FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

A Multiagent Architecture to Support the Implementation of Sustainable Households

Sofia Cardoso Martins

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Rosaldo J. F. Rossetti

June 31, 2021

A Multiagent Architecture to Support the Implementation of Sustainable Households

Sofia Cardoso Martins

Mestrado Integrado em Engenharia Informática e Computação

June 31, 2021

Abstract

The continuous growth in world population, ever since the 19th century, has given rise to a joint international effort to ensure the sustainable management of resources. Given the substantial role that the urban residential sector plays in consuming energy and water resources, household-oriented solutions can drive forward sustainable consumption patterns at the city level. Today, the pervasiveness and increased affordability of smart devices composing the Internet of Things (IoT) motivate the development of Home Automation Systems (HAS) that can contribute to that effect. Yet, despite their potential, most HASs fail to address four fundamental requirements altogether: (1) easy adaptability and extensibility to new services (2) efficient access to remote services, particularly for data acquisition (3) interoperability between heterogeneous devices employing various communication standards and protocols (4) software portability, encompassing the distribution of software artifacts and offloading intensive computation tasks.

Based on the challenges posed to current HASs, the present work aimed at designing and implementing a general-purpose HAS to favour comfort and sustainable resource management. The layered and modular architecture relies on a Multiagent layer for the autonomous management and control of home appliances. In turn, the FIWARE IoT middleware guarantees interoperability and provides services to the upper layer, namely in terms of data access, command flow and subscription setup. To assess the potential of the HAS architecture for sustainability and comfort, this work proceeded with the specification of a Home Energy Management System (HEMS), which relied on three models. The first model depicts the Multiagent System (MAS) from a resource-based integrated market perspective, describing the resources, markets, agents and regulation policies according to the ResMAS model. It is followed by an AgentSpeak(L) model that specifies the beliefs, goals and plans of each cognitive agent class. The third model integrates various Unified Modeling Language (UML) diagrams that describe the implementation of the two other models as software artifacts. As a result of this modelling process, the HEMS system was implemented in Jadex and deployed together with FIWARE components, emulated sensor and actuator devices. Finally, a realistic case study was specified and simulated under this architecture.

The simulations conducted allowed for assessing the influence of different regulation policies (Green, Bill and Comfort), grid tariffs and photovoltaic panel capacities on the scheduling of smart appliances. In particular, the simulation process resulted in a comprehensive analysis of the effects on total daily energy consumption, exploitation of energy generated by a photovoltaic panel, grid dependency and daily expenses. Interestingly, these findings suggest that the system can coordinate the activity of the appliances so as to attend to homeowners' needs, whether they are focused on reducing the electricity bill, minimising dependency on the grid or augmenting overall comfort. In addition, the multiagent system provides mechanisms to efficiently react to unforeseen events that may render unfeasible the planned scheduling of appliances.

Taken together, the results of this work suggest that the HAS has the potential to aid homeowners in decision making, raise awareness as of their resource consumption profiles, and provoke behavioural changes leading to more sustainable consumption patterns. In addition, its modular architecture, based on an IoT middleware, enables portability, adaptability, scalability, connectivity and interoperability, all of which are features that strengthen this architecture and may accelerate the development of this type of system.

Keywords: Internet of Things, Multiagent System, Home Automation System, Home Energy Management System, Demand Side Management

Resumo

O crescimento contínuo da população mundial, desde o século XIX, deu origem a um esforço internacional conjunto para assegurar a gestão sustentável dos recursos. Dado o papel substancial que o sector residencial urbano desempenha no consumo de energia e recursos hídricos, decorre que as soluções orientadas para o consumo doméstico têm o potencial de impulsionar padrões de consumo sustentáveis a nível da cidade. Hoje em dia, a omnipresença e o aumento da acessibilidade dos dispositivos inteligentes que compõem a Internet das Coisas (IoT) motivam o desenvolvimento de sistemas de Domótica que possam contribuir para esse efeito. No entanto, apesar do seu potencial, a maioria destes sistemas não consegue abordar simultaneamente quatro requisitos fundamentais: (1) fácil adaptabilidade e extensibilidade a novos serviços (2) acesso eficiente a serviços remotos, particularmente para aquisição de dados (3) interoperabilidade entre dispositivos heterogéneos que empregam várias normas e protocolos de comunicação (4) portabilidade de *software*, englobando a distribuição de artefactos de *software* e o *offloading* de tarefas de computação intensiva.

Com base nos desafios colocados aos atuais sistemas de Domótica, o presente trabalho visou a concepção e implementação de um sistema de Domótica generalizado com vista a favorecer o conforto e a gestão sustentável de recursos. A sua arquitectura em camadas e modular assenta numa camada multiagente para a gestão e controlo autónomo dos eletrodomésticos. Por sua vez, o *middleware* de IoT FIWARE garante a interoperabilidade e assegura serviços à camada superior, nomeadamente em termos de acesso a dados, fluxo de comandos e configuração de subscrições. Para estudar o potencial de sustentabilidade e conforto desta arquitetura, este trabalho prosseguiu com a especificação de um modelo de um Sistema de Gestão Residencial de Energia segundo de três modelos. O primeiro modelo retrata o Sistema Multiagente (SMA) segundo uma perspectiva de mercado integrado baseado em recursos, descrevendo assim os recursos, mercados, agentes e políticas reguladoras de acordo com o modelo ResMAS. É seguido por um modelo AgentSpeak(L) que especifica as crenças, objectivos e planos de cada classe de agentes cognitivos. O terceiro modelo integra vários diagramas da *Unified Modeling Language* (UML) que descrevem a implementação dos restantes modelos como artefactos de software. Como resultado deste processo de modelação, o sistema foi implementado em Jadex e instalado juntamente com componentes FIWARE, sensores e actuadores emulados. Finalmente, um caso de estudo realista foi especificado e simulado sob esta arquitectura.

As simulações realizadas permitiram avaliar a influência de diferentes políticas de regulação (*Green*, *Bill* e *Comfort*), tarifas do fornecedor de energia e capacidades de painéis fotovoltaicos no escalonamento de aparelhos inteligentes. Em particular, o processo de simulação resultou na análise objetiva de efeitos sobre o consumo diário total de energia, exploração da energia gerada por um painel fotovoltaico, dependência da rede e despesas diárias. Curiosamente, os resultados deste estudo revelam que o sistema consegue coordenar a actividade dos aparelhos de modo a atender às necessidades dos proprietários, quer essas se concentrem na redução da conta de electricidade, na minimização da dependência da rede ou no aumento do conforto geral. Além disso, o Sistema Multiagente fornece mecanismos para reagir eficazmente a eventos imprevistos que possam inviabilizar o escalonamento de aparelhos previamente projetado.

No seu conjunto, os resultados deste trabalho sugerem que a arquitetura de sistema de Domótica proposta tem o potencial de ajudar o proprietários das casas na tomada de decisões, sensibilizá-los quanto aos seus perfis de consumo de recursos, e provocar mudanças comportamentais que conduzam a padrões de consumo mais sustentáveis. Além disso, a sua arquitectura modular, baseada num *middleware* de IoT, permite a portabilidade, adaptabilidade, escalabilidade, conectividade e interoperabilidade, todas elas características que reforçam esta arquitectura e podem acelerar o desenvolvimento deste tipo de sistemas.

Keywords: Sistema Multiagente, Domótica, Sistema de Gestão Residencial de Energia, Gestão do Lado da Procura

Acknowledgements

Developing a dense project over such a long period of time is not easy, and I think that no matter how much we try to make ourselves aware of it beforehand, we are never fully prepared for what lies ahead. It is made even more difficult when a pandemic forces us to deny everything that is normal to us. Fortunately, I was lucky enough to have people by my side who supported me throughout this journey. They are friends I have known for years. They are those I have met in the meantime who have also become friends. They are family.

In particular, I would like to thank my parents, for always ensuring that I lacked nothing and for motivating me when I needed it most. I would also like to thank my aunt, godmother, and cousin, who were a great example of courage, determination and overcoming to me. Even when we were further away, we never forgot each other. I would also thank my closest friends, my team. I feel very grateful to have met them. And I also thank my boyfriend, who was my support, who cheered me up in the lowest moments, and who always inspired me from day to day.

Finally, I would like to thank some people I could never forget, who really touched me and whose role I respect immensely. They are teachers. Although I do not mention their names, I hope that, when you read this acknowledgement, you will see yourselves in it. Thank you for believing in me more than I did and being such great examples of your role. You were and continue to be a great role model to me.

Sofia Martins

"Technology is nothing. What's important is that you have a faith in people, that they're basically good and smart, and if you give them tools, they'll do wonderful things with them."

Steve Jobs

Contents

List of Figures

List of Tables

List of Abbreviations

ACL	Agent Communication Language
AMI	Advanced Metering Infrastructure
AOSE	Agent-Oriented Software Engineering
AS	Appliance Scheduler
ASL	Ambient Assisted Living
BDI	Belief-Desire-Intention
CNP	Contract Net Protocol
CoAP	Constrained Application Protocol
DMS	Demand Management System
DSM	Demand-Side Management
EE	Energy Efficiency
EET	Earliest End Time
ESS	Energy Storage System
FIPA	Foundation for Intelligent Physical Agents
GUI	Graphical User Interface
HAS	Home Automation System
HCI	Human-Computer Interface
HEM	Home Energy Manager
HEMS	Home Energy Management System
HTTP	Hypertext Transfer Protocol
IoT	Internet of Things
JADE	Java Agent DEvelopment Framework
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
LET	Latest End Time
MAHAS	Multiagent Home Automation System
MAS	Multiagent System
MC	Minimum Charge Level
MQTT	Message Queuing Telemetry Transport
PADE	Python Agent Development Framework
PV	Photovoltaic Panel
RC	Required Charge Level
REST	Representational State Transfer
RET	Required End Time
SDG	Sustainable Development Goal
SoC	System-on-chip
SR	Spinning Reserve
TOU	Time of Use

UI	User Interface
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
XML	Extensible Markup Language

List of Symbols

A . . . . Set of agents

B . . . . Total bill borne by the consumer (€)

C_a . . . . Critical satisfaction of agent a

E_{PV} . . . . Energy taken from the PV (kWh)

E_{grid} . . . . Energy taken from the grid (kWh)

E_{load} . . . . Total energy consumption (kWh)

M . . . . Set of markets

N_{AS} . . . . Number of Appliance Scheduler agents

P^t_{PV} . . . . Power generated by the PV at time slot t (kW)

P^t_{load} . . . . Power load at time slot t (kW)

P_{max} . . . . Maximum power load at any given moment (kW)

P_{peak} . . . . Peak Load (kW)

R . . . . Set of resources

S_a . . . . Satisfaction of agent a

W_{PV} . . . . Energy wasted from the PV (kWh)

$\overline{S_{AS}}$ . . . . Mean Appliance Scheduler satisfaction

p^t . . . . Unit price of energy at time slot t (€/kWh)

Chapter 1

Introduction

This first chapter begins with the contextualisation and motivation of this work from the perspective of resource management, under Section 1.1. Then, Section 1.2 introduces the problem involved in current engineering solutions in the previously mentioned domain, which justifies the relevance of this work as a solution to said problem. The problem statement is followed by the introduction of the objectives and tasks, under Section 1.3, and the methodology involved therein, which is presented as part of Section 1.4. Finally, Section 1.5 closes the chapter with an overview of the structure of this document.

1.1 Context and Motivation

Back in 2015, the United Nations Member States adopted the 2030 Agenda for Sustainable Development, which encompasses 17 Sustainable Development Goals (SDGs) [Nations, 2014]; in doing so, they have committed to implementing national-level integrated policies and fostering new projects in an attempt to collaboratively achieve these goals. Sustainable development spans over three main dimensions, namely economic, social and environmental. Among the latter, the SDGs highlight the need for sustainable consumption and production patterns leading to the preservation of natural resources and the minimisation and eventual suppression of global warming.

The need for sustainable consumption and production patterns has been rising considerably since 2008, which is regarded as the year when, for the first time in history, the global urban population outnumbered the rural population [Nations, 2021]; today, it is expected that by 2050 two-thirds of the world population will be living in urban areas, which will exacerbate urban demand for resources such as water and energy. As resource demand increases, the current model of resource management becomes unsustainable, preventing them from reaching future generations. The energy sector is just one among many sectors that are already experiencing this effect. Estimations for the European countries point to the continued electrification of the residential sector, mainly due to the increasing penetration of electric appliances [European Commission and Technology, 2020]. The same occurs in others parts of the world, as in the United States, which will most likely experience the same continuous increase in electricity demand across all end-use sectors [Administration, 2020].

Given the substantial role that the urban residential sector plays in resource consumption, it is crucial to empower households with mechanisms that enable them to achieve sustainable Demand-Side Management (DSM) without compromising human comfort, which is very much in line with Goal 11. This effort will involve increasing both water-use and energy-use efficiency, as dictated by Goals 6 and 7, respectively. Over the years, many proposals in support of these goals have been put forward, including, for example, improvements on equipment's energy efficiency [European Commission and Technology, 2020] and promoting the water-saving message among communities [Addo et al., 2019]. Another solution comprises (1) equipping homes with technology that guarantees sustainable consumption profiles and (2) using this technology to increase homeowners resource-use awareness, leading to behavioural changes, as Goal 12 suggests.

