Personal and Ubiquitous Computing manuscript No.
(will be inserted by the editor)

On-demand Energy Monitoring and Response Architecture in a Ubiquitous World

Oihane Kamara-Esteban · Ander Pijoan · Ainhoa Alonso-Vicario · Cruz E. Borges

Received: date / Accepted: date

Abstract Energy demand is increasing globally and in consequence greenhouse gas (GHG) emissions from this sector are on the rise as well. This trend is set to continue, driven primarily by the economic growth and the rising population. Solutions in this area go hand in hand with the worldwide deployment of policies that look forward a better management and usage of energy in both domestic and industrial scopes. In this line, load balancing through demand-response strategies comes out as one of the most effective and immediate actions aimed at achieving efficiency in the use of energy resources. We present GeoWorldSim, an agent-based simulation platform that integrates the development of a human activity model as well as the communication middleware known as FI-WARE in order to test the best communication architectures available for the implementation of demand-response strategies.

Keywords Agent-based Simulation · Ubiquitous World · Demand-Response · Smart-Grid

1 Introduction

The world is currently channelling all its efforts towards fighting climate change. Since the first Conference of the Parties (COP) in 1995, greenhouse-gas (GHG) emissions have risen by more than one-quarter, with their atmospheric concentration increasing steadily to 435 parts per million carbon-dioxide equivalent (ppm CO_{2eq}) by 2015 [2]. In particular, greenhouse-gas emissions from the energy sector represent about two-thirds of all the greenhouse-gas emissions worldwide [3], therefore the deployment of effective actions in this area is essential in order to tackle one of the main drivers of climate change.

On its part, the EU has set forward a variety of policies which devise a novel deal for energy consumers in response to the new changes in consumption patterns [13]. One aspect of these changes has to do with the arrival of small-scale decentralised energy installations located in domestic backyards, often linked to the notion of *energy prosumers*. This term refers to entities such as households, cooperatives and local enterprises that are both producers and consumers of energy. Hand in hand with this measure, the massive implementation of demand-response control [5] is one of the most forward and immediate strategies available to reach the energy efficiency objectives set in the EU road map for climate change.

The traditional control strategy of the electrical grid in terms of energy generation management aims to balance the amount of energy being generated and the amount of energy being consumed by acting over the *generation*. In contrast, demand-response strategies try to act over the *demand*. To achieve this goal, demand-response strategies rely heavily on the constant monitoring of all the entities living within the energy grid and the thorough analysis of the behavioural patterns that define the shape of the energy loads. Energy loads can be classified into three types: loads that occur automatically when a device needs energy (such as the engine of a fridge or the resistance of a heater), loads that can be timetabled along the day (like those of the dishwasher or the washing machine), and loads that need to take place on demand (like those of the TV or the oven). Traditionally, the objective of a demand-response con-

DeustoTech - Fundación Deusto, Avda Universidades, 24, 48007, Bilbao.
Facultad Ingeniería, Universidad de Deusto, Avda. Universidades, 24, 48007, Bilbao.
E-mail: {oihane.esteban, ander.pijoan, ainhoa.alonso, cruz.borges}@deusto.es

trol is the creation of an accurate schedule for the first and second load types considering both as equals when, clearly, in the second case a human decision needs to be taken into consideration. Nevertheless, failing to regard human aspects within a demand-response scenario may lead to undesirable side effects [9].

Several deployments of Smart Homes and Smart Grids in real environments have been carried out so as to understand the particular aspects of demand-response strategies. Smart Home technologies are based on what is known as the Internet of Things (IoT). IoT refers to a group of interconnected objects identifiable through a digital network that can sense and be controlled remotely, resulting in improved efficiency, accuracy, and economic benefits for the final user [38]. In fact, the versatility of IoT ranges from Home Automation on a simple, isolated level, to its use in Smart Grids for the adequate management and balancing of resources for several buildings connected to a common grid.

The MavHome project [15] is a clear example of implementation of Smart Home in real environments, aimed at creating a sandbox that perceives the state of the home through sensors and acts upon it through device controllers. Its architecture is divided into four layers: a) physical (60×10 hardware devices plugged into the home electric wiring system), b) communication (exchange of information among devices) c) information (generation of knowledge for decision making) d) decision (execution of actions based on the information supplied). Decision making is based on a finite-order Markov model which predicts inhabitant future actions so as to automate repetitive activities and meet the house goals in terms of comfort and costs. Other projects like GatorTech [20] are heavily focused on elder care and proactive health. In this case, the authors propose a middleware architecture that comprises separate physical, sensor-platform, service, knowledge, context-management, and application layers. The central system collects the data provided by the sensors and gives indications to whether there is any anomaly in the residents behaviour or vital signs.

On a broader scale, IoT can be applied to distribution networks or Smart Grids, enabling a whole new myriad of services including, among others, energy load balancing, intelligent water provisioning, maintenance of sewerage networks, avoidance of traffic congestions, demand-response, and even waste management. The quest for sustainable energy models is the main factor driving the research on Smart Grid technology, since it represents the bridging paradigm to enable highly efficient energy production, transport, and consumption through the the entire distribution network. Specifically, in the case of Smart Grids, the deployment comprises the leverage of several Smart Homes connected to a common grid. Data from several sources is then aggregated and analysed from a centralised point of view with the purpose of balancing shared resources.

Projects like Nemo&Coded (NEtworked MOnitoring & COntrol, Diagnostic for Electrical Distribution) [28] focus on the modelling, design, implementation and operation of networked hardware/software smart devices for the low voltage electrical distribution domain by building dynamic energy efficiency services [29]. The infrastructure consists of an acquisition platform for collecting energy data in real time by guaranteeing the interoperability of the hardware/software devices, regardless of their equipment or communication technologies. The main novelty lies in the use of intelligent and autonomous distributed nodes to process semantic information known as PGDINs [17], which integrate several of the large number of existing standards in the electricity sector, enabling the autonomous processing of events and data.

Another example [19] is the PRICE project, which tested several topics related to the Smart Grid, such as supervision and automation, methodologies, new energy management solutions, and demand-response strategies, on a huge real deployment (around half million users were involved). Concerning demand-response, 50 homes granted their Distribution System Operators direct control of their smart appliances (including Air Conditioning systems) to test demand-response strategies.

The smart grid experimental system implemented by the Fukushima National College of Technology [21] is an interesting example of real scale deployment for teaching purposes. The system consists of a gas engine co-generation system, a wind power system, a solar power system, a battery system, and an uninterruptible power supply system that provides power during failures in the distribution network. A SCADA system monitors the electricity demand and balances resources according to the needs of the users and the optimum operation of the whole system.

In all cases, there are several important issues that must be taken into account, such as the lengthy process of configuration and maintenance of the electronic devices, as well as the amount of time needed to acquire a sufficient amount of data for analysis. Moreover, the high economic investment and time restrictions related to the deployment of Smart Homes and Smart Grids in real or scaled environments highlights the advantages of *simulation models*.

The Interactive Smart Home Simulator (ISS) [36] proposes a context aware simulation system comprised of several electronic devices distributed throughout an apartment. The simulator relies on a context retriever that requests and receives sensor information from the home appliances. A centralised server represents the central control which purpose is to take decisions according to changes in the environment and the behaviour of the house occupants. The aim of this simulator is to demonstrate the exchange and update of information as perceived by the Smart Home.

