Overview, challenges and novel orientations. Information Fusion 2020; 64: 99–120.
https://doi.org/10.1016/j.inffus.2020.07.003.

10. Stojkoska B L R, Trivodaliev K V. A review of internet of things for smart home: Challenges and solutions. Journal of Cleaner Production 2017; 1(140): 1454–1464.
https://doi.org/10.1016/j.jclepro.2016.10.006.

11. Carrillo E, Benitez V, Mendoza C, Pacheco J. IoT framework for smart buildings with cloud computing. In: 2015 1st International Conference of Smart Cities (ISC2), Guadalajara, Mexico, (Oct 2015), 1–6.
https://doi: 10.1109/ISC2.2015.7366197.

12. Tushar W, Yuen C, Li K, Wood K, Wei Z, Xiang L. Design of cloud-connected IoT system for smart buildings on energy management (invited paper). EAI Endorsed Transactions on Industrial Networks and Intelligent Systems 2016; 3(6): 1–9.
https://doi: 10.4108/eai.1-1-2016.150813.

13. Faruque M-A-A, Vatanparvar K. Energy management-as-a-service over fog computing platform. IEEE Internet of Things Journal 2016; 3(2): 161–169.
https://doi: 10.1109/JIOT.2015.2471260.

14. Javed A, Rana O, Marmaras C, Cipcigan L. Fog paradigm for local energy management systems. In: 2nd EAI International Conference on Cloud, Networking for IoT Systems, Brindisi, Italy, (Apr 2017), 1–10.
https://doi.org/10.1007/978-3-319-67636-4_18.

15. Naranjo P G V, Pooranian Z, Shojafar M, Conti M, Buyya R. FOCAN: A Fog-supported smart city network architecture for management of applications in the Internet of Everything environments. Journal of Parallel and Distributed Computing 2019; 132: 274–283.
https://doi.org/10.1016/j.jpdc.2018.07.003.

16. Lin H, Liu G, Li F, Zuo Y. Where to go? Predicting next location in IoT environment. Front. Comput. Sci

2020; 15(1): 151306 (2021).
https://doi.org/10.1007/s11704-019-9118-9.

17. Pan J, Jain R, Paul S, Vu T, Saifullah A, Sha M. An internet of things framework for smart energy in buildings: Designs, prototype, and experiments. IEEE Internet of Things Journal 2015; 2(6): 527–537.
https://doi: 10.1109/JIOT.2015.2413397.

18. Moreno M V, Zamora M A, Skarmeta A F. User-centric smart buildings for energy sustainable smart cities. Trans. Emerging Tel. Tech. 2014; 25(1): 41–55.
https://doi.org/10.1002/ett.2771.

19. Luan T-H, Gao L, Li Z, Xiang Y, Sun L. Fog computing: Focusing on mobile users at the edge. Preprint, (2015), 1–11.
available online arXiv:1502.01815 [cs.NI].

20. Albataineh M, Jarrah M. DEVS-IoT: performance evaluation of smart home devices network. Multimedia Tools and Applications (2020), 1–29.
https://doi.org/10.1007/s11042-020-09186-w.

21. Bonino D, Corno F D. Dogsim: a state chart simulator for domotic environments. In: International conference on pervasive computing and communications workshops (PERCOM workshops), IEEE (2010), 208–213.
https://doi: 10.1109/PERCOMW.2010.5470666.

22. Yi S, Hao Z, Qin Z, Li Q. Fog computing: Platform and applications, in: Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb), Washington, DC, USA, (Nov 2015), 73–78.
https://doi: 10.1109/HotWeb.2015.22.

23. Zeigler BP, Praehofer H, Kim TG. Theory of modeling and simulation: integrating discrete event and continuous complex dynamic systems. Academic press, San Diego (2000). ISBN: 0127784551.

24. Capocchi L, Bernardi F, Federici D, Bisgambiglia P-A. BFS-DEVS: A general DEVS-based formalism for behavioral fault simulation. Simulation Modelling Practice and Theory 2006; 14(7): 945–970.
https://doi.org/10.1016/j.simpat.2006.05.002.