This path is proving increasingly feasible. As forecasts reveal that the consumer share of total devices will vastly exceed that of businesses in the near future [CISCO, 2020], one may expect Smart Homes and entailed Home Automation Systems (HAS) to soon proliferate. Furthermore, with the emergence of low-cost sensors and actuators and middleware that allow overcoming the interoperability problem, it becomes more and more accessible for homeowners to install Advanced Metering Infrastructures (AMI) to support HAS solutions. Such network of interconnected devices - best known as the Internet of Things (IoT) - together with remote services, will provide HASs with all data needed to render Smart Homes more efficient, comfortable and, ultimately, more sustainable.

1.2 Problem Statement

The Smart Home concept was officially used for the first time in 1984 [Harper, 2003]. However, despite all developments that happened ever since, current solutions still fall short of expectations. Today, Smart Homes are recognised as an IoT sub-domain, with Home Automation Systems being its cornerstone. These are systems that manage homes' appliances and other connected devices independently, on top of an IoT infrastructure offering methods to exchange data and services [Kim et al., 2015]. The opportunities raised by HASs are numerous and widely accepted, particularly in what concerns sustainable resource management; however, most HAS solutions available today face the following challenges.

- **Adaptability/Extensibility** The HAS solution is service-specific. As a result, the homeowner needs to patch together various HASs offering different services (e.g: energy management and security management). Ideally, these systems should be extensible, highly adaptable and enable multiple services to be deployed together as part of a single broader control system.

- **Connectivity** The HAS solution cannot access remote services. Thus, the HAS control over devices is based only on on-site data collected by sensors deployed in the Smart Home itself.

Other data (e.g. meteorological data), which in principle cannot be obtained through these sensors, do not participate in the process despite their relevance.

- **Interoperability** The HAS solution targets devices from a specific vendor. This is one of the major challenges faced by IoT today [Sovacool and Furszyfer Del Rio, 2020]. Homes are bundled with heterogeneous devices - both sensors and actuators - each employing a specific set of communication standards and protocols [Almusaylim and Zaman, 2019]. HAS solutions should be vendor agnostic, that is, not tied to the products of a specific manufacturer.

- **Software Portability** The HAS solution does not allows for alternative deployment architectures. Generally, not all sensing, computation, and memory resources can be provisioned by a single computing node [Zhang et al., 2018]. This becomes even more evident in the Smart Home environment, where computation resources tend to be limited. Distributed architectures, enabling intensive computation tasks to be offloaded lead to scalable HASs.

This study aims at designing a reference architecture for the control of Smart Home devices while seeking to address the challenges listed above. It is built on the hypotheses that system autonomy and control can be guaranteed by a Multiagent layer, where expertise can be spatially distributed among cooperating agents. In principle, by leveraging this software paradigm, it is possible to design a distributed HAS that is computationally efficient, flexible, reusable and highly extensible through the provisioning of new services. On the other hand, IoT infrastructure interoperability and connectivity can be ensured by means of an IoT middleware. Specifically, this study focuses on designing an architecture that extends the architecture of FIWARE [1] - an IoT middleware that has been employed as part of projects across several European cities. Consequently, combining a general-purpose multiagent system layer with a validated IoT middleware should result in a layered and modular architecture capable of meeting all the outlined challenges.

1.3 Aim and Goals

The primary objective of this dissertation can be summed up in the following sentence.

> *To design and implement a general purpose HAS favouring comfort and sustainable resource management.*

Achieving this goal encompasses five objectives, which are aligned with the SMART framework [Ogbeiwi, 2017]. These are outlined below.

- *Design* a HAS architecture based on integration of a multiagent system layer with the FIWARE middleware.

[1] FIWARE Foundation

- *Model* a Home Energy Management System (HEMS) that (1) may be implemented according to the devised architecture and (2) adopts the above-mentioned regulation criteria (comfort, sustainability).
- *Implement* and *Deploy* the HEMS.
- *Design* a realistic HEMS case study.
- *Simulate* the devised case study to prove the main hypotheses that HASs can contribute towards comfort and sustainability.

The accomplishment of all these tasks, besides leading to the achievement of the main objective, should also result in a set of significant contributions, as listed below.

- **Scientific** Propose a HAS architecture that may serve as reference and inspiration for new architectures.
- **Scientific** Add new findings to a growing body of literature on HASs (e.g. reporting new challenges involved with the implementation of these systems).
- **Scientific** Gather comprehensive results that reinforce the contribution of HASs to the United Nations' Sustainable Development Goals.
- **Applicational** Propose a HEMS and demonstrate its potential for sustainable resource management and comfort.
- **Technological** Provide a set of reusable software modules that re-used, adapted and extended in future projects.

1.4 Methodology

The tasks presented in the previous section were organised according to the Waterfall methodology. In a first phase - known as *Software Specification*, the requirements of the architecture were collected, which allowed identifying constraints and functionalities to be guaranteed by HAS architecture. Next, during the *System Modelling* phase, this architecture was modelled together with the HEMS, with the latter task employing a set of MAS-specific models that led to the full specification of the multiagent system. Then followed the identification of the system's components, their relationships and distribution - in a process known as *Architectural Design*. All components were then implemented during the *Implementation* phase, which resulted in a set of software modules that were deployed together with FIWARE ones. In the fourth phase, the HEMS scenario was defined, together with a set of evaluation metrics. Finally, the HEMS was tested under various simulations during the *Simulation* phase, which can be broadly compared to a *Software Testing* phase [Sommerville, 2010].

1.5 Structure of the Document

The remainder of the document is structured as follows. Chapter 2 introduces the main concepts within the scope of this dissertation as well as a review of past research guided by relevant research questions. Chapter 3 then introduces the proposed HAS architecture and the models of the HEMS that has been deployed in accordance with this architecture. Chapter 4 begins with a description of the simulation method that guided the assessment of the HEMS and examines the results of these simulations. A reflection on the outcomes of this dissertation and references for future work are presented in Chapter 5, thereby closing this document.

Chapter 2

Literature Review

This chapter concentrates the results of the literature review performed in order to understand the domain of the problem that this work aims to address. Thus, Section 2.1 begins by presenting background knowledge acquired namely from seminal works. Section 2.2 then introduces the Systematic Mapping Study undertaken so as to gather a collection of relevant previous works from which stem the research *gap*. Accordingly, the latter section introduces the results and conclusions obtained from said study, thus closing the chapter.

2.1 Background Knowledge

This section presents a description of fundamental concepts related to the problem domain addressed in this work, namely, Internet of Things (IoT), Smart Homes, Multiagent Systems and Home Energy Management Systems.

2.1.1 Internet of Things

The concept of Internet-of-Things (IoT) came into existence at the time of the Internet revolution, and is currently the subject of high expectations from scientists and researchers. The first IoT application, the Trojan Room coffee pot, was introduced back in 1993, but the IoT term was only coined in 1999 by the Massachusetts Institute of Technology (MIT) Auto-ID Labs [Khanna and Kaur, 2020]. Today, there's no standard definition of IoT; many definitions can be found across research projects, such as the IoT European Research Cluster (IERC) and Industrial players, like CISCO, but, in general terms, all revolve around the notion of a network of interconnected devices and services.

Likewise, various models of IoT architectures have been proposed over time, and, to date, this collection involves two well-known proposals: the three-layer and five-layer models [Al-Fuqaha et al., 2015][Said and Masud, 2013]. According to the five-layer model, which adds more levels of abstraction when compared to the former one, a Objects Layer seats at the basis of the IoT architecture and consists of the physical devices that collect and process information [Al-Fuqaha et al., 2015]; above the latter reside the communication technologies such as RFID and Wi-Fi, in what is known as the Object Abstraction layer, whose function is to transfer the data generated by the IoT devices to third layer - the Service Management Layer [Al-Fuqaha et al., 2015]. It is the function of the third layer, also known as the Middleware layer, to process incoming data and deliver it to the required services [Al-Fuqaha et al., 2015] which run at the layer above; according to five-layer model, this layer enables programmers to design new IoT applications, independently of any hardware specification. Consequently, the services are implemented at the fourth layer, the Application Layer, and can cover many domains, such as the Smart Home one. Finally, the Business Layer enables the proper management of the Application Layer's services [Al-Fuqaha et al., 2015].

According to [Al-Fuqaha et al., 2015], The IoT can also be understood as a collection of elements, namely those related to Identification, Sensing, Communication, Computation, Services and Semantics. Sensors, actuators and wearable devices are among the Sensing elements of the IoT ecosystem and communicate between each other and with Computation elements using different Communication technologies, such as RFID or NFC. In this sense, Hardware Computation elements refer to processing units such as microcontrollers (Arduino, ESP8266, ESP32, etc.) and single-board computers (Raspberry Pi, Micro Bit, etc). Data can be delivered from these Hardware elements to specific Services implemented at the Application Layer; similarly, data can also flow in the opposite direction, for example, to operate some actuator device which, in both cases, relies upon an Identification mechanism, composed of Naming and Addressing strategies that make it possible to locate both elements. Semantics are also an essential element of the IoT ecosystem and refer to the ability to extract knowledge from data and model information [Al-Fuqaha et al., 2015]; to achieve this, services need to be able to access data from heterogeneous IoT elements and, in addition, be able to understand it. This poses a major challenge, better known as the interoperability problem, as IoT devices usually communicate over a vendor-defined language, whereas services resort to XML or JSON in most cases. Application protocols, such as the Constrained Application Protocol (CoAP) or the Message Queuing Telemetry Transport (MQTT) alleviate this problem, by implementing the mechanisms through which data from heterogeneous devices can reach the services, thus bridging the Object Abstraction and Application Layers; however, they do not define marshalling formats, data structures, ontologies or management services (such as device monitoring) that are essential for services to implement the Semantic mechanisms mentioned earlier. In this domain, OneM2M, Lightweight M2M and FIWARE [1] are among the IoT middlewares that have received the most attention in recent years; most often, these IoT middlewares make use of HTTP or one Application protocol like MQTT at the lower layers of their architectures.

As the variety of the elements in the IoT domain keeps on expanding, the IoT community, from both industry and academia, is confronted with new issues and challenges. Today, it faces many open-issues such as scalability, interoperability, security and privacy, plus challenges such as those related to the standardisation of the field [Khanna and Kaur, 2020]. Despite this, the community keeps pushing forward new advancements in the field, given its potential for social, environmental

[1] FIWARE Foundation

and economic impact, which may span over all verticals of human societies, such as the Mobility, Healthcare and Smart Home sectors [Khanna and Kaur, 2020][Al-Fuqaha et al., 2015].

2.1.2 Smart Homes

A Smart Home is defined as an automated building equipped with IoT devices, namely sensors and actuators that interact with each other [Alaa et al., 2017]; it is also considered an IoT subdomain, or application [Alaa et al., 2017]. Moreover, Home Automation Systems or domotics are terms that are used interchangeably thus referring to systems that provide infrastructure and methods to exchange all types of appliance information and services [Alaa et al., 2017], so as to render houses Smart.

Within the Smart Home domain, multiple applications arise, such as those related to healthcare, entertainment, energy conservation, comfort and entertainment [Alaa et al., 2017] [Taiwo et al., 2020]. Context-aware computing, which is an important characteristic of ubiquitous computing, is the major enabler of these applications; by analysing and interpreting raw data collected by sensors installed in a Smart Home, context-aware systems extract meaningful information that can assist people in achieving their requirements and make their quality of life much easier [Almusaylim and Zaman, 2019].

Despite the great potentials of Home Automation Systems, there are also challenges that need to be considered upon design, such as those related to security and privacy, considering that these depend upon devices that are connected to the Internet and shared via a home network to provide convenience services [Almusaylim and Zaman, 2019]. At the same time, the heterogeneity of IoT standards and protocols brings in an interoperability problem that cannot be undervalued [Almusaylim and Zaman, 2019].

2.1.3 Multiagent Systems

The field of Distributed Artificial Intelligence has a long background, with roots dating all the way back to around the 1970s. It is described by scholars as the subfield of Artificial Intelligence (AI) that is concerned with coordinated, concurrent action and problem-solving [Bond and Gasser, 2014]. The DAI sub-field is further divided into three fields, namely: Distributed Problem Solving, Parallel Artificial Intelligence and Multiagent Systems (MAS), with the latter being concerned with coordinating intelligent behaviour among a collection of (possibly pre-existing) autonomous intelligent *agents*.

In the MAS research area, multiple definitions of an *agent* exist. Authors of [Russell and Norvig, 2020] defined an agent as *a flexible autonomous entity capable of perceiving the environment through the sensors connected to it*, while in [Jennings et al., 1998] authors define it as a *a computer system, situated in some environment, that is capable of flexible autonomous action in order to meet its design objectives*. Either definition encapsulate four key concepts: the *entity* (or the type of agent), the *environment* (the place where the agent is located), the *parameters* (types of data that the agent can sense from the environment) and *action* (that the agent performs, resulting in changes to the surrounding environment) [Dorri et al., 2018]. Moreover, a weak notion of agency characterises an agent as a software-based computer system that enjoys the properties of autonomy, social ability, reactivity and pro-activeness [Wooldridge and Jennings, 1995].