The Intelligent Project Home (IHome) [26] uses multi-agent system technologies to manage an intelligent Smart Home environment. The simulation is populated with distributed intelligent home-control agents that manage appliances and negotiate over shared resources with the objective of automating human tasks. Each household appliance, represented by an intelligent agent, is used as input to a model that coordinates electricity, hot water, noise or sound levels, and temperature in each of the rooms modelled. The agents make decisions about the activity performed by the occupant depending on resource availability, i.e., whether there is not enough hot water, the external temperature, etc.

Within Smart Grids, projects like MASGrip [27] propose a multi-agent system to model the internal operation of all the participants typically involved along the entire distribution network. Each participant, represented by an intelligent agent, has the capability of simulating the actions that the corresponding real world entity can carry out in terms of decision-making and communication. Apart from the agents representing the Smart Grid devices, MASGrip uses a special type of agent which purpose is to simulate negotiations within the electricity market (MASCEM). Please note that, providing the means to reproduce this negotiation is essential in order to build an efficient quality analysis of the simulated smart grid management. This information, however, is not always publicly available.

Karnouskos et al. [24] propose an agent simulation system that simulates discrete heterogeneous devices that consume and/or produce energy and are able to act and collaborate autonomously. The bottom layer is the simulator layer and contains all the agents which represent energy generating and/or consuming devices. The agents use the full communication capabilities offered by the agent platform in order to collaborate and maximise user comfort.

A careful analysis in terms of real Smart Grid deployments and simulators shows that none of them were built with the purpose of analysing the human aspects of demand-response strategies: What are the potential impacts that may derive from the implementation of demand-response strategies? What are the effects of deploying these solutions in case that human factors are not considered? What strategies will work better and how can they be implemented in order to avoid rejection from citizens?

In order to try to clarify all these questions, we present the first steps towards the implementation of a Smart City Agent Based Simulator, GeoWorldSim, for evaluating demand-response strategies, following the GreenSoul methodology [7] [8]. The main objective of this paper is to study the best communication architectures available for implementing a demand-response strategy. Later articles in this research program will deal with the selection of an appropriate middleware, the development and testing of several control strategies for the grids and finally, the enrichment of these control strategies to take into consideration human behaviour and foster its modification. See Section 6 for further information about the next steps.

The rest of the paper will consist on a presentation of the GeoWorldSim architecture (Section 2), a detailed explanation of the implementation (Section 3), the description of the experiments carried out (Section 4) and an assessment of the results (Section 5).

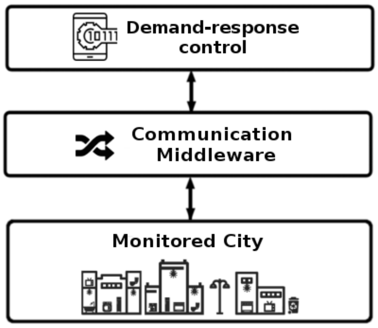
2 Ideal architecture for a demand-response strategy scenario

Figure 1 shows the ideal architecture of a demand-response scenario and how we have mapped each element to our simulation approach. The bottom layer of the architecture represents a city which energy demand is constantly monitored. Information regarding the energy consumption of devices within the city is sent to a common communication middleware which centralises all the information, registering all the events and notifying them to the Demand-Response Control. The Demand-Response Control analyses all the data received and provides a set of signals back to the monitored city so as to influence citizens behaviour with the objective of leveraging energy demand and energy production.

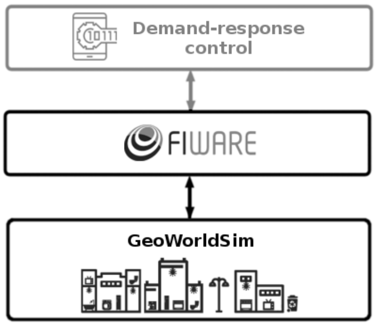
Following these schema, our approach consists on developing an agent-based system (GeoWorldSim) that emulates the behaviour of a city in terms of energy consumption, and evaluates the feasibility of deploying several communication architectures depending on the amount and type of devices connected to the communication middleware chosen, in this case FI-WARE. The development and implementation of the Demand-Response Control will be carried out in future research.

2.1 GeoWorldSim

GeoWorldSim comprises an Environment-centric Situated Multi Agent System, which is an extension of



(a) *Real architecture*



(b) *Simulation architecture*

Fig. 1: Real and simulation architectures for the demand-response strategy scenario.

our human activity simulator MASSHA [22]. The Environment consists of a physical environment and a social environment, which come together to form the Intelligent Environment where all agents, whether they are active (interaction initiators) or passive (reactors to stimuli), are situated. The physical environment ensures that none of the spatial constraints are violated while the social environment is responsible for managing communications, roles and knowledge by giving agents a social and communicative context. Further information about how the agents emulate human behaviour and energy consumption is detailed in Section 3.1. GeoWorldSim is built upon Qt [32], a widely used cross-platform framework for developing native C++ applications that benefits from interesting extensions such as *signals and slots* for remote event invocation, a meta-object compiler, and the ability to detach pieces of code and move them to separate threads. All of Qt features make it possible to model accurate agents and passive entities by using an event-driven paradigm. The execution flow of an agent is heavily influenced by events such as messages from other agents, sensor outputs, or even its own actions just like in real life, where humans are constantly bombarded by stimuli. Parallelisation and queuing management are controlled by splitting the workload into individual threads/event queues, and by moving agents from one to the other by working out the most efficient execution flow while preserving access to critical resources and attributes between threads. Sensor messaging is built upon the system of Qt's signals and slots, a publish/subscribe mechanism where objects emit notifications that are 'listened to' by other objects in the environment. Figure 2 shows a comprehensive diagram of the architecture implemented.

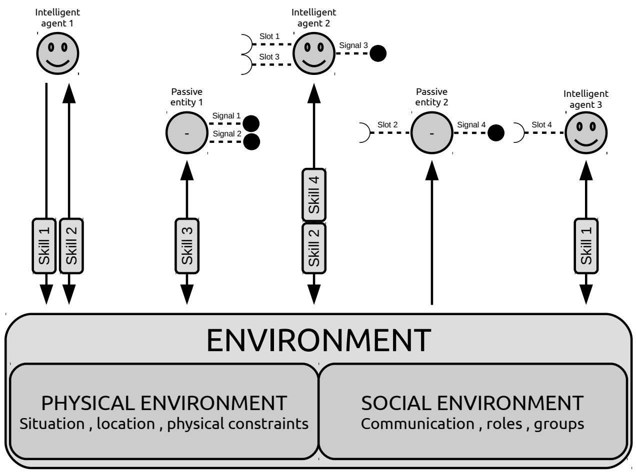


Fig. 2: Architecture of the environment and agents.

2.2 FI-WARE Middleware

GeoWorldSim relies on FI-WARE as the central middleware for creating a link between the monitored devices and the core of the demand-response control. FI-WARE is a new cloud infrastructure created by the European Commission and several major European ICT companies for the development and global deployment of services and applications of the Future Internet [14]. FI-WARE is based on a set of modules, known as General Enablers (GE), which provide services such as shared cloud computing resources, big data analysis, data management, modules for integrating web applications to the FI-WARE infrastructure, and security control.