25. Stojmenovic I, Wen S. The fog computing paradigm: Scenarios and security issues, in: Federated Conference on Computer Science and Information Systems (FedCSIS), Warsaw, Poland, (Sep. 7–10, 2014), 1–8.
https://doi: 10.15439/2014F503.

26. Varghese B, Wang N, Nikolopoulos D S, Buyya R. Feasibility of fog computing. arXiv preprint, arXiv:1701.05451. 2017. [Online]. Available: https://arxiv.org/abs/1701.05451.

27. Guo X, Ansari N, Hu F, Shao Y, Elikplim N R, . Li L. A survey on fusion-based indoor positioning. IEEE Comm. Surveys & Tutorials. 1st Quart. 2020; 22(1): 566–594.
https://doi: 10.1109/COMST.2019.2951036.

28. Cesconetto J, Augusto Silva L, Bortoluzzi F, Navarro-Cáceres M, A. Zeferino C, R Q Leithardt V. PRIPRO—Privacy Profiles: User Profiling Management for Smart Environments. Electronics. 2020; 9(9): 1519.
https://doi.org/10.3390/electronics9091519.

29. Abyaneh A Y, Pourahmadi V, Foumani A-H-G. CSI-based authentication: Extracting stable features using deep neural networks . Trans. Emerging Tel. Tech. 2019; 30(11): 1–11. https://doi: 10.1002/ett.3795.

30. Belhadi A, Djenouri Y, Srivastava G, Djenouri D, Lin J C W, Fortino G. Deep learning for pedestrian collective behavior analysis in smart cities: A model of group trajectory outlier detection. Information Fusion. 2021; 65: 13–20.
https://doi.org/10.1016/j.inffus.2020.08.003.

31. Bittencourt L F, Diaz-Montes J, Buyya R, Rana O F, Parashar M. Mobility-aware application scheduling in fog computing. IEEE Cloud Comput. Mar./Apr. 2017; 4(2): 26–35. https://doi: 10.1109/MCC.2017.27.

32. Filippi J-B, Bisgambiglia P. JDEVS: an

implementation of a DEVS based formal framework for environmental modeling. Environmental Modelling & Software 2004; 19(3): 261–274. tthps://doi.org/10.1016/j.envsoft.2003.08.016.

33. Geotools (2003). Geotools groupe d'utilisateurs. http ://www.geotools.org.

34. Shah A S, Nasir H, Fayaz M, Lajis A, Shah A. A review on energy consumption optimization techniques in IoT based smart building environments. Information 2019; 10: 108. https://doi: 10.3390/info10030108.

Abdelfettah Maatoug received his Magister degree in Computer Science from the National Superior School of Computer Science of Algiers, Algeria, in 2015. He is an assistant professor in the University of TIARET, Algeria**,** and is a PhD candidate in Department of computer science, Faculty of exact and applied sciences, University of Oran1 Ahmed Ben Bella, Algeria. He is a member of Computer Science Laboratory of Oran. His research interests include Distributed Systems, IoT, Grid computing, Cloud and Fog computing, Energy Management, Economic models, Smart Cities, Smart Homes and Smart Buildings.

Ghalem Belalem graduated from Department of computer science, Faculty of exact and applied sciences, University of Oran1 Ahmed Ben Bella, Algeria, where he received PhD degree in computer science in 2007. His current research interests are distributed system, grid computing, cloud computing, replication, consistency, fault tolerance, resource management, economic models, energy consumption, Big data, IoT, mobile environment, images processing, Supply chain optimization, Decision support systems, High Performance Computing.

Saïd Mahmoudi received his MS in Computer Science from University of Lille1, France in 1999. In 2003, He obtained his PhD in Computer Science at the University of Lille1. Between 2003 and 2005, he was Associate Lecturer at the University of Lille3. Since September 2005, he is Associate Professor at the Faculty of Engineering of the University of Mons, Belgium. His research interests include artificial intelligence, internet of things, medical images analysis, 2D and 3D retrieval and indexing, Big Data and large scale multimedia management. He published over then 150 journal and conference papers to date. He participates in numerous scientific and academic activities (member of the organizing committee for some conferences, reviewer and guest editor for several internationals journals).