Referring to the first key concept of the agent definition, different types of agent architectures have emerged over the course of research in MAS; in the latter field, these architectures are thought of as software engineering models of agents [Wooldridge and Jennings, 1995]. In this sense, three main approaches to the agent architectures exist: Deliberative, Reactive, Hybrid. The first, which is a classical approach, is based on the notion of a symbolical, or purely logical reasoning; the second, being an alternative approach, attempts to solve the problems that are associated with symbolic AI, and is usually associated to its best-known example, the subsumption architecture, in which an agent's decision-making is accomplished through a set of task-accomplishing behaviours [Wooldridge, 2009]. Despite the attempt to solve the problems associated with the classical approach, the alternative approach also presents fundamental, unsolved problems, so the third approach arises as a combination of the two, consisting of two (or more subsystems), a deliberative one and a reactive one, to form a hierarchy of subsystems [Wooldridge and Jennings, 1995]. These set of notions relate to the micro-level concerning the MAS field and serve as a basis to build complex systems, at a macro level, where these individual entities, called agents, interact with each other.

One fundamental issue emerges at the MAS macro-level, namely, how to ensure that agents can understand each other so as to move to build complex patterns of interaction. In this line of thought, Ontologies provide the mechanisms to solve the communication problem; according to [Wooldridge, 2009] an Ontology is a specification of a set of terms, intended to provide a common basis of understanding about some domain. However, the definition of a common language in itself is not sufficient to enable agent-to-agent communication. Other aspects, such as synchronisation [Wooldridge, 2009] and message semantics [Dorri et al., 2018] have also been studied extensively, having resulted in a set of three widely used approaches for communication: speech-acts (first introduced by John Austin in [Austin, 1975]), message passing and Blackboard [Dorri et al., 2018]. Consequently, these approaches influenced and motivated the need for Agent Communication Languages (ACL), which provide a unique message format and ontology for all agents to communicate and interpret received messages [Dorri et al., 2018]; one of the most well-known ACLs is that proposed by the Foundation of Intelligent Physical Agents (FIPA), which has been supported by several platforms aimed at leveraging the rapid development of multiagent systems [Wooldridge, 2009]. One of such platforms is the Java Agent Development Environment (JADE) [Bellifemine et al., 2007], in which agents are implemented as Java threads that run inside a Java container. Other development tools and frameworks for building multiagent systems have been proposed, such as the Python Agent Development Framework (PADE) [Melo et al., 2019] but JADE, whose origins date back to 1998, still remains the best-known and most widely used one.

Employing one development tool such as JADE entails careful planning in designing multiagent systems. In this regard, the agent-oriented software engineering (AOSE) community has also proposed various methodologies for the analysis and design of agent-based systems, such as: Gaia [Wooldridge et al., 2000], Tropos [Bresciani et al., 2004] or Prometheus [Winikoff and Padgham, 2004], which differ in the set of processes they consider, which can span over different stages of a project's life-cycle, and even in terms of the models that form the artefacts over which these methodologies operate.

While following the guidelines of an agent-oriented methodology, the models become increasingly more detailed as the design process moves forward. It is thus expected that, as software engineers approach the implementation phase, agents' roles, interactions and decision-making mechanisms are thoroughly detailed. Different interactions between software agents can be considered in a multiagent system, resembling the interactions that take place in human communities; common interactions include Task Sharing and Result Sharing, Cooperative Distributed Problem Solving and Handling Inconsistency, as explained in [Wooldridge, 2009]. Of course, as in human societies too, these interactions have a great dependency upon individual decision-making processes, which become even more important within groups of self-interested agents. As presented in [Raiffa, 1988], *decision-theory is a means of analysing which of a series of options should be taken when it is uncertain exactly what the result of taking the option will be* and draws out many similarities with the Game Theory field, which studies interactions between self-interested agents [Parsons and Wooldridge, 2002].

From a broader perspective, multiagent systems applications serve many real-world needs, while at the same time spanning over multiple domains such as Workflow and Business Process Management, Human-Computer Interfaces, E-Commerce [Wooldridge, 2009], Cloud Computing and Smart Grids [Dorri et al., 2018], just name a few. Presently, the MAS field still posses many challenges [Dorri et al., 2018] that are being defied by the research community, but it also prevails as a powerful sub-field of DAI given that it has the potential to considerably redefine many types of software [Jennings et al., 1998].

2.1.4 Home Energy Management Systems

Home Energy Management Systems (HEMS) are defined as systems which monitor, control, and optimize the flow and use of energy in the Smart Home environment [Liu et al., 2016] on top of an IoT infrastructure. These systems range from basic systems that collect energy usage data using Advanced Metering Infrastructures (AMI) and output relevant information in the form of metrics and graphics, to automated ones, where appliances are controlled either locally or remotely [Asare-Bediako et al., 2012]. Often, these systems are characterised in terms of (1) their regulation strategies or goals (e.g. minimisation of the energy bill, reducing carbon emissions, improving comfort) (2) the strategy employed to achieve these goals, in particular, how to schedule the operation of various appliances (3) management of household appliances and (4) how the latter are modelled [Leitão et al., 2020].

To achieve the goals defined by HEMSs at the demand-side, two techniques are often employed: Demand Side Management (DSM) and Demand Response (DR) programs. These terms are often used interchangably, despite having different meanings. On the one hand, Demand Response refers to changes in normal patterns of demand-side energy consumption as a result of some external event caused by an energy provider as a strategy to lower energy use when wholesale market prices are high, for example, offering financial incentives or offering time-based rates [2]. On the other hand, Demand Side Management refers to all techniques employed at the consumption-side of an energy system [Palensky and Dietrich, 2011]; therefore, DR is classified as a sub-category of DSM, together with other DSM techniques such as Energy Efficiency (EE), Time of Use (TOU) and Spinning Reserve (SR) [Palensky and Dietrich, 2011], among others, as there is no general consensus over the categorisation of DSM.

At the demand-side improving homes' energy efficiency without compromising comfort depends on a infrastructure for monitoring of energy consumption as well as a strategy for the coordination of appliances, which can be achieved via consumption reduction or consumption shifting [Leitão et al., 2020]. The latter are among the most commonly used techniques in the residential field and are implemented based on critical choices regarding the modelling of devices, the scheduling criteria, the operational constraints and the method used to solve the scheduling problem (e.g: mathematical optimisation or decentralised techniques) [Leitão et al., 2020].

Decentralized techniques, namely MAS-based ones, can solve the scheduling problem while decreasing the computation burden that is common to centralised methods of mathematical optimisation [Gazafroudi et al., 2017], however, this raises a trade-off between reduced computation burden and global optimisation as the problem's constraints are distributed among agents in the multiagent system (in the form of knowledge) and inter-agent constraints [Yokoo and Hirayama, 2001]. As a result, no agent has a global view of the scheduling problem - which can be naturally mapped to a constraint optimisation problem, whereby constraint satisfaction stands as a technique to model and solve combinatorial optimisation problems [Barták et al., 2010].

The constrained optimisation problem involves a set of constraints and single or multiple objectives. Common objectives include reducing the electricity bill, reducing carbon emissions and improving customer comfort [Leitão et al., 2020], and are usually referred to as regulation criteria or regulation policies. Constraints, on the other hand, cover many aspects of the home environment, such as the household contracted power, the electricity bill, the preferred time window of the appliances, the power drawn by them, and other aspects related to energy storage and energy generation equipments (when those are considered), just to name a few. Constraints can be further tailored according to their underlying temporal discretisation, which can vary in the level of granularity, being equal-length hourly intervals an option. Nevertheless, constraints can be addressed in a different manner, by including them as part of the objective function within an unconstrained multi-objective framework [Leitão et al., 2020].

Another important aspect of HEMSs refers to the way in which appliances are modelled. This step should be performed according to the category of each appliance, however, there is yet no consensus as of (1) the categorisation of appliances and (2) mapping of appliances to categories. Nevertheless, appliances are usually distinguished on the basis of several features such as enabling

[2] Based on U.S. Office of Electricity: Demand Response

(1) adjustment of energy consumption, (2) interruption and (3) rescheduling, just to give a few examples.

In sum, HEMSs stand as an important solution to the energy management problem. They can deploy demand response mechanisms that react to changes in the grid, as well as efficient Demand Side Management strategies that improve overall energy efficiency and comfort. As a result, they can attain multiple benefits for homeowners, such as lowering electricity bills, serving as transparent energy aware systems [Asare-Bediako et al., 2012] that monitor and automate the scheduling of appliances on the basis of multiple regulation criteria.

2.2 Related Work

This section begins by introducing the methodology and objectives behind the Systematic Mapping Study that has been undertaken so as to gather a collection of relevant previous research works from which stem a research gap. Accordingly, the section closes with reference to the research gap and with the listing of current challenges mentioned in literature.

2.2.1 Systematic Mapping Study

The Literature Review process was guided by a well-defined Systematic Mapping methodology. In this regard, it aimed to answer the following research questions:

- **RQ1:** Are IoT application protocols or middlewares applied to Smart Homes?
- **RQ2:** Are there uses of Multiagent System architectures to support services for Smart Homes?
- **RQ3:** Are there Home Energy Management Systems based on Multiagent Systems?

Following the definition of the research questions, the scope of the study was characterised in terms of Population, Intervention, Outcomes and Experimental Design, as given below.

- **Population:** Smart Homes.
- **Intervention:** studies combining the following topics: (1) Multiagent Systems, (2) IoT application protocols or middlewares.
- **Outcomes of relevance:** Quantity of research combining the two previous technologies, identification of current applications of these in the Smart Home domain, as well as identification of related challenges and opportunities.
- **Experimental design:** Any research that includes one of the following: architecture or framework proposal, review of existing applications of the previous technologies.

Furthermore, two research engines, Scopus and Engineering Village (Inspec), were used to collect papers matching the aforementioned scope and the search-queries defined in advance for each of the research questions. Following the collection of these papers, a screening process took place, to filter those using an inclusion/exclusion criteria.

- Exclusion: All papers related to forms of study other than the following: Validation Research, Solution Proposal, Philosophical Paper.
- Exclusion: Grey Literature.
- Exclusion: Studies with more than 10 years.

From the screening process on, all the relevant papers were analysed to provide answers to the research questions and identify potential challenges in the Smart Home domain. The next sections focus on answering the three research questions, with the latter concluding the Literature Review with the identification of the *knowledge gap* and the listing of challenges.

2.2.2 Study Outcome

The next sections outline the results of the Systematic Mapping Study, which have been arranged according to the research questions mentioned in the previous section.

2.2.2.1 IoT application protocols and middlewares in the Smart Home domain

To answer this particular research question, the research process was oriented towards finding studies that took advantage of one of the following technologies: FIWARE, OneM2M, MQTT and SensorThings; this collection of application protocols and middlewares was chosen for this study not only due to the fact that they are among the most used in IoT projects today, but also because they are still in continuous development. In a second iteration, however, the application protocol/middleware requirement was dropped, thus enabling studies to be found in the field of Smart Homes, which could either integrate one such technology or not. As this study has revealed, IoT application protocols and middlewares are in fact applied in the Smart Home domain, however, they are still overlooked by the scientific community.

Starting from the domain-specific studies, one clear usage of IoT in the Smart Home domain concerns the Smart Grid applications. Studies [Mezquita et al., 2019] and [Tom et al., 2020] explore the role of Smart Homes in this domain, and serve as an example of studies that do not employ any IoT middleware. Instead, the former study relies on the usage of Arduino ESP32 modules that connect to Smart Meters to store the data captured from the real world, as well as batteries and photovoltaic panels for control; the latter, on the other hand, simply states that the Smart Meters should be connected to the home computer system, adding that the metering infrastructure should be 6LoWPAN-based.

Besides the Home Energy Management Systems, there are other domotics use-cases that are worth considering. In [Jabbar et al., 2019], for example, a Smart Cradle is introduced as a low-cost IoT-based Baby Monitoring System. The authors resort to a Node Micro-Controller Unit

(NodeMCU) Controller Board which acts as an intermediary between sensors and an AdaFruit MQTT server, collecting the data sent by the sensors and uploading them via Wi-Fi to the server; the latter is used by parents to monitor the baby's condition by exploring the sensor-collected data and also to react accordingly by issuing instructions to the actuators that are connected to the NodeMCU central control unit. The NodeMCU is an open-source software and development board that is embedded with a System-on-chip (SoC) named ESP8266 [Jabbar et al., 2019]. Today, microcontrollers, such as ESP8266 and its successor ESP32, which are commercialised by Espressif Systems, are becoming increasingly popular in the domotics domain, due to their low-cost and low-power properties. Similarly, single-board microcontrollers, such as Arduino, and single-board computers, like Raspberry Pi, have made possible some of the studies that have been found in result of the research process.