Figure 3 depicts a general overview of FI-WARE's architecture, with the central core of the solution being the Orion Context Broker GE. This GE is a C++ implementation of the Publish/Subscribe mechanism, providing interfaces to the NGSI9 and NGSI10 API's with services such as registering context producer applications (e.g. sensor humidity within a room), update control information (e.g. send updates about changes in humidity), subscription and notification on changes on

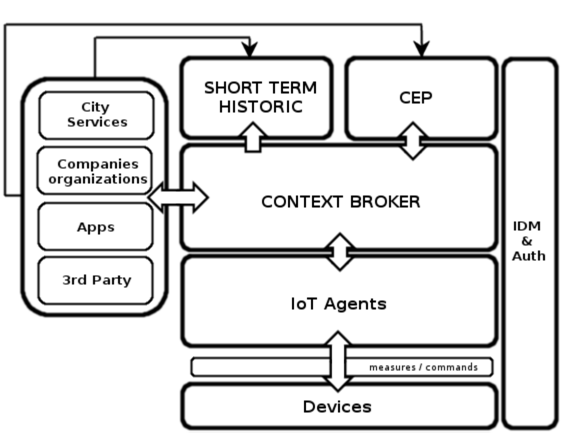


Fig. 3: Diagram with the overall architecture of FI-WARE [14]

context information (e.g. when the humidity changes) or with a given frequency (e.g. retrieve humidity values every hour), and query context information about registered applications.

The Context Broker has the ability of managing Context Information at large scale by keeping virtual representations of physical devices. The interaction with the devices takes place on the event of modifying or updating their virtual representations in the Context Broker. When a petition of registration or update is performed, the APIs map each device to a specific Context Provider in order to retrieve all the necessary information through a Uniform Resource Identification protocol. Device types are grouped in domains according to their hierarchical schema and queries can be done in less or more deepness of the hierarchy. Section 2.2.1 describes in detail this hierarchical representation and the messages needed for communication.

According to FI-WARE documentation, the proper way to connect any device to the platform is through the use of an entity called IoTAgent which bridges system specific communication with FI-WARE middleware, regardless of the protocol and data representation. Specifically, in our system, we have implemented a particular IoTAgent in charge of binding the REST API synchronous calls of the Context Broker to the asynchronous and non-blocking Qt Signal and Slots communication architecture.

2.2.1 Messaging format

Each FI-WARE message must contain a minimum set of properties for the Context Broker to be able to identify the service, domain and device the message belongs to. Therefore, the smallest piece of information to be sent will on average be of 180 bytes.

Additionally, messages can describe other entity attributes, values and control commands. Each attribute must contain a name and type of value. An example of such control command can be seen in Source Codes 1–3.

Source Code 1: Header example.

```
{
  "content-type": "application/json",
  "X-Auth-Token": "[TOKEN]",
  "Fiware-Service": "HouseManager",
  "Fiware - ServicePath ": " / "
}
```

Source Code 2: Payload example.

```
{
  "devices": [{
      "device_id": "[DEV_ID]",
      "entity_name": "[ENTITY_ID]",
      "entity_type": "thing",
      "protocol": "IoTA-UL",
      "timezone": "Europe/Madrid",
  }]
}
```

Entities are registered in FI-WARE Context Broker using HTTP POST operations and sending the entire headers and payload description. Updates, however, are reported using a HTTP PATCH operation and just sending the headers together with the attributes that need to be updated, reducing the size of the bandwidth needed.

3 Implementation of GeoWorldSim

GeoWorldSim is a Smart City Agent Based Simulator developed for analysing how demand-response strategies can influence human behaviour in order to achieve an optimal operation of the electrical grid. To this end, the main interest lies in the proper characterisation of the energy consumption of appliances within homes, specifically those which usage can be timetabled along the day, such as washing machines or dishwashers. However, the great majority of the publicly available energy consumption datasets provide hourly data about the aggregated load per energy meter. Information detailing the specifics of the interaction between the house

Source Code 3: Control message example.

```
{
  {
    "device_id":"[DEV_ID]",
    "entity_name": "[ENTITY_ID]",
    "entity_type": "thing",
    "timezone": "Europe/Paris",
    "endpoint": "http://{{DEV_IP}}:{{PORT}}",
    "attributes": [
        {
          "object_id": "t",
          "name": "temperature",
          "type": "int"
        }
      ],
    "commands": [
        {
          "name": "ping",
          "type": "command",
          "value": "[Dev_ID]@ping|%s"
        },
      ]
  }
}
```

occupants and the house appliances is not always available. Even so, the datasets that detail appliances usage only comply with the monitoring of a certain amount of buildings over a limited number of days. In order to replicate these datasets for a whole city, the *Block Bootstrap* methodology [25] could be used by randomly resampling data from the original datasets. However, this data would be heavily dependant on the season in which the appliances where monitored, failing to characterise seasonal loads, i.e. loads that take place in winter against loads that take place in summer.

The need to go further in the modelling of appliances and human behaviour called for the use of a Multi-Agent System (MAS). MAS are computer systems composed by intelligent pieces of software called agents that emulate the behaviour of an entity [37]. Agents perceive their environment, process perceptions, respond, and act rationally like the entity they represent. All of the agents are executed in parallel, either cooperating or competing against each other to achieve a certain goal. In this way, these systems can create networks of agents working together to address complex issues. The use of MAS involves readjusting the traditional centralised governing paradigm to a bottom-up dynamical model where every entity that lives in the simulation is able to make its own decisions. In this sense, we differentiate between agents and passive entities. In the first case, agents are characterised by its set of skills, methods, and a state defined by the values of their attributes. By contrast, passive entities are those which although having a *logic*, they just react to stimuli initiated by other entities.

Another important concept within MAS technologies is the Agent Environment. The Agent Environment can be considered as an independent and living first-class entity called to play a relevant role in modelling dynamic real world problems [34]. From a functionality point of view, the Agent Environment consists of both a physical and a social environment which together form the Intelligent Environment where both objects and people are situated. On the one hand, the physical environment ensures none of the spatial constraints are violated and, on the other hand, the social environment is responsible for managing communications, roles and knowledge by giving agents a social and communicative context.

Following this roadmap, GeoWorldSim defines two types of agents: humans and house appliances. Humans or house occupants are modelled as active agents capable of taking decisions and interacting with house appliances. House appliances, on their side, are modelled as passive agents which react to the interaction initiated by the occupants. The Agent Environment in particular, consists of mainly of the skills inherent to each agent and that define how each one of them operates, as well, as the communication context on which they interact.

3.1 Human Behaviour model

Human behaviour plays a pivotal role in the analysis of viable demand-response strategies, therefore a proper characterisation of how humans act is important so that the scenario studied resembles the real one as close as possible. Demand-response strategies aim at reducing the amount of energy supply that a utility must have available in order to meet the peak energy demand, thereby improving the operational efficiency of the energy grid. The reduction can be achieved by applying several scheduling mechanisms so as to modify the energy load [6] and that have a positive influence both on the energy grid and on the customer.