In [Andreas et al., 2019], the authors resort to an ESP32 microcontroller, coupled with an MQTT Broker to monitor and control a Door Security System; as a result, the built-in sensors of the microcontroller generate data that is then published to the Cloud MQTT Broker, using its Wi-Fi module. Other devices, such as user's mobile phones, can then subscribe to topics on the Broker's data, thereby triggering useful notification messages, as in the case of a security breach. At the same time, the Broker acts as a middleman between users' mobile phones and the microcontroller, meaning that every command issued by the users is first processed by the Broker and then delivered to it. The advantages of such design are clear, as adding other monitoring systems other than the door one would be as simple as adding a second microcontroller, offering new services through connection with the same Broker. A similar architecture is proposed in [Xu et al., 2018], where the role of the microcontroller is undertaken by a Raspberry Pi and a platform coined Blynk is used in place of the Cloud MQTT server. Even though Blynk was not included among the middleware solutions that were considered for this study, it is worth mentioning that it serves as an alternative middleware for smaller IoT projects, whose main feature is a customisable mobile dashboard.

Another example of a MQTT-based system is presented in [Cornel - Cristian et al., 2019], where authors employ ESP8266 microcontrollers to monitor temperature and humidity conditions in a Smart Home setting. Despite being one of the most used protocols for IoT applications, the authors of study [Muhammad et al., 2019] argue that MQTT lacks some security features, so it can be employed as the application-layer protocol in combination with OneM2M middleware standards to achieve secure message exchange between multi-vendor IoT devices; moreover, the authors demonstrate how this combination results in an architecture that is suitable for a Smart Home monitoring and control system composed of various sensors and home appliances.

FIWARE is another middleware that has given rise to many ongoing projects in Europe, given the funding from the European Union and the European Commission; however, there are few academic research projects that integrate FIWARE in the context of Smart Homes. In fact, only two studies could be found at the intersection of the two concepts, and, not surprisingly, both originated from European universities. Study [de la Vega et al., 2018] seeks to build a Blockchain-based communication system on top of the FIWARE architecture to enable a data marketplace where fog nodes, such as Smart Homes, hold the generated and acquired data using FIWARE's

Context Broker and share it with interested stakeholders; in addition, the proposed architecture relies on FIWARE's Data Models to define a common syntax that can be used in transactions over multiple domains, other than the Smart Home one. The second study [Vlachostergiou et al., 2016], on the other hand, expands the FIWARE standards by defining a semantic representation that maps inputs, following the FIWARE data models, operators and reference values to instructions that target the home appliances, so has to formally represent homeowners' set of rules for the management of the Smart Home.

The communication between smart devices and between these and the entities external to the Smart Home can also be achieved without resorting to an intermediary, such as a Broker. In [Dorri et al., 2017], authors suggest that each device, once connected to the Internet, should be able to establish these communications directly, for example, to request data from another device inside the home. To overcome the storage limitation, in this scenario, the devices connect to a Local Storage device, such as an external hard-drive, which should implement a First-in-First-out (FIFO) access-policy; the authors do not delve into the interoperability problem that arises when multi-vendor devices come into play, so one may argue that this problem may render the solution unfeasible.

2.2.2.2 MAS in the Smart Home domain

Regarding the second research question, the applications of MAS in the Smart Home domain can be divided into three categories, of which the first two are directly related to the Smart Grid domain, as given below.

- MAS as a mechanism to enable negotiation of energy transactions in a P2P network (decentralised approach).
- MAS as a mechanism to enable in-home agents to adapt energy consumption based on a cooperation strategy mediated by a central agent (centralised approach).
- MAS as a mechanism to enable agents in control of home appliances to cooperate, based on user-defined requirements.

Starting from the first application, study [Mezquita et al., 2019] serves as an example in which agents directly participate in the electricity market through a P2P network. The authors propose the deployment and association of a multiagent system to each building, acting as consumers, produces or even prosumers of energy; this system is composed of three sub-systems: one which groups agents that control smart devices (smart meters, batteries and photovoltaic panels), a second one which monitors the state of the microgrid (namely the real-time price of energy, and power balance) and a third one that extracts knowledge from the data available within the system and makes predictions that are useful for the decision-making process, whereby the agents attempt to optimize their payoffs in the electricity market using a non-cooperative Game Theory model for demand-side management [Mezquita et al., 2019]. Moreover, the two latter sub-systems are deployed in the cloud, whereas the first is deployed in ESP32 microcontrollers that connect to the smart devices.

The second application of the integration of MAS in the Smart Home sub-domain of Smart Grid, seems, nevertheless, to be the one that has been the subject of most studies, when compared to the first approach. However, the two applications should not be directly compared since they differ in the type of problem that they attempt to solve; while the first application is concerned with minimising buildings' energy costs and maximising profit from local energy production, it does not attempt to minimise energy costs at the neighbourhood level or balance the peak load consumption, which is the focus of the second type of application. Thus, the core idea of the latter approach is to employ an agent that monitors the state of the microgrid through communication with agents acting as either consumers, producers or prosumers of energy; through this centralised scheme, the central agent runs an optimisation algorithm in an attempt to balance the energy consumption at the neighbourhood level. Consequently, the agent uses the result of this optimisation process to issue proposals to consumer agents, so as to change their energy consumption profile at a given moment.

An example of the second application is provided in [Celik et al., 2017]. The authors propose a MAS system where home agents participate in a decision-making process using a game theory model; in this scenario, the home agents, acting as controllers of the home appliances, do not communicate with each other, but rather engage in a coordination process with an aggregator agent, whose role is to supervise the neighbourhood energy consumption profile. At the same time, the home agents optimise the respective home's electric profile, i.e., attempt to reduce the electricity costs, using a genetic algorithm to decide which instructions to issue to the home appliances. Similarly, study [Tom et al., 2020] propose a policy-based agent negotiation process, by which a *Fog* agent, installed at a neighbourhood, negotiates in real-time with *Meter* agents installed at the Smart Homes to reduce their power consumption profile during peak hours, as well as to consume energy at off-peak hours. Moreover, the authors also propose a *Cloud* agent, whose purpose is to monitor the energy demand at the city level, by collecting information issued by the *Fog* agents, and reacting accordingly by issuing policy updates to the latter agents. The same approach can be seen in [Lotfi et al., 2019], in which a Demand Management System (DMS) plays a similar role as the one of the *Cloud* agent, thus monitoring energy loads at the city level. However, in [Lotfi et al., 2019], the DMS does not intervene in the decision-making process; instead, the DMS acts as an aggregator of information, and feeds the electricity prices, gathered from external providers, to agents deployed at the Smart Homes, so that these can decide on how to operate the smart devices. The latter, in turn, shares electricity usage data with the DMS for forecasting purposes.

The third application of MAS in the Smart Home domain expands the scope considered in the two previous applications. Therefore, the role of the MAS is not only to support a Home Energy Management System (HEMS), but rather to support any user-defined requirement, such as to turn off the lights of a room whenever it is not occupied by any member of the family. The first example of this type of application is presented in [Liang et al., 2008]. According to this study, three different CC2430 microcontrollers are in charge of a specific aspect of the home environment:

either security, appliances, or environmental conditions. These controllers communicate with a central AT90USB162 microcontroller which, in turn, establishes a connection with an agent deployed at a local computer, thereby sharing the collected sensing data; in the opposite direction, instructions issued to the appliances are triggered by this agent and are translated by the central microcontroller. In this architecture, a MAS is deployed at the layer above that of the latter agent, in which different types of agents publish their services to a Broker agent, enabling cooperation between them. In addition, a dedicated agent provides an assistive user interface, through which the homeowner can set a multi-goal strategy that constrains the decision-making processes of the agents from the lower layers.

A study similar to the previous one, [Sun et al., 2013], proposes a MAS design framework targeted to the Smart Home, based on the belief-desire-intention (BDI) agent behaviour model and on a policy mechanism for multiagent collaboration. Briefly, the system considers four different types of agents, (1) sensing, (2) action, (3) decision and (4) database, all designed as BDI agents. This design pattern seems, in fact, to be recurrent in MAS architectures oriented to the Smart Home domain, where one set of agents is dedicated to the control of sensors and the other to that of actuators, the latter being able to implement themselves a decision mechanism or delegate this responsibility to a central agent, which monitors all devices. In the specific case of [Sun et al., 2013], the authors also propose a User Interface (UI) through which users' goals and home's conditions and are translated into a set of beliefs, desires and intentions for each of the agents considered in the proposed architecture. Moreover, it also enables users to define a collaboration mechanism and resource management scheme - this being an essential system requirement, the implementation details of which are not specified in [Liang et al., 2008]. Authors in [Betts and Müller, 2014] also propose a multi-layer MAS system, coined *Layered Agent Framework (LAF*, which focuses on modularity, dynamic reconfiguration and integration with non-MAS systems. It differs from other studies by focusing on the implementation of agents at different levels of abstraction, which are mapped to specific layers of the proposed architecture. As such, a developer can program an agent according to any level of abstraction, thus specialising the baseline agent layer considered in the architecture.

As a further example, resembling [Sun et al., 2013], study [van Moergestel et al., 2013] proposes a three-layer architecture, composed of (1) a Graphical User Interface (GUI) layer through which users can monitor and control the home environment, (2) a Blackboard layer where all the data that can be shared with the system's agents is collected and (3) a Device layer where agents associated to specific devices (either sensors or actuators) are implemented. In the latter layer, agents subscribe to information for a certain topic, by communicating with the Blackboard agent from the middle layer; conversely, these agents can also publish their data to the Blackboard agent. In this architecture, however, the focus is placed on the data sharing strategy rather than on the cooperation mechanisms between the agents from the lower layer.

Finally, study [Britz et al., 2014] delves on problems related to the previous architectures. Namely, the authors argue that the Smart Home environment includes two components; on the one hand, there are components being controlled (sensors and actuators), and, on the other hand, the controlling or monitoring components, which include user interfaces, applications and intelligent services [Britz et al., 2014]. According to this reasoning, middleware technologies target the first components, whereas multiagent systems target the second components. Seeing this as a common problem in the Home Automation domain, the authors attempt to bridge the gap between the two research lines by presenting an architecture that integrates the FIPA specifications with the ISO/IEC 24752 Universal Remote Console (URC) middleware. This results in a highly adaptable architecture, which solves the well-known interoperability problem within the IoT ecosystem. Moreover, as in all other proposals presented in this study, agents can both control/monitor the state of the hardware devices, and provide a user interface. Despite the added benefits of this architecture, the authors do not consider the access to remote services; specifically, agents can provide services to each other and to the users, but no communication between the Smart Home (implementing the middleware technology) and other infrastructures is considered, so the knowledge extraction process (which drives agents' decision-making processes) is limited to the data collected by the smart sensors that are built into the Smart Home.

2.2.2.3 MAS-based HEMSs

In the domain of agent-based HEMSs, important considerations stand out among them, namely whether they:

- Centralise the optimisation problem on a single agent, which gathers all metering information and equipment needs and constraints from other agents, or in alternative, whether they impose a coordination mechanism (e.g. based on negotiation) between various agents so that knowledge over the optimisation problem is distributed among them.
- Enable or not Demand Response programs that allow, for example, for dynamic pricing strategies to be employed by energy providers.

In [Abras et al., 2008] and [Abras et al., 2010], authors propose a Multiagent Home Automation System (MAHAS) involving two distinct agent classes. The multiagent system is composed of a set of *resource* agents which are in direct control over energy resources (such as solar energy), as well as *equipment* agents, following a one-to-one mapping of the latter agents to equipments. In this architecture, the control systems of all agents are independent, thus forcing the latter to follow a coordination mechanism based on negotiation. The negotiation protocol is employed under two scheduling mechanisms, an Anticipation Mechanism and an Emergency Mechanism. On one hand, the Anticipation Mechanism allows *equipment* agents to derive a profile of power consumption corresponding to a large time window; however, this mechanism forces all agents to implement learning algorithms leading to predictions as of power production or power consumption needs (also called *predicted set points*), according to the agent class. As a result of the uncertainty involved with the latter mechanism, the Emergency Mechanism is introduced to enable agents to react quickly to avoid violations of energy constraints or homeowner's requirements, as this can not be anticipated. Each agent is assigned a satisfaction function which dictates the level of satisfaction of the agent in accordance with some characteristic variable that reflects the level of satisfaction of the homeowner with a specific service. The Emergency Mechanism is thus triggered when the satisfaction level, which is constantly monitored, reaches a critical level defined by the homeowner. Moreover, the negotiation protocol is based on the Contract Net Protocol (CNP), and results in either a power consumption profile that guarantees similar satisfactions for all *equipment* agents or on a negotiation with the homeowner when no feasible scheduling solution can be achieved based on the homeowner's constraints. One important point to emphasise from this HEMS proposal is that resource agents always select the consumption proposals that lead to the maximum possible satisfaction value for all *equipment* agents, so it does not take into other regulation criteria other than comfort (e.g: reducing the electricity bill).