Humans in GeoWorldSim follow the MASSHA [22] activity definition and execution model to emulate daily duties and the consequent interaction with appliances. This model follows the well established hierarchy of behaviour, activity and action [12], [10], as seen in Figure 4. Activities are seen as sequences of actions in a time frame ranging from minutes to hours, such as preparing lunch, making coffee or washing clothes. Actions are short-timed events executed by a person that can be perceived by sensors, such as opening a door or turning on a washing machine. The behaviour helps define the way in which the person will perform the activities, in regards of ordering, inter-dependencies and priorities, within the intelligent environment.

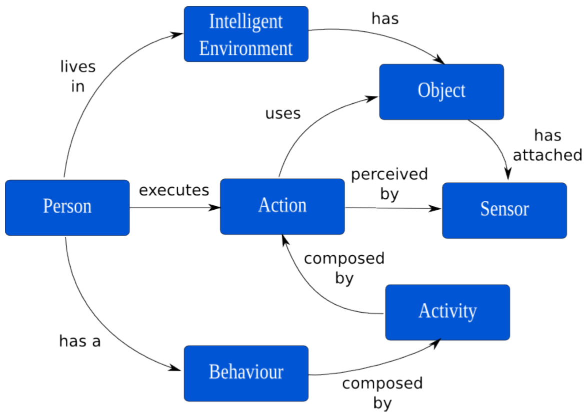


Fig. 4: MASSHA's behaviour model.

The activity definition and execution model enables to create random or fully detailed and staggered behaviours by specifying the following activity attributes:

- **Type:** Code that identifies the activity type.
- **Attendance:** Attribute for describing whether a person has to be physically present during the execution of the activity.
- **Estimated duration:** Amount of time that the activity will take.
- **Priority:** Number determining the importance an agent gives to a certain activity.
- **Desirable start time:** Indicative start time at which the person likes to start performing the activity. If possible, the agent will try to schedule the activity at this time.
- **Desirable end time:** Indicative end time at which the agent would like to have the activity finished. If possible, the agent will try to schedule the activity before this time.
- **Mandatory start time:** Compulsory time at which the activity must start. Necessary to model time critical activities such as going to work or watching a certain TV show.
- **Mandatory end time:** Compulsory time at which the activity must finish.
- **Preconditions:** Activity codes for modelling prerequisites, i.e. activities that must be already finished by the time this activity is chosen.
- **Outcomes:** Performing some activities may lead to the appearance of others, like needing to take a shower after doing exercise or toileting after having lunch.

The human behaviour model is built by defining a set of all the possible activities that people may carry out. Each activity involves the execution of a set of actions which are then directly mapped to messages published in the middleware depending on the appliances used (see Section 2.2 for more information). For example, the activity of `Prepare Breakfast` may involve actions such as `Use the Microwave` or `Open the Fridge` which will act upon the status of the attached appliances. Activities may be independent, i.e., wash the dishes, or may depend on the fulfilment of another previous activity in order to be executed, i.e. the `Dressing activity` may be executed only after the `Shower activity` has been finished. In addition, activities may be *attended*, in which the person must be present during the execution of the activity, i.e. `Mow the lawn`, or *unattended*, which allows the agent to execute several activities at the same time, i.e. `Watch TV` while `Use the Microwave`.

It is important to note, even as our model does not deal to deal with cooperative tasks, we have assigned several individuals per house as a way to simulate the real conflicting usage of appliances within a common space.

3.2 Appliances Modelling

GeoWorldSim follows the dense sensing monitoring approach [12], which relies on miniaturised simple sensors installed on objects of interest, such as appliances, urban equipment, power generation plants, etc. These sensors are able to trace invocations and can monitor boolean states (i.e. activated / deactivated) as well as continuous values, such as those given by power consumption or power generation. It is worth mentioning that objects may also have a behaviour. For example, when a person turns on an appliance (executes action `run` over an object of type *appliance*), the state of the appliance changes to `running` and thus, it can not be started again by another person (until the state of the appliance changes again to `stopped`).

Following this approach we have defined several types of appliances:

Urban equipment: These appliances refer to the devices deployed outside the houses and along public spaces, such as lamp posts and recycling containers. Generally, this type of devices is associated with a public service. For example, a waste container will issue a notification when the fill level is above a certain threshold, or a lamp post will only be functioning during specific hours.

Common Appliances: These are tools that have a constant behaviour and need to be manually started and stopped. Most of these appliances are used to carry out activities that involve being in contact with the device and not leaving it unattended, such as using an oven, a shower, or a sink faucet. Even though there are people that may overlap the use of more than one common appliance at a time (e.g. listening to the radio while having a shower or prepare a coffee without turning off the television) we will assume that all people turn off an appliance before carrying out another activity.

Countdown Appliances: These appliances have the capacity of auto stopping themselves, such as a microwave or a toilet flush. Therefore the activities involving these appliances can be parallelised with other tasks since the initiator does not have to stop them manually.

Program Appliances: Some Countdown Appliances may have different behaviours throughout the cycles that make up their scheduling program. That is the case of washing or coffee machines. This sequence of actions is emulated by scheduling the activation and deactivation of inner components and having each of them associated to a specific power consumption.

Idle Appliances: Appliances such as fridges or boilers usually remain idle, monitoring some properties to verify whether they stay under a certain threshold. Once this condition is surpassed, the appliance starts working and tries to bring that property back to the desired state.

Finally, several appliances of Program and Idle type have the *schedule* property. Appliances with this property are the ones that can be used by the demand-response controller. This attribute refers to those appliances which execution can be delayed to any other moment of the day, like for example times at which the energy grid is not saturated or the energy price is lower.

3.3 Generation/Consumption skills

In the real world, skills are the degree of competence that differentiates and empowers people to perform tasks. The same concept can be applied to agents. The skills serve both to characterise entities in the MAS and are the way to perceive or propagate actions into the agent environment and other entities. Agents have a set of skills that will help them in their decision making process. However, every skill needs to be contained within a context and behave under some restrictions. We believe that is the task of the Agent Environment to hold the responsibility of ensuring that these constraints are met and limit the capabilities of each entity [30].

In this way, each appliance of GeoWorldSim is given a set of skills upon creation that define its main core functionality. Taking a `Fridge` for example, this device is given a `Temperature` skill with methods to retrieve or change said temperature value. In addition, we can define skills that act upon skills, such as `Heating skill` or a `Cooling skill` which can modify the `Temperature skill` by gradually simulating the increase or decrease in temperature over time.

Considering the demand-response scenario where power demand monitoring is the main pillar of this strategy, the set of skills assigned to an agent will not only modify the attributes of the agent, but also implicitly demand an energy load. In the `Fridge` example, the execution of a `Heating` or a `Cooling` skill will not only modify the temperature inside the `Fridge`, but will also generate a load, i.e. the energy needed in order to perform such change in temperature. The notation of the method is common to all consuming skills in which each device is supplied power and, depending on how the appliance works and what skills are defined, demands a specific amount of power.

4 Methods

In order to assess the advantages and drawbacks of several architectures in regards of implementing a demand-response solution in a Ubiquitous World, at city level, we propose the analysis of two middleware architectures (Centralised and Federated) and two publication strategies (Periodical and Event-based):

Centralised Architecture: As seen in Figure 5, this architecture consists of a single Context Broker. This configuration provides an easy management and a clear monetising of the services provided. However, since all the users share the same Context Broker instance, the bandwidth requirements are very high, as seen in Section 5, the resilience to failures decreases significantly, and the process to guarantee privacy to citizen data grows in complexity.