The latter study considers a system in which the control of home appliances is distributed. By contrast, authors in [Li et al., 2016] employ a single agent that decides which appliances to switch on or off. At each hour of the day, it runs a sorting algorithm to make this decision; under this setting, two sorting strategies may be employed: sorting based on user comfort - considering a priority list that reflects the importance of which appliance to the homeowner - or device energy consumption - taking into account which devices consume less energy. The algorithm takes as input the hourly prices of energy as announced by the energy provider, the value of available energy supply and the energy needs of each appliance, which are exchanged between equipment agents and the agent in charge of the scheduling mechanism. As a result, the latter agent communicates to each equipment agent his decision to switch on or off the corresponding appliance by iterating over the priority list until the total supply can no longer sustain the total demand. Thus, control is delegated to a single agent having knowledge of all constraints involved in the scheduling problem and, as a result, an optimum scheduling can be achieved based on homeowners' criteria.

Unlike the two previous studies, studies [Asare-Bediako et al., 2013b] and [Asare-Bediako et al., 2013a] look at the energy management problem from a *macro* perspective, where residential appliances are operated to respond to external signals as part of a Demand Response strategy. A Dynamic Price Mechanism is applied outside the Smart Homes and the Dynamic Price value is communicated to each one at every time interval t (set to 45 minutes under simulations). From the demand-side, the architecture of the Smart Home lies on a hierarchical multiagent system composed of four collaborating agent-groups: Control and Monitoring Agents (CMA), Information Agents (IA), Application Agents (AA) and Management and optimisation Agents (MOA). Most importantly, the MOA agents monitor and coordinate the activities of all other agents while solving the optimisation problem based on a control strategy set by the homeowner (either *Comfort*, *Cost*, *Green* or *Smart*). As a result, the different energy management and optimisation strategies offer flexibility to the homeowner while supporting the distribution network and the energy supplier.

Dynamic pricing strategies can also be employed as part of an isolated Smart Home system, as demonstrated in [Feron and Monti, 2017]. In the latter study, a hierarchical market-based MAS is deployed to control the domestic electrical or thermo-electrical devices using internal dynamic pricing. In this architecture, all controllable devices are regarded as market participants aimed at fulfilling their needs. They interact via a bidding process in specific markets deployed as part of

Table 2.1: Summarised results of the Literature Review.

	Domain Specific	Middleware Support	Contains intelligent objects	Contains intelligent, communicating objects	Connected home	Learning home	Alert home
[Kang et al., 2018]	X		X		X		
[Mezquita et al., 2019]	X		X	X	X		
[de la Vega et al., 2018]		X			X		
[Dorri et al., 2017]					X		
[Kouzinopoulos et al., 2018]					X		
[Xu et al., 2018]		X					
[Tom et al., 2020]	X		X	X	X	X	
[Jabbar et al., 2019]	X	X					
[Andreas et al., 2019]		X					
[Cornel - Cristian et al., 2019]		X					
[Muhammad et al., 2019]		X					
[Vlachostergiou et al., 2016]		X	X	X	X		
[Celik et al., 2017]	X		X		X		
[Lotfi et al., 2019]	X		X	X	X		
[Liang et al., 2008]			X	X			
[Sun et al., 2013]			X	X			
[Betts and Müller, 2014]			X	X			
[van Moergestel et al., 2013]			X				
[Britz et al., 2014]		X	X	X			
[Abras et al., 2008]	X		X	X	X		
[Li et al., 2016]	X		X	X	X		
[Asare-Bediako et al., 2013a]	X		X	X	X		
[Feron and Monti, 2017]	X		X	X	X		

various rooms, so as to respond to the space heating demand in each of them. In addition to the internal heat and cooling markets, the system introduces an electricity market which is based on the Powermatcher approach. Globally, the system integrates three agent classes: *device* agents acting on behalf of a specific device following a one-to-one mapping, *local aggregator* agents and *global aggregator* agents. Each *device* agent translates the corresponding device's objective function and constraints into a bid that is communicated to the *local aggregator* agent deployed at the corresponding room. The latter translates these bids depending on electricity and heat/cooling price into a bid depending solely on electricity price, with this, in turn, being communicated over to the *global aggregator* agent which determines the electricity market price and ensures the electrical power balance of the system as a whole.

2.3 Summary

The previous sections outlined the main applications of IoT protocol applications and middlewares, and multiagent systems in the Smart Home domain, while providing examples of the synergy between them. The results of the Literature Review process are summarised in Table 2.1, where the last five columns relate to the Smart Home categories considered in [Harper, 2003].

By consulting the aforementioned table, one may verify that there is a *gap* in the domain of smart-home oriented technologies; namely, there are few studies that encompass all the following characteristics:

- Being general purpose.
- Enabling the coordination between smart objects.
- Enabling access to services external to the home.
- Relying on an IoT middleware.

In addition to identifying the *knowledge gap* mentioned above, this study process also enabled the identification of some current challenges involved with the three mentioned technologies; these challenges are listed below, and should be taken into consideration when implementing a system that tries to bridge the mentioned *gap*.

- If a fog computing infrastructure is to be implemented, where producers of data are directly responsible for performing local computation, there's the need to install, at the Smart Homes, an hardware infrastructure that supports these (computing power and storage) needs, or, alternatively, to subscribe to a Cloud solution which may become expensive and overcome the benefits of a data-monetisation solution from the data producer's perspective.
- Given the lightweight property of most microcontrollers, they must devote their computational power to running core functionalities. Security and privacy protocols, which could be run by these devices, introduce processing and energy overheads.
- In terms of time criticality, some IoT systems need to deliver messages to the user almost instantaneously (for example, in the case of a home security breach). Although these systems require service guarantees, the fact that messages travel through the Internet (which is subject to varying delays and network congestion) might render them impractical.
- It is difficulty to embed Java-based agents into lightweight microcontrollers. Since Java is an interpreted layer, which requires a JAVA Virtual Machine to be constantly active, it introduces memory and processing overheads. Compiled languages, such as C++, offer a better solution, but, at the moment, there are no MAS frameworks based on such languages.
- The design of a unified system that can deal with the collaborative control of home appliances is particularly challenging given the heterogeneity of these devices.
- Running a Broker at the Smart Home level, which mediates data exchange between smart devices, increases the chances of occurrence of a widespread failure of the Smart Home Management System, even if the smart devices remain operational (for example, most low-power devices are connected to a battery), as the Broker acts as a single point of failure. In addition, it adds scalability issues, as the smart-device network can only expand as much as the Broker supports it.

Given these considerations, both the *gap* and the current challenges, the next sections introduce the proposed Smart Home architecture, which attempts to tackle the limitations of current studies encountered in the course of the Literature Review.

Chapter 3

Modelling Home Automation Systems

This chapter presents the various models that were developed as part of the design of the proposed HAS architecture. Section 3.1 begins the chapter by introducing this architecture has a layered and modular architecture that combines a multiagent layer with an IoT middelware. Next, Section 3.2 presents an external viewpoint model of a HEMS that integrates with the proposed HAS architecture; this is a resource-based integrated market model, which describes the resources, markets, agents and regulation policies involved in the system. To complement the previous model with the specification of each cognitive agent class, an internal viewpoint model based on the AgentSpeak(L) semantics is introduced as part of Section 3.3. Then, Section 3.4 presents multiple Unified Modeling Language (UML) models that clearly capture the implementation of the HEMS, namely in terms of developed components and possible deployment strategies. Finally, Section 3.5 summarises and highlights the main features of the HAS architecture according to the models presented in the previous sections.

3.1 Home Automation System Architecture

The proposed HAS architecture encompasses four layers, as depicted in Figure 3.1. The three lower layers are generally integrated as part of a FIWARE-based system, providing the interoperability of this architecture. At the bottom layer lie the actuator and sensor devices that communicate over diverse transport protocols (e.g. HTTP, MQTT) and message protocols (e.g. JSON, Ultralight). Sensor measurements flow from the sensor devices to the Broker, and commands flow from the Broker to the actuator devices; however, the Broker can only interpret requests over HTTP and in accordance with its NGSI-v2 interface (with a JSON payload). To overcome this challenge, one particular type of component - known as *IoT Agent* - acts as an intermediary between the Broker and the devices, thereby translating any device-native protocol to that of the Broker. In this architecture, the Orion Context Broker plays a fundamental role. This component is essentially a NGSIv2 server offering a REST API interface that enables the creation and management of context data. The context of an entity is interpreted as the state of a physical or conceptual object (an entity) that exists in the real world, for example, an appliance represented as part of the bottom

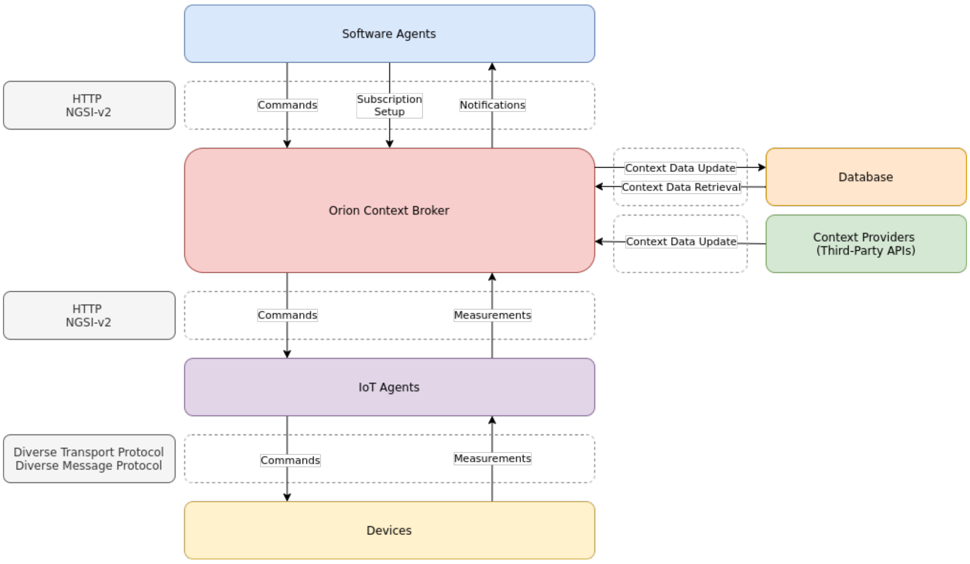


Figure 3.1: Proposed Home Automation System architecture.

layer. The Broker relies on a database system to store the context data of these entities, to update or retrieve it following the request of other components of this architecture. Moreover, the architecture can be tailored to retrieve context data, such as weather forecasts, from third-party APIs - also known as *Context Providers*. Taken together, the three bottom layers enable interoperability, access to remove services, creation, storage and management of context data.

On the other hand, autonomy and control of the home environment is guaranteed by a Multiagent System layer. Until now, many proposals have been put forward as of the mapping of agents to actuator and sensor devices. While many consider a one-to-many mapping of agents to actuator devices, this architecture opts for a one-to-one mapping, which stands as a more natural representation and enables agent's to specialise on a single device which leads to easier adaptability and extensibility. Furthermore, as Figure 3.2 suggests, a one-to-many mapping of agents to sensor devices is considered, given that agents will often rely on more than one sensory input for their decision making regarding the management of actuator devices. It is important to note that, by integrating the Multiagent System layer with the FIWARE architecture, the latter creates an abstraction over physical devices so that agents don't have to specialise on any transport or message protocol, having to adapt only to context data attributes and available commands. In this sense, context data flows from the Broker to the agents, and for this reason, the agents must expose an HTTP communication endpoint that must be supplied when requesting a subscription setup; this mechanism enables agents to subscribe to changes of context information so that they can be notified asynchronously when something meaningful happens, such as the update of a particular attribute of an entity. In turn, agents can update their knowledge base upon receiving these notifications. On the other hand, agents are able to change the state of their environment by issuing commands as HTTP requests to the Broker, thus causing the command to propagate through the various layers until reaching the lowest layer.

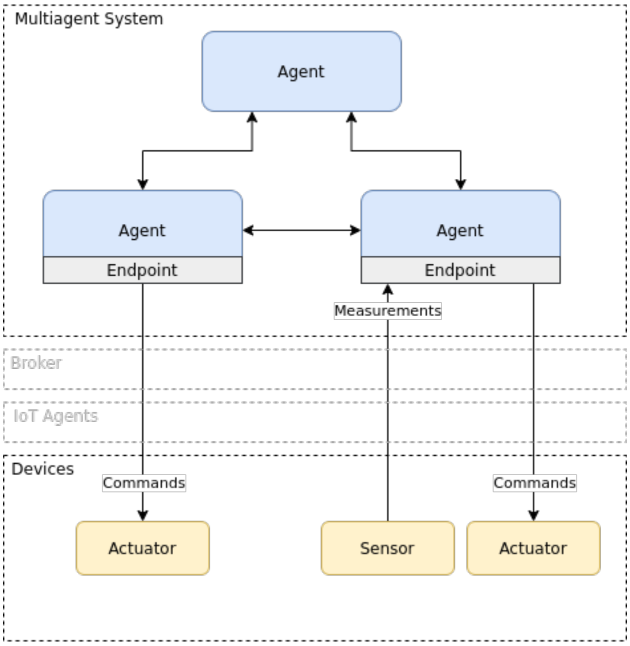


Figure 3.2: Interactions and mapping between agents and IoT devices.

As for the multiagent system itself, this architecture does not impose any particular organization. Interaction mechanisms and roles may be freely explored, as long as the previous device-to-agent mapping is employed. Consequently, one may employ a hierarchical MAS design, whereby service-specific agents serve as mediators for the coordination of the activities of agents in control of specific appliances. In fact, as the latter operate over devices that consume shared resources (such as energy and water), they must cooperate and agree over the usage of these resources. To illustrate this, the following sections describe a specific MAS design for energy management, considering an agent that serves as a mediator for the scheduling of energy consumption by various appliances.

3.2 External Viewpoint Model

The interactions taking place in the HEMS were inspired by the work of [Abras et al., 2008][Abras et al., 2010], in which agents exercise direct control over household appliances. Constraints over the home's resources, namely in terms of energy resources consumed by these appliances, impose the adoption of a coordination mechanism. In this model, as in [Abras et al., 2010], Belief-Desire-Intention (BDI) agents engage in a negotiation process mediated by a regulatory agent, with the ultimate goal of scheduling their appliances' energy consumption.

As the aforementioned system exhibits features of a resource-based market, the selection of ResMAS [Rúbio et al., 2017] as a modelling technique took precedence over available alternatives, namely those proposed by [Passos et al., 2011] and [Rúbio et al., 2019]. Furthermore, the ResMAS model is complemented and fine-tuned by introducing an Agent Class model [Kinny and Georgeff,

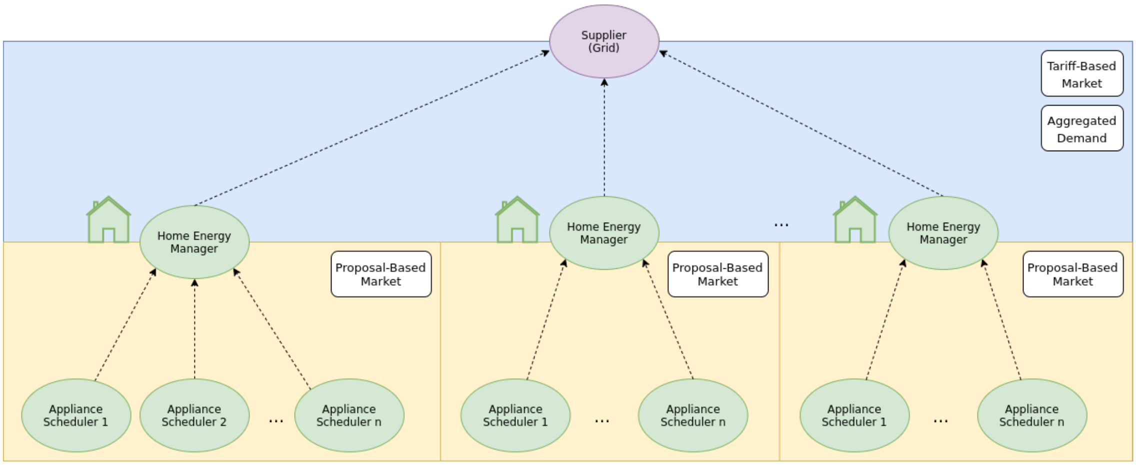


Figure 3.3: Markets considered in the model.

1997], as the former model alone cannot convey the distinctive features of agents that, having the same role in a market, exhibit significantly different behaviours.

According to the ResMAS modelling technique, the specification of the resource-based marked follows an iterative approach starting from the market definition as in (3.1).

$$ResMAS = \langle R, M, A \rangle \tag{3.1}$$

3.2.1 Resources

The market-based model considers only electricity resources, as given by (3.2). In practice, agents in the HEMS negotiate and, thereby, schedule the consumption of instances of these resources.

Electricity resources move through two distinct markets (3.3), depicted in Figure 3.3, each characterised by their allowed participants, artefacts, regulation mechanisms and processes that describe resource allocation mechanisms [Rúbio et al., 2017].

$$R = \{\langle ENERGY, kW \rangle\} \tag{3.2}$$

3.2.2 Markets

The m_1 market (3.4) is the source market for energy resources, whereby Home Energy Manager (HEM) agents, each associated with one home, can purchase electricity from Energy Suppliers (3.5). For the sake of simplicity, no distinction is drawn between Energy Suppliers and Energy Providers, so the latter is perceived to encapsulate both roles. Market m_1 is further characterised as a tariff-based market in which low volumes of resources are exchanged following the announcement of payment rates in a day-ahead fashion. In this regard, the m_1 market is free of any kind of fees (such as sign-up fees). Tariffs (3.6) are simply classified in terms of $Rates$, that is, an association between each hour of the day and the corresponding price of a unit of energy resource. Hence, the Supplier agent transacts instances of resources to the HEM agents at each hour (see (3.7)), following the established contract (3.8). The latter, in turn, become accountable for the payments (3.9) corresponding to these transactions, which are computed based on the announced *Rates*.

$$M = \{m_1, m_2\} \tag{3.3}$$

$$\begin{aligned} m_1 = &\langle Part_1, \{tariffs_1, transactions_1, contract_1\}, \\ &SimpleAuthorization, TariffBased\rangle \end{aligned} \tag{3.4}$$

$$Part_1 = \{Supplier, HomeEnergyManager\} \tag{3.5}$$

$$tariffs_1 = \langle Rates, Fees\rangle = \langle\{\langle r, 1€, 0h\rangle, ...\}, \emptyset\rangle, \quad s.t.\ r \in R \ \ \& \ |Rates| = 24 \tag{3.6}$$

$$\begin{aligned} transactions_1 = &\{\langle Supplier, HomeEnergyManager, \quad 10kW, 0h\rangle, ...\} \\ &s.t.\ |transactions_1| = 24 \end{aligned} \tag{3.7}$$

$$\begin{aligned} contract_1 = &\langle Parties, Service, Obligations\rangle = \\ &\langle\{Supplier, HomeEnergyManager\}, \quad transactions_1, payments_1\rangle \end{aligned} \tag{3.8}$$

$$payments_1 = \{2€, 3€, ...\}\, s.t.\, |payments_1| = 24 \tag{3.9}$$

$$\begin{aligned} m_2 = &\langle Part_2, \{proposal_2, transactions_2\}, \\ &SimpleAuthorization, ProposalBased\rangle \end{aligned} \tag{3.10}$$

$$\begin{aligned} Part_2 = &\{HomeEnergyManager, \\ &ApplianceScheduler\} \end{aligned} \tag{3.11}$$

$$proposal_2 = \{\langle 20kW, 0h\rangle, ...\}\, s.t.\ |proposal_2| = 24 \tag{3.12}$$

$$\begin{aligned} transactions_2 = &\{\langle HomeEnergyManager, \\ &ApplianceScheduler, 1kW, 0h\rangle, ...\} \\ &s.t.\ |transactions_2| = 24 \end{aligned} \tag{3.13}$$

In the m_1 market, HEM agents purchase resource instances from the Suppliers following the needs of the Appliance Scheduler (AS) agents that engage in the m_2 market (3.10). Each of the latter agents controls a single appliance and participates in a proposal-based negotiation process with the corresponding HEM Agent (3.11), leading to a schedule of energy consumption for each appliance.

In this scenario, the AS agents put forward proposals of daily energy consumption, as depicted in (3.12), following a Contract-Net protocol. As the result of this negotiation, the $transactions_2$ (3.13) set reports the daily energy consumption profile of each AS agent considering, once again, a 24-hour period and the temporal discretisation in the order of the *hour*.

3.2.3 Agents

From the previous discussion, one can group the various agents into three distinct classes according to their role in the agent-based market (3.14). First, *Supplier* agents are defined as agents who have access to infinite resources so that they can always respond to increasing demand, as given by the second property in (3.15). Following the ResMAS model, the third property specifies that these agents do not require any resource to operate, as they are just producers and never consumers of resources. In addition, this model does not impose any goal-set Γ_1.

As of the second agent-class, a_2, the HEM agents (3.16) are the ones responsible for purchasing energy in m_1 whenever the availability of solar energy is not sufficient to meet the AS agents' needs ($Req^t \equiv P^t_{load}$). The availability (*Avail*) of resources is not only dependent on the home's energy-production capacity (P^t_{PV}, kW), driven by photovoltaic panels, but also on a fixed maximum energy-load permitted at each hour (P_{max}). According to the consumer's preference, the HEM Agent may adopt one of three goals Γ_2 (3.17): *Bill* (3.18), considering the unit price of energy as announced by the supplier (p^t), *Green* (3.19) or *Comfort* (3.21).

This model also incorporates the concept of *satisfaction*, which has been refined beyond its adoption in [Abras et al., 2008] and [Abras et al., 2010]. In this regard, the HEM agents are characterised by a critical satisfaction value, given by the consumer, in the $[0,100]$ interval, where 100 corresponds to the maximum satisfaction. Furthermore, HEM agents are subject to the constraint that the output of their goal function must always be equal to or greater than their critical satisfaction value. It is the role of the consumer to define the satisfaction function of the HEM Agent depending on one of the three criteria given above.

Similarly, a critical satisfaction value C also characterises the Appliance Scheduler agents (3.22). This value, which is also determined by the consumer, is always compared against the output of a satisfaction function S which directly maps to the agents' goals (3.24); the latter function enables the agent to establish its preferences Π_3 over the possible schedules of the appliance. The satisfaction function depends on the type of appliance being controlled by the a_3 agent (3.25), so further breakdown of this agent class into three distinct classes is needed, as given in Figure 3.4. It is worth noting that the Agent Class model presented in the latter figure introduces two agent classes (*GasDetector* and *LeakageDetector*) which do not interfere with agents' roles within the electricity market; nevertheless, these classes are discussed in detail in the following sections.

The HEMS model distinguishes between Uninterruptible, Curtailable, and Interruptible Loads [Beaudin and Zareipour, 2015]. The assignment of appliances to these categories still lacks consensus, however, this model takes into account the one that can be inferred by inspection of the agent class diagram. Thus, distinctive characteristics have been ascribed to each of these categories, giving rise to some beliefs of the corresponding agents.

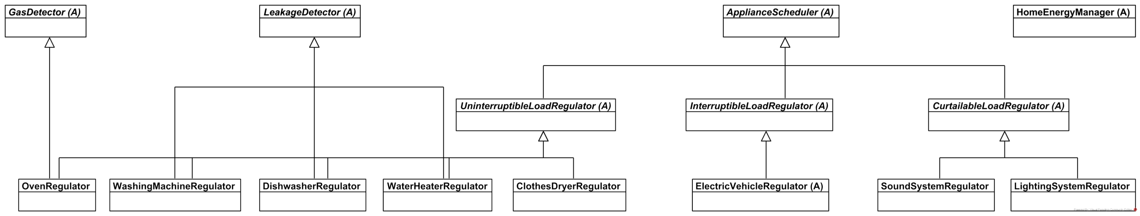


Figure 3.4: Agent Class model corresponding to the m_2 market.

Firstly, a fixed power consumption value (measured in kilowatts) and a cycle duration (measured in hours) characterise the Uninterruptible appliances. Agents related to this category are given static knowledge as of the consumer's preferences, namely: the *Earliest* (EET), *Required* (RET) and *Latest* (LET) end-times that the appliance's operation should comply with. Thus, the satisfaction of an agent of this category is given as a function of the *actual* end-time (x) in relation to the three other values provided by the consumer (3.26).

The Curtailable appliances, as opposed to the previous ones, are capable of regulating mid-operation the power they consume, so they are caracterised by *minimum* and *maximum* power-consumption values. When it comes to the consumer's preferences, the latter should indicate the *start* and *end* times for the operation of this type of appliance, assuming it will always be working within said period. Agents in charge of regulating these appliances will compute their satisfaction as a function of a *characteristic variable* (x), as given in (3.27). In the case of the *Lighting System* and the *Sound System* agents, given in Figure 3.4, their satisfaction depends on the *illuminance* and the *volume* levels, respectively. Again, this function also takes into consideration the consumer's preferences in terms of *minimum* (MIN), *required* (REQ) and *maximum* (MAX) values for the characteristic variable.