Fig. 5: Centralised architecture schema

Federated Architecture: In this architecture, each house has its own Federated Context Broker instance (FCB), that in turn is connected to another Context Broker instance on a higher hierarchical level. The low level Context Broker stores and aggregates data from the house's appliances and publishes them upstream. It also transmits other events relevant to the demand-response strategy, such as the need to schedule loads that can be *delayed*, like the operation of washing machines, dishwashers or laundry driers. Urban equipment like streetlights and waste containers publish to their own private low level Context Broker which also sends the data to the high level Context Broker. Figure 6 shows an example of the Federation Architecture where several Context Brokers form a hierarchy. Our results suggest that the use of a single high level Context Broker gives solutions to several of the drawbacks of the Centralised Architecture. For example, a) the privacy of citizen data is better preserved since there is a less amount of data travelling outside the home network [33]; b) the architecture is more resilient: if one Context Broker malfunctions, the rest of the network can continue operating normally; c) the bandwidth need is several orders of magnitude lower. From a service provider company, however, this system is less attractive in terms of exploitation, due to the complexity of its management.

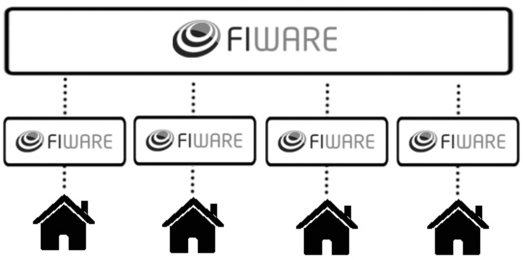


Fig. 6: Federation architecture schema

Periodical Publication Strategy: In this strategy, all sensors publish their state or consumption over a fixed period of time. This strategy makes the development of demand-response controls simpler and similar to real time monitoring. However, the energy consumption of the devices and the bandwidth requirements increase significantly and become unmanageable for large sets of houses or appliances.

Event based Publication Strategy: In this strategy, the sensors only publish data when their state changes (e.g. switched on/off, opened/closed, etc). When a sensor is switched off, in addition to notifying this change, it also provides information about the power consumption occurred during the time the associated appliance was operative. This approach solves the drawbacks of the periodic publication strategy but increases the complexity of the demand-response control since these sensor events need to be mapped to particular consumption patterns.

We also identified three future scenarios that pave the way to a fully Ubiquitous World on which to test the previous architectures and publication strategies.

Smart City (SC): In this scenario we will simulate the traditional vision envisaged by the managers of the myriad of distribution grids that feed a city. In this sense, houses periodically provide information about the aggregated load (whether it is water, energy or amount of waste) for billing purposes. The publication frequency is fairly low, which makes it feasible to connect the publication feed to a single centralised Context Broker. The federation architecture has not been tested in this scenario, since it would be an over-engineering example. Figure 7 shows a graphical representation of this scenario.

IoT Jungle (IOTJ): Within this scenario, all the IoT devices within a house publish information to a centralised Context Broker. This publication can be performed both periodically or event based triggered by a change in the state of the device. Depending on the publishing method, the bandwidth needs and size of published message may vary since, for example, event based messages need to send bigger pieces of information. Figure 8 shows a graphical representation of this scenario.

Ubiquitous World (UW): This scenario combines the advantages of the SC and IOTJ scenarios, a vision envisaged by IoT researchers. In this sense, all the devices within a house are connected to a private FCB. This Context Broker is in charge of storing and aggregating all the data received from the devices. This aggregation is then published to a centralised Context Broker. Urban equipment also publish to their own FCB. Each FCB, in addition, publishes other events relevant to the demand-response strategy, such as the need to schedule loads that can be *delayed*. Figure 9 shows a graphical representation of this scenario. This architecture is fully aligned with the *Edge Computing* paradigm [35] in which services are moved from centralised nodes to the extremes of the network where the information is generated.

The aforementioned scenarios only cover the Periodical Publication Strategy for the first case. The reason is that the amount of messages increases with the number of houses and quickly flow the network driver. Simple scenarios could require more than $250\,\mathrm{MB\,s^{-1}}$ if the publication period is short. Obviously, if the publication period is too large, this strategy loses its purpose.

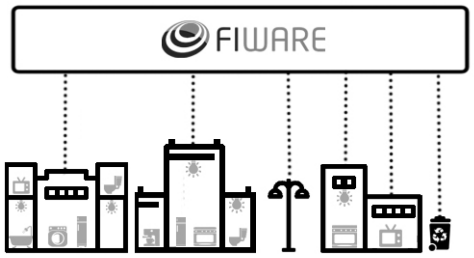


Fig. 7: Diagram of the Smart City scenario

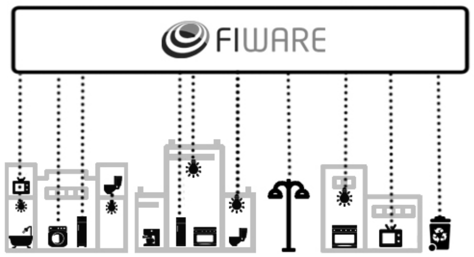


Fig. 8: Diagram of the IoT Jungle scenario

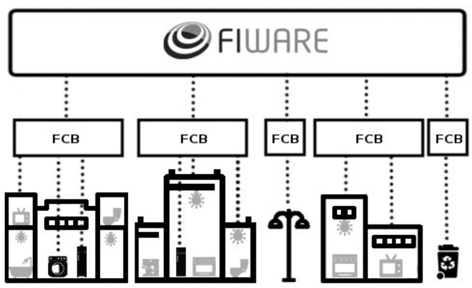


Fig. 9: Diagram of the Ubiquitous World scenario

5 Results

The simulation setup is composed of an Acer TravelMatep238 laptop Core i5 2.3 GHz with 16GB of RAM using a Gigabit Ethernet class connection to reach the cloud server [16] (where GeoWorldSim is being run) and the FI-WARE-LAB cloud environment [18] (where the middleware is deployed). The FI-WARE-LAB cloud environment provided a working production instance of the FI-WARE stack without any Service Level Agreement. As in any other cloud environment, the virtual instance is under our control (namely, we can guarantee that no one else is using that instance) but we share virtual machines with several other users (we cannot guarantee the amount of bandwidth at our disposal nor the overall load of the physical server). In order to asses the different middleware architectures, we defined two Key Performance Indicators (KPI) of interest:

Bandwidth: number of bytes transmitted per unit of time. In this particular example, we have recorded the number of bytes per message without including the overheads related to the specific protocols used to transport the information. Our objective is to estimate the real bandwidth needs in a real deployment without assuming a simultaneity factor. Note that this value is specially relevant in the Smart City scenario, as the deployment is traditionally performed over SIGFOX or LoRa networks that have limited bandwidth [1].

Number of Messages Lost: number of messages emitted by GeoWorldSim that are not processed by the Context Broker because they may get lost during the transmission. This value gives information about the quality of the service provided by the architecture. In our experiments, none of the transmitted messages were lost due to the fact that the Context Broker uses an asynchronous database storage, and therefore, during high load peaks, message insertion gets asynchronously scheduled. This is a good indicator of the reliability of the solution.

In order to accelerate the procurement of results and balance both the length of the configuration file and RAM consumption, the simulation time was accelerated. We can define that 1 second of the simulation corresponds to n second in terms of real time. n have been taken accordingly to ease the deployment of the simulation. Modification in n has a direct impact in the relation between the measured KPIs and the real KPIs (i.e. if $n = 1$ have been used). In this sense, an speedup factor of n would be equivalent to:

- multiply by n the number of houses
- divide by n the bandwidth

Several city sizes were tested for each of the scenarios defined in 4, specifically, Village (250 houses), Small City (2500 houses), Medium City (25 000 houses), Large City (250 000 houses) and Megalopolis (2 500 000 houses). Each house accommodates two persons that interact with 18 smart devices while performing their daily duties. In addition, the scenarios were provided with ($5 \times$ #number of houses) streetlights and ($1 \times$ #number of houses) recycling containers.

The simulation has been carried out for 6 hours, the busiest of the day, that is 3 in the morning and 3 in the evening:

Morning: This portion of the simulation covers the time ranging from the moment the citizens wake up to the moment they go to work. The appliances activated are those related to the activities that involve getting ready for going outside (showering, dressing, grooming), and preparing breakfast. The number of appliances used is rather low but all of them are used almost at the same time.

Evening: This portion of the simulation covers the period of time between having dinner and going to bed. This period includes the prime time. The appliances activated range from the ones typically used in the kitchen to the ones used for entertainment. In this time period, the number of appliances used is high and diverse but their activation is more spread over time.

Since each simulation run is done over a long period of time, we consider that there is no need to repeat the experimentation due to the fact that all the possible variability that may appear will probably be already captured. A video of a complete simulation can be found in `http://bit.ly/2fbX54p`.

For each simulation we register two logs: the timestamp and types of message sent by GeoWorldSim and the equivalent log from the FI-WARE Context Broker so as to easily extract the KPIs values. For example, Figure 10 shows a graphical representation of the number of messages per device transmitted during the simulation time. The bandwidth is measured by multiplying the number of messages by their length and dividing the result by the unit of time.

Figure 11 shows several boxplots [11] depicting the bandwidth distribution used per number of houses in the Smart City scenario. Please note that this is a log-log plot. As can be seen, the bandwidth grows and its saturation takes place at the *Medium City* level, with a saturation bandwidth around 1 MB s^{-1}. Take into consideration that this scenario is the only one where the amount of messages allows to perform a simulation of the Megalopolis. Clearly, even though we were able to perform this simulation, the computational needs superseded the bandwidth restrictions of this case.

The IoT Jungle represents the most problematic scenario. Figure 12 shows, as in the previous case, several boxplots describing the distribution of the bandwidth used per size of the city. As can be seen, the saturation is reached under the same case *Medium City*, but it uses 10 times more bandwidth than the Smart City scenario. However, in this case, it seems that the bandwidth demands of the *Large City* have produces a change of phase and the bandwidth collapses [4].

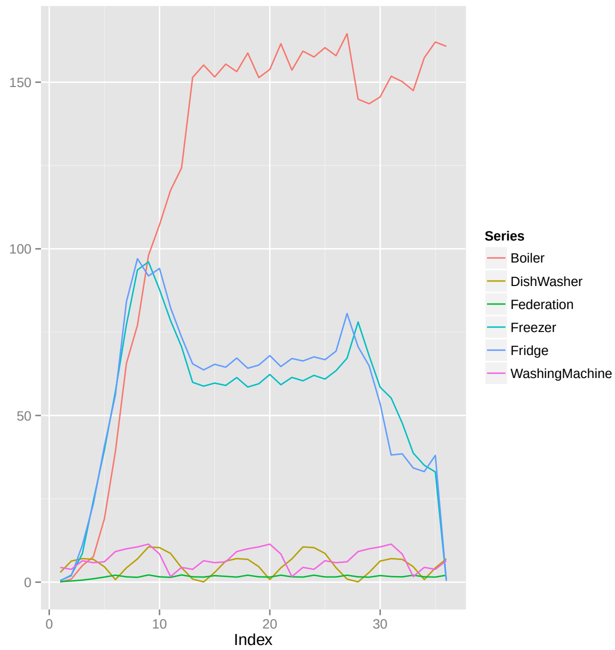


Fig. 10: Evolution of the number and type of messages per time in a simulation.

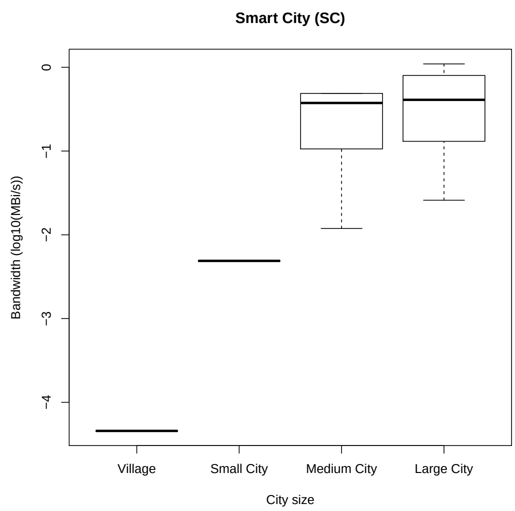


Fig. 11: Bandwidth distribution per number of houses in the Smart City scenario.

Federation architectures need to have several Context Brokers operating simultaneously. We only have access to a single instance of a Context Broker, therefore we cannot directly asses this architecture. However, we can perform the simulation in two steps. First, we simulate the behaviour on a node on the federation (a house)

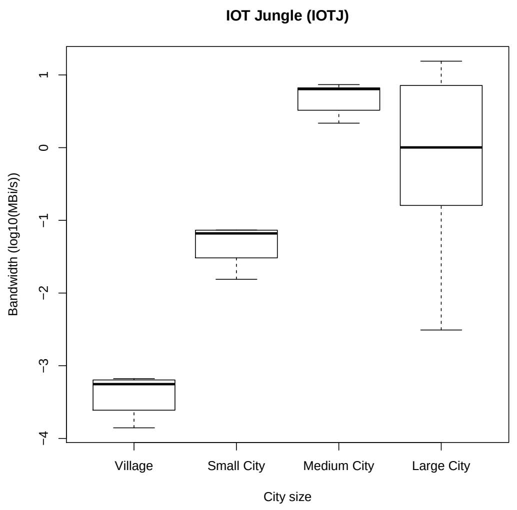


Fig. 12: Bandwidth used per number of houses in the IoT Jungle scenario.

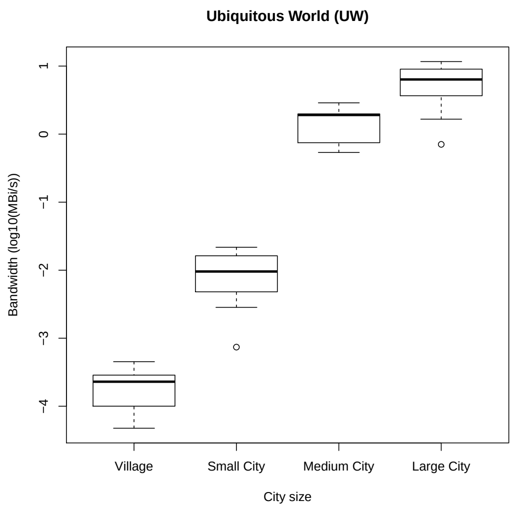


Fig. 13: Bandwidth used per number of houses in the Ubiquitous World scenario

and register all the messages that should have been sent to the federation. Then we repeat this process for every node. Finally, we replay the message record from the nodes to the centralized Context Broker. Figure 13 shows the bandwidth used per number of houses in the Ubiquitous World scenario. In this case the saturation takes place in the *Large City* case, in contrast with the previous two scenarios.