The ability to interrupt and resume the operation at a later time is what distinguishes the third category of household appliances. Hence, Interruptible appliances are defined by a fixed power consumption value and the *start* and *end* times which qualify the period within which they are allowed to operate. As for this category, the model illustrates the particular case of the electric vehicle, which is further described by a charging rate (in % per hour). Again, the consumer must supply the *minimum* (MC) and *required* (RC) charge levels that give rise to the corresponding agent's satisfaction function (3.28). As a consequence, the characteristic variable concerning the electric vehicle corresponds to its charge level at the end of the period allowed for recharging. It is worth noting that the present model does not incorporate the hypothesis of battery discharging in periods when the vehicle is not charging, yet there are other models that have set focus on this assumption.

$$A = \{a_1, a_2, a_3\} \tag{3.14}$$

$$a_1 = \langle SupplierAgent, \infty, \emptyset, \Gamma_1 \rangle \tag{3.15}$$

$$a_2 = \langle HomeEnergyManager, Avail, Req, \Gamma_2 \rangle \tag{3.16}$$

$$\Gamma_2 = \{\Gamma^1, \Gamma^2, \Gamma^3\} \tag{3.17}$$

$$\Gamma^1(Bill) : min\ B = \sum_{t=0}^{23} p^t.(P^t_{load} - P^t_{PV}).h^t$$
$$s.t.\ P^t_{load} \leq P_{max} \ \ \& \ \ S_{HEM}(B) \geq C_{HEM} \tag{3.18}$$

$$\Gamma^2(Green) : min\ E_{grid} = \sum_{t=0}^{23} (P^t_{load} - P^t_{PV}).h^t$$
$$s.t.\ P^t_{load} \leq P_{max} \ \ \& \ \ S_{HEM}(E_{grid}) \geq C_{HEM} \tag{3.19}$$

$$h^t = \begin{cases} 0 & \text{if } P^t_{load} \leq P^t_{PV} \\ 1 & \text{if } P^t_{load} > P^t_{PV} \end{cases} \tag{3.20}$$

$$\Gamma^3(Comfort) : max\ \overline{S_{AS}} = \frac{1}{N_{AS}} \sum_{i=1}^{N_{AS}} S_{AS_i}$$
$$s.t.\ P^t_{load} \leq P_{max} \ \ \& \ \ S_{HEM}(\overline{S_{AS}}) \geq C_{HEM} \tag{3.21}$$

$$a_3 = \langle ApplianceScheduler, \emptyset, Req, \Gamma_3, App, \Pi_3 \rangle \tag{3.22}$$

$$Req = \{\langle 0h, 1kW \rangle, ...\}\ s.t.\ |Req| = 24 \tag{3.23}$$

$$\Gamma_3 : max\ S_{AS}\ s.t.\ S_{AS} \geq C_{AS} \tag{3.24}$$

$$App \in \{WashingMachine, DryerMachine, ...\} \tag{3.25}$$

$$S_{unint} = \begin{cases} 0 & if \quad x < EET \\ \frac{100.(x-EET)}{RET-EET} & if \quad EET \leq x \leq RET \\ \frac{100.(x-LET)}{RET-LET} & if \quad RET < x \leq LET \\ 0 & if \quad x > LET \end{cases} \tag{3.26}$$

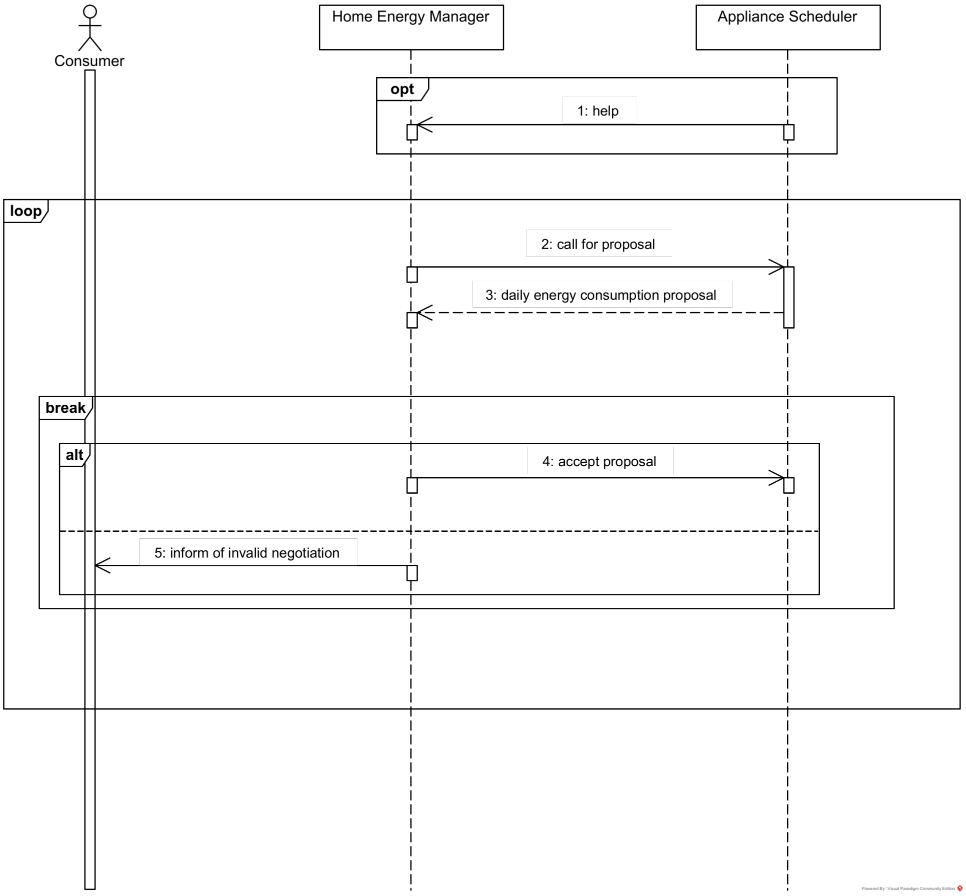


Figure 3.5: Negotiation protocol.

$$S_{curt} = \begin{cases} 0 & if \quad x < MIN \\ \frac{100.(x-MIN)}{REQ-MIN} & if \quad MIN \leq x \leq REQ \\ \frac{100.(x-MAX)}{REQ-MAX} & if \quad REQ < x \leq MAX \\ 0 & if \quad x > MAX \end{cases} \tag{3.27}$$

$$S_{int} = \begin{cases} 0 & if \quad x < MC \\ \frac{100.(x-MC)}{RC-MC} & if \quad MC \leq x \leq RC \\ 100 & if \quad x > RC \end{cases} \tag{3.28}$$

3.2.4 Market Protocols

The ResMAS modelling technique introduces two processes, namely tariff-based and proposal-based, that describe markets as a whole. In the case of market m_1, the former process can fully qualify the flow of resources from the Supplier agent to the Home Energy Manager agent, starting from the day-ahead tariff publication and ending with the establishment of a contract. In the m_2 market, however, a negotiation protocol is also introduced as a sub-process of the proposal-based market process [Rúbio et al., 2017].

In the proposed HEMS model, the HEM Agent is considered to be the initiator of the request for proposals, as given in Figure 3.5. In this request, the initiator states a *satisfaction* value that will guide the request recipients to search for a suitable proposal. It then follows that Appliance Scheduler agents will issue proposals for daily-power consumption that are subject to the constraint that the satisfaction achieved with that scheduling shall be equal to or less than the satisfaction value required by the HEM Agent. Thus, under a resource-constrained scenario, one may expect the first round of the negotiation protocol to fail, as HEM agents will start by requesting AS agents to submit proposals for a maximum satisfaction value.

The negotiation protocol will span over multiple rounds, as the HEM Agent will issue new requests, decreasing at each iteration of the protocol the required satisfaction value by a percentage that can be customised. This sub-process will then terminate when the HEM Agent selects, at a certain iteration, the best possible scheduling of appliances according to the regulation criteria. It is important to note that this schedule shall (1) be selected among all possible combinations of energy consumption proposals issued during said iteration and (2) equal or exceed the HEM Agent's critical satisfaction value as well as that of all the AS agents. Supposing there is no such scenario that can accommodate all the consumer preferences, the HEM Agent will inform the consumer that there is no viable solution; in such case, the constraints, namely those related to the critical satisfaction values, shall be relaxed. Thus, this sub-process results in reduced computation burden in comparison to centralised approaches, as the problem's constraints are distributed among agents in the Multiagent System, and inter-agent constraints [Yokoo and Hirayama, 2001].

The negotiation protocol described above is a sub-process of the broader day-ahead market process. In works [Abras et al., 2010] and [Abras et al., 2008], two distinct processes are considered to integrate this negotiation protocol, of which the first, known as the Anticipative Mechanism, is equivalent to said day-ahead process. The second - the Emergency Mechanism - can be activated multiple times throughout the day, unlike the first. The purpose of this mechanism is to enable the AS agents to request a new negotiation as a reaction to unexpected events that prevent them from fulfilling the daily consumption profile resulting from the Anticipative Mechanism. This mechanism is illustrated by the *help* message issued by the AS Agent in Figure 3.5.

A clear difference between this work and the previously mentioned ones is that the present one considers that the Emergency Mechanism allows a renegotiation of consumption that will always involve the period from the following hour until the end of the day. This allows agents in control of *Uninterruptible* and *Interruptible* appliances to also participate in this mechanism, which would not be possible considering a negotiation in short *checking* periods, as proposed in the previous works. A possible rationale for this approach could be that the authors did not consider the segregation of appliances. Still along this line of reasoning, it is important to note that the former works also disregard different regulation policies, which differentiates them from the present one.

3.3 Internal Viewpoint Model

Following the approach taken by [Kinny and Georgeff, 1997], the MAS system considered as part of the HEMS has been modelled according to external and internal viewpoint models. The previous section introduced the system from an external viewpoint, which allowed for the definition of agent classes, their roles, responsibilities, services and external interactions in the multiagent system. From an internal viewpoint, structuring each agent class requires a model that can capture their internal behaviour. It follows that, in this work, the internal viewpoint model relies on the semantics of AgentSpeak(L), which is a language that targets the formalisation of Belief-Desire-Intention (BDI) agents, with proven operational and proof-theoretic semantics [Rao, 1996].

The AgentSpeak(L) language has long been regarded as a tool that bridges the gap between formal (logical) specification and practical implementation of cognitive agents, thus being used as a model for formalising mutiagent systems in various domains, such as that of Intelligent Transportation Systems [Rossetti et al., 2002]. In comparison to other models, this one stands out for its validated operational semantics. Given these benefits, proposals for AgentSpeak(L) interpreters have emerged, of which Jason [Bordini and Hübner, 2006] is arguably the most prominent one.

In this work, the modelling process resulted in the specification presented in Section A. A top-down strategy was followed with reference to the class model depicted in Figure 3.4, which guided the definition of the belief and plan sets for each agent class. Following the principle outlined in [Kinny and Georgeff, 1997], this model benefits from the inheritance mechanism within class hierarchies, of which some classes are abstract (those marked with an *A* in the class diagram), i.e. cannot be instantiated as opposed to concrete classes. Moreover, this model focuses on those aspects of agents' plan and belief sets that are fundamental for (1) their participation in the negotiation process involved in market m_2 and (2) their reaction to external events such as a leakage in a room, leading to the activation of the Emergency Mechanism when appropriate. Finally, it is also worth noting that the model considers other semantic constructs which provide reduced modelling complexity so that the model focuses solely on aspects that are fundamental to the definition of agent classes. In this regard, these constructs include some operators of the Jason API (such as *nth*), as well as inference rules whose purpose is further detailed together with the model.

3.4 System Implementation and Deployment

The previous sections presented the models of the envisioned multiagent system from two perspectives, *external* and *internal*. In addition, from the perspective of the system's implementation, it is necessary to describe it from an architectural point of view, which involves the specification of its classes, components and artifacts. The Unified Modeling Language (UML), specifically its most recent version (2.5.1)[1], is a very powerful tool for modelling the latter. Accordingly, this section presents the UML diagrams developed as part of the modelling process.

[1] Unified Modeling Language Standard

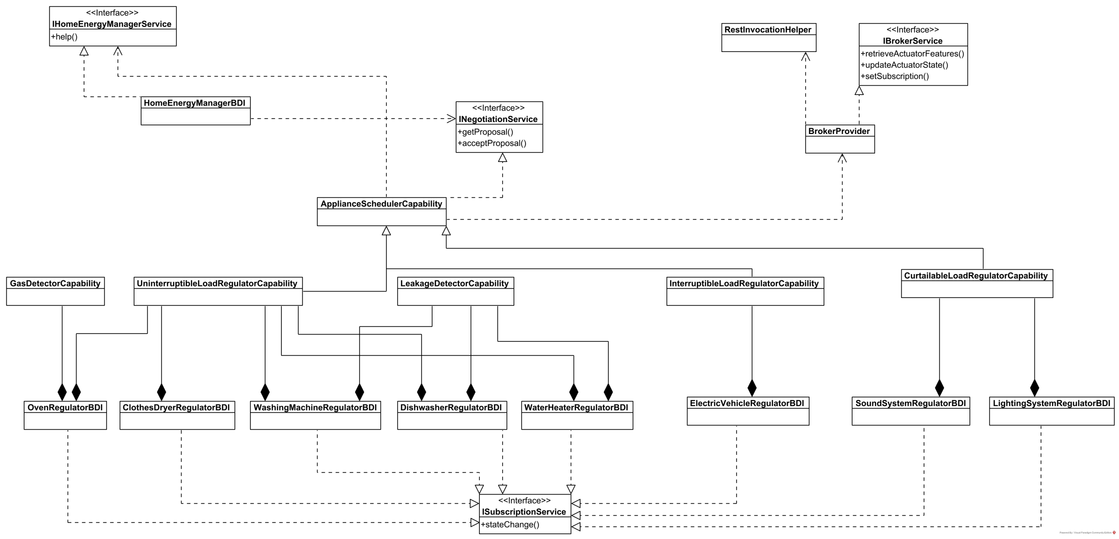


Figure 3.6: Class diagram of the proposed HEMS.