Table 1: Results of the Kruskal-Wallis with a Conover post hoc statistical test.

Test result	Village	Small City	Medium City	Large City
Kruskal-Wallis	1.4×10^{-6}	0.0037	9.4×10^{-8}	0.000 93
SC-IOTJ	2.0×10^{-11}	0.000 87	2×10^{-16}	**0.064**
SC-UW	1.1×10^{-7}	**0.11**	1.5×10^{-9}	0.000 17
IOTJ-UW	0.0019	0.011	1.1×10^{-8}	0.028

6 Conclusions and next steps

Results suggest that the most suitable architecture to carry on demand-response controls at city level is the event based Federation (Ubiquitous World Scenario). Even though the control strategies to be built will be more complex, the bandwidth demand is several orders of magnitude lower and the overall architecture is both more secure and resilient.

Figure 14 shows the boxplots comparing the bandwidth needs for the three architectures in the four scenarios. Table 1 shows the p-values of a Kruskal-Wallis test with a Conover post hoc [31]. Several conclusions can be drawn out from this information:

– The bandwidth needs for the Smart City scenario are statistically significant lower than those of the other architectures in almost all the scenarios but in turn, this architecture provides less information.
– The bandwidth needs for the IoT Jungle are statistically significant higher than those of the other architectures in almost all scenarios but the amount of information that this architecture provide is similar to that provided by the Ubiquitous World architecture.

The next step towards the fulfilment of GeoWorlSim is in test the different middlewares currently available [39]. In this sense, we will develop several of the services needed to perform demand-response in different middlewares and perform a qualitative and quantitative analysis to assess its suitability.

Then, an adequate simulation of the Smart Grid goes hand in hand with the precise definition of how the electrical system works and the implementation of several new agent types: producers of energy, storage and control nodes. Up to this point, we have used simplistic load models, since the objective of the study is the resiliency of the architecture over a great number of devices publishing power consumption. The future steps in this area involve introducing a realistic model of the different grids to be able to simulate different smart controllers. This smart energy controllers will listen to what the energy needs of devices within a house, leveraging demand with energy generation coming from Photovoltaic panels,

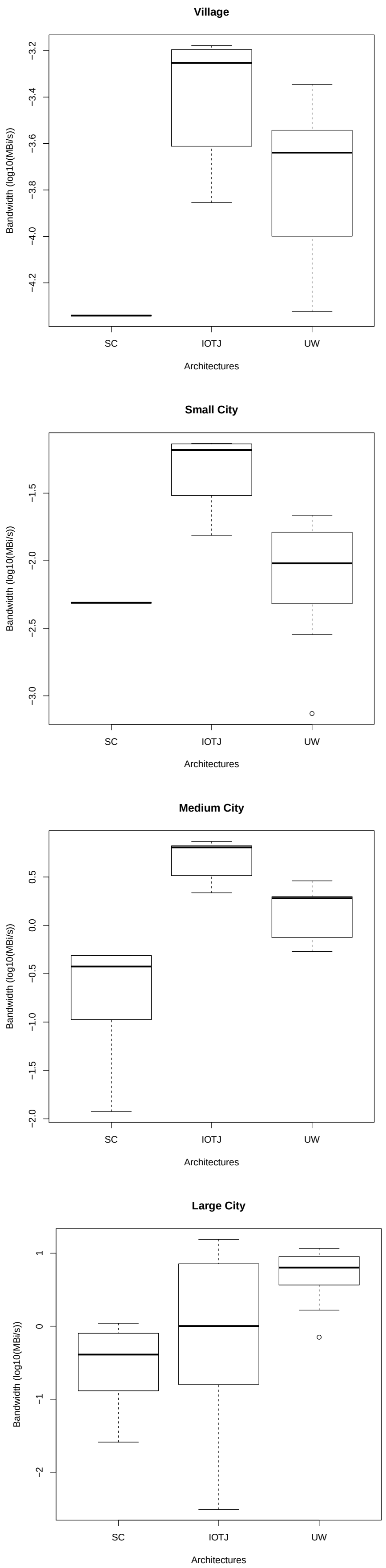


Fig. 14: Comparison of the bandwidth needs of the architectures per scenario.

batteries or even the distribution network. Moreover, simple models of both the new agents and the electrical distribution system could be implemented directly in GeoWorlSim but more complex ones would require being coupled via specialised tools like Matlab. To this end, probably, the best strategy would be to follow the same as in this article, namely to implement an IoTAgent in Matlab and run both systems in parallel [23].

Finally, another line of future work has to do with the development a full human behaviour model that dictates the execution of activities of every agent in the simulation platform. Now, it heavily relies on a previous definition of a list of possible activities in order to schedule the temporal distribution. The next iteration in this area will test the use of Hidden Markov Models, Probabilistic Finite State Machines, Neural Networks or Support Vector Machines not only to model an adequate temporal distribution of activities but also to be able to predict the behaviour change when a persuasive methodology is used [8]. Moreover, given that the communication systems is actually working in a hardware-in-the-loop configuration, this simulator could be used by a real demand-response controller to evaluate the impact produced by and specific control decision.

Acknowledgements This work was carried out with the financial support of a) the European Union's Horizon 2020 Research and Innovation programme under grant agreement No. 696129, given to GREENSOUL project b) Industrial Ph.D. grant given by the University of Deusto (2015- 2018) and c) Ph.D. grant PRE_2015_2_0003 given by the Basque Government.

References

1. Adelantado F, Vilajosana X, Tuset-Peiró P, Martínez B, Melià J (2016) Understanding the limits of lorawan. CoRR abs/1607.08011, URL `http://arxiv.org/abs/1607.08011`
2. Agency EE (2015) Air quality in europe 2015 report. EEA Report 5
3. Agency IE (2015) Energy climate and change. World Energy Outlook Special Report
4. Al-Bahadili H (2012) Simulation in Computer Network Design and Modeling: Use and Analysis. IGI Global
5. Albadi MH, El-Saadany E (2008) A summary of demand response in electricity markets. Electric power systems research 78(11):1989–1996
6. Allcott H, Mullainathan S (2010) Behavior and energy policy. Science 327(5970):1204–1205
7. Apostolou G, Krinidis S, Ioannidis D, Tzovaras D, Borges CE, Casado-Mansilla D, de Ipiña DL (2016) Greensoul; a novel platform for the reduction

of energy consumption in communal and shared spaces. In: 2016 4th International Symposium on Environmental Friendly Energies and Applications (EFEA), pp 1–6, DOI 10.1109/EFEA.2016.7748783
8. Apostolou G, Krinidis S, Ioannidis D, Tzovaras D, Borges CE, Casado-Mansilla D, Lopez-de Ipina D (2016) Eco-aware persuasive networked data devices for user engagement in energy efficiency: The greensoul concept. In: In Proceedings of the 4th International Symposium on Environment-Friendly Energies and Applications
9. Lopez-de Armentia J, Casado-Mansilla D, Lopez-de Ipia D (2013) Saving Energy through Collaborative Eco-aware Everyday Things. In: The Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IEEE Xplore, Taichung, Taiwan, pp 489 – 493, DOI 10.1109/IMIS.2013.88, URL http://ieee.164288.com/xpl/articleDetails.jsp?reload=true&arnumber=6603721
10. Azkune G, Almeida A, López-de Ipiña D, Chen L (2015) Combining Users' Activity Survey and Simulators to Evaluate Human Activity Recognition Systems. Sensors 15(4):8192–8213, DOI 10.3390/s150408192, URL http://www.mdpi.com/1424-8220/15/4/8192
11. Benjamini Y (1988) Opening the box of a boxplot. The American Statistician 42(4):257–262, URL http://www.jstor.org/stable/2685133
12. Chen L, Hoey J, Nugent C, Cook D, Yu Z (2012) Sensor-based activity recognition. IEEE Transactions on Systems, Man, and Cybernetics-Part C 42(6):790–808, DOI 10.1109/TSMCC.2012.2198883, URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6208895
13. Commision E (2015) A framework strategy for a Resilient Energy Union with a Forward-Looking Climate Change Policy. Communication from the Commision to the European Parliament, the Council and the Comitte of the Regions
14. Consortium FW (2012–2016) FI-WARE architecture. http://cordis.europa.eu/fp7/ict/netinnovation/deliverables/fi-ware/fi-ware-d231b.pdf
15. Cook DJ, Youngblood M, Heierman III EO, Gopalratnam K, Rao S, Litvin A, Khawaja F (2003) MavHome: an agent-based smart home. In: null, IEEE, p 521
16. Education BG (2015–2016) I2Basque Network. http://www.i2basque.es/
17. Espinoza A, Penya Y, Nieves JC, Ortega M, Pea A, Rodrguez D (2013) Supporting business workflows in smart grids: An intelligent nodes-based approach. IEEE Transactions on Industrial Informatics 9(3):1384–1397, DOI 10.1109/TII.2013.2256792
18. FI-WARE (2015–2016) Cloud FI-WARE. https://account.lab.fiware.org/
19. Fornié MAS, Sainz-Maza RG (2013) Desarrollos de smart grids para el fomento de la eficiencia energética. In: Anales de mecánica y electricidad, Asociacion de Ingenieros del ICAI, vol 90, pp 62–67
20. Helal S, Mann W, El-Zabadani H, King J, Kaddoura Y, Jansen E (2005) The Gator Tech Smart House: a programmable pervasive space. Computer 38(3):50–60, DOI 10.1109/MC.2005.107
21. Higuchi N, Xu Y, Yamamoto T, Nara K (2014) Functions and operational experiences of real scale smart grid experimental system. In: 2014 IEEE Innovative Smart Grid Technologies-Asia (ISGT ASIA), IEEE, pp 624–629
22. Kamara-Esteban O, GAzkune, Pijoan A, Borges CE, de Ipiña DL (2016) MASSHA: An agent-based approach for human activity simulation in intelligent environments, submitted
23. Kamara-Esteban O, Sorrosal G, Pijoan A, Castillo-Calzadilla T, Iriarte-Lopez X, Macarulla-Arenaza AM, Martin C, Alonso-Vicario A, Borges CE (2016) Bridging the gap between real and simulated environments: an agent-based hybrid smart home simulator for complex systems. In: Proceedings of the Intl IEEE Conferences on Ubiquitous Intelligence & Computing, pp 220–227
24. Karnouskos S, De Holanda TN (2009) Simulation of a smart grid city with software agents. In: Computer Modeling and Simulation, 2009. EMS'09. Third UKSim European Symposium on, IEEE, pp 424–429
25. Kunsch HR (1989) The jackknife and the bootstrap for general stationary observations. The Annals of Statistics pp 1217–1241
26. Lesser V, Atighetchi M, Benyo B, Horling B, Raja A, Vincent R, Wagner T, Xuan P, Zhang SX (1999) The UMASS intelligent home project. In: Proceedings of the third annual conference on Autonomous Agents, ACM, pp 291–298
27. Oliveira P, Pinto T, Morais H, Vale Z (2012) Masgrip: A multi-agent smart grid simulation platform. In: Power and Energy Society General Meeting, 2012 IEEE, pp 1–8, DOI 10.1109/PESGM.2012.6345649
28. Penya YK, Nieves JC, Espinoza A, Borges CE, Peña A, Ortega M (2012) Distributed semantic architecture for smart grids. Energies 5(11):4824–4843
29. Penya YK, Borges CE, Pea A, Esteban OK (2013) Service-orientation vs. real-time: Integrating smart-homes into the smart-grid. In: Industrial Electronics Society, IECON 2013 - 39th Annual Conference of the IEEE, pp 5755–5760, DOI 10.1109/IECON.2013.

6700077

30. Pijoan A, Kamara-Esteban O, Borges CE (2015) Environment modelling for spatial load forecasting. In: Weyns D, Michel F (eds) Agent Environments for Multi-Agent Systems IV, Lecture Notes in Computer Science, vol 9068, Springer International Publishing, pp 188–206, DOI 10.1007/978-3-319-23850-0_12
31. Pohlert T (2014) The pairwise multiple comparison of mean ranks package (pmcmr). R package
32. Project Q (1995–2016) Qt | Cross-platform application & UI development framework. http://www.qt.io/
33. Response SS (2016) IoT devices being increasingly used for DDoS attacks. https://www.symantec.com/connect/blogs/iot-devices-being-increasingly-used-ddos-attacks
34. Ricci A, Rodriguez-Aguilar J, Pijoan A, Zambonelli F (2015) Mixed environments for mas: Bringing humans in the loop. In: Weyns D, Michel F (eds) Agent Environments for Multi-Agent Systems IV, Lecture Notes in Computer Science, vol 9068, Springer International Publishing, pp 52–60, DOI 10.1007/978-3-319-23850-0_4, URL http://dx.doi.org/10.1007/978-3-319-23850-0_4
35. Skala K, Davidovic D, Afgan E, Sovic I, Sojat Z (2015) Scalable distributed computing hierarchy: Cloud, fog and dew computing. Open Journal of Cloud Computing (OJCC) 2(1):16–24, DOI 10.19210/1002.2.1.16, URL http://www.ronpub.com/publications/OJCC_2015v2i1n03_Skala.pdf
36. Van Nguyen T, Kim JG, Choi D (2009) ISS: the interactive smart home simulator. In: Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on, IEEE, vol 3, pp 1828–1833
37. Wooldridge M, Jennings NR (1995) Intelligent agents: theory and practice. The Knowledge Engineering Review 10:115–152, DOI 10.1017/S0269888900008122
38. Xia F, Yang LT, Wang L, Vinel A (2012) Internet of things. International Journal of Communication Systems 25(9):1101
39. Zdravković M, Trajanović M, Sarraipa J, Jardim-Gonçalves R, Lezoche M, Aubry A, Panetto H (2016) Survey of Internet-of-Things platforms. In: 6th International Conference on Information Society and Techology, ICIST 2016, Kopaonik, Serbia, vol 1, pp 216–220, URL https://hal.archives-ouvertes.fr/hal-01298141, iSBN: 978-86-85525-18-6