The component diagram depicted in Figure 3.7 presents all the components, ports, provided and required interfaces. The separation of the model elements into components follows the logic behind the Active Components approach. In fact, Jadex [2] - an Agent Programming Language based on the same approach - was the language chosen for implementing the multiagent system. In developing a Jadex distributed system, all entities are regarded as Active Components, which are ascribed sets of provided and required service interfaces that enable them to interact with their environment. Thus, the component diagram can capture all these key features. The diagram also depicts the components that are part of the FIWARE framework, namely: Orion Context Broker, IoT Agent, Mongo DB, actuators and sensors. For the sake of simplicity, only one device of each category is mentioned in the diagram, even though the architecture supports a larger number of devices.

The Jadex framework enables developers to implement agents following the BDI software model. At the start of the implementation phase of the project, version 3.0.117 of the framework was the most recent stable version with updated documentation. Meanwhile, version 4.0.241 was already available as a development version, although its documentation was not yet fully published; thus, in light of these factors, version 3.0.117 was selected for the implementation phase and, accordingly, so was Java SE Development Kit 8. Under this version, agents are fully implemented in Java according to a clear separation of beliefs, plans and goals that is made possible by Java's annotation mechanism. Capabilities are another useful feature of this framework, enabling the encapsulation of common functionality between agents into a single software component. In this way, agents can inherit beliefs, goals and plans from these components, which is very much in line with what had been introduced in Figure 3.4. These time, capabilities are clearly depicted in Figure 3.6 and associated with the respective agents, whose class names bear the *BDI* suffix, as required by the Jadex BDI V3 kernel. Together, *capabilities* and BDI V3 agents' *annotations* com-

[2] Jadex Active Components

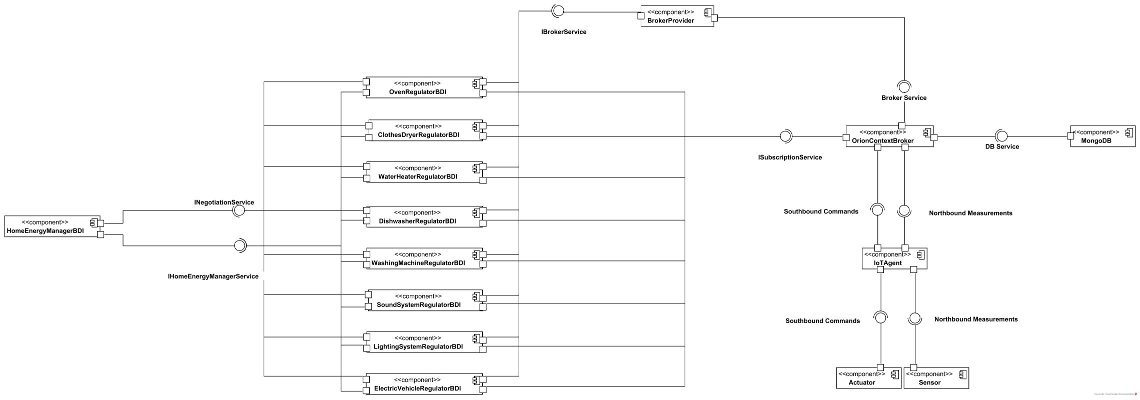


Figure 3.7: Component diagram of the proposed HEMS.

prise fundamental features for the implementation of BDI agents' reactivity and pro-activeness in Jadex.

On the other hand, social-ability in Jadex is ensured by its service-oriented architecture. Agents can interact with each other by offering services, that is, by implementing services defined via an interface that determines which methods are available for that purpose. This mechanism enables Appliance Scheduler agents to participate in the negotiation process introduced beforehand. The protocol mediator defined as part of this market process, or HomeEnergyManagerBDI, makes use of the service provided by the previous agents, namely, a negotiation service that allows requesting and accepting proposals, thus ensuring the flow of messages in both directions, as shown in Figures 3.6 and 3.7. It is precisely the implementation of this service that enables the previous agent to locate other agents participating in the market process. Conversely, the AS agents can also locate the HEM Agent provided that it implements the *IHomeEnergyManagerService* interface, which describes the service that enables them to request a renegotiation. Together, these two services fully describe the interactions between agents, as described in Section 3.2.4.

Apart from the interactions within the multiagent system, the scheduler agents have to communicate with the Orion Context Broker, a FIWARE component that lies outside said system. The Broker serves as the access point to the information collected by various sensors, which is stored in a Mongo DB database. In parallel, commands directed to actuators (e.g. lighting system, washing machine) also flow through this component. To enable scheduler agents to send these messages to the Broker, a new type of component has been programmed in Jadex - the *BrokerProvider*. The latter, which is not a BDI agent, offers a communication service with the Broker, which comprises the collection of actuator features (e.g. power consumption), sending commands and setting up subscriptions to state changes of the entities managed by the Broker; this paradigm thus conform with the Facade software design pattern given that scheduler agents do not engage directly with the Broker, but instead forward communication requests to the BrokerProvider, which abstracts the complexity of the code needed for outward communication. In this regard, HTTP is the transport protocol that enables the latter component, belonging to the MAS environment, to communicate outwards, as depicted in the deployment diagram in Figure 3.8. This feature is made possible by

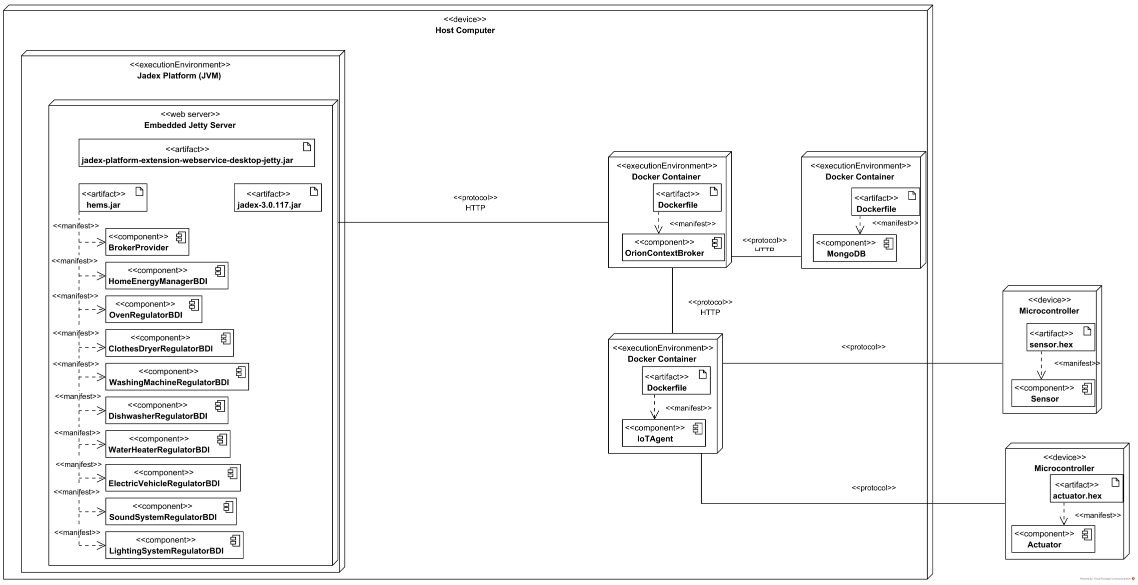


Figure 3.8: Deployment diagram of the proposed HEMS.

embedding the Jetty web-service library into the Jadex project, which is packaged as an add-on [3].

The Broker offers a well-defined NGSIv2 REST API [4]. Thus, to perform JSON-based REST calls, the BrokerProvider component relies on the RestInvocationHelper class, which is packaged along with the *extension* package. All FIWARE components except for the devices themselves reside in Docker containers, as shown in Figure 3.8. Actually, according to the FIWARE Foundation, it's recommended to run both the Broker and the MongoDB database as Docker containers, with the latter being linked to the Broker using a Docker Compose file; in addition, this is the fastest installation method. The Broker, however, can only interpret requests that make use of both HTTP as the transport protocol and NGSI-v2 as the message protocol. As a consequence, any device that is either not HTTP-compatible or not able to communicate using the NGSI-v2 message format will require a FIWARE IoT Agent that's tailored to its protocols to act as an intermediary (e.g. the IoT Agent for the Ultralight protocol). Therefore, it is up to the designer of the distributed system to determine which protocols should run between the IoT Agent and the devices; thus, the deployment diagram does not depict any particular protocol of this sort. It is also worth highlighting that this constitutes a reference deployment, which is by no means the only possible deployment solution. In fact, developers may consider running the various Docker Containers in separate devices or even run the Jadex platform as an extension to a JADE [Bellifemine et al., 2007] environment, thus allowing the BDI agents to be distributed across different devices.

Despite the selected distribution scheme for the deployment of artifacts among nodes, the flow of messages in the system will remain the same. Figures 3.9, 3.10 and 3.11 depict the message interchange between the various software components, of which the WashingMachineRegulatorBDI has been selected as representative of the Appliance Scheduler agents. The first indirect interaction

[3] The Jadex Jetty Web Service Extension is available as an add-on to the framework's *standard package*
[4] NGSIv2 REST API Specification

between the scheduler agents and the Broker follows the message interchange presented in Figure 3.9, thus corresponding to the request for device features, which are an indispensable piece of knowledge for this agent class. To achieve this, the request initiator needs to provide an identifier for the entity being tracked by the Broker, which should correspond to a physical device such as a washing machine, in the illustrated scenario. This information is to be included in the Uniform Resource Identifier (URI) corresponding to the HTTP request that is forwarded to the Broker, as given in Listings 3.1 and 3.2 that present Wireshark [5] captures performed on the deployed system.

Once the AS agents are aware of the characteristics of their equipment, they can begin to interact with them by issuing commands. This interaction is clearly depicted in Figure 3.10, which shows that the message exchange between the Broker and the Scheduler Agents upon issuing commands involves a third element - the IoT Agent. Agents will only issue these messages between time slots, to either disable or enable the corresponding appliance. In addition to the *status* key, agents should also provide a *power* key in the JSON payload, which is particularly important in the case of Curtailable appliances, which operate under variable power-consumption levels. In the meantime, messages may also flow in the opposite direction, from the Broker to the Scheduler Agents.

To achieve the latter interaction, agents must first subscribe to changes in the context of an entity of interest, such as a change in the value reported by a leakage sensor (Listing 3.3). It then follows that the Broker will notify the agent of relevant context changes by issuing HTTP requests to the endpoint provided at the time of the subscription setup request (Listing 3.4). Based on the deployment diagram, it is possible to infer that the embedded Jetty server is responsible for setting up these endpoints, so that each agent will receive one of these, allowing only the agent involved in the subscription to receive the applicable context update, as in Listing 3.4.

[5] Wireshark Network Protocol Analyser

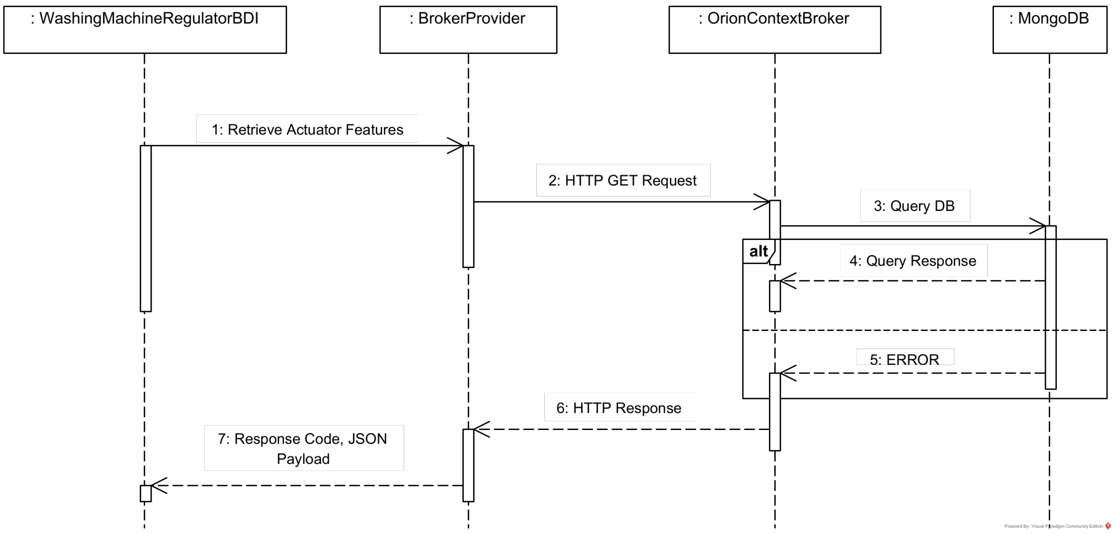


Figure 3.9: Sequence diagram: retrieve actuator features interaction.

```
GET /v2/entities/washing-machine-room-1/attrs HTTP/1.1
Accept: application/json
Fiware-Service: home
User-Agent: Jersey/2.11 (HttpUrlConnection 1.8.0_282)
Cache-Control: no-cache
Pragma: no-cache
Host: localhost:1026
Connection: keep-alive
```

Listing 3.1: HTTP request captured by Wireshark, which corresponds to the retrieval of features of a particular actuator device - the washing machine. The Broker keeps track of an entity with identifier *washing-machine-room-1*, so the latter is included in the URI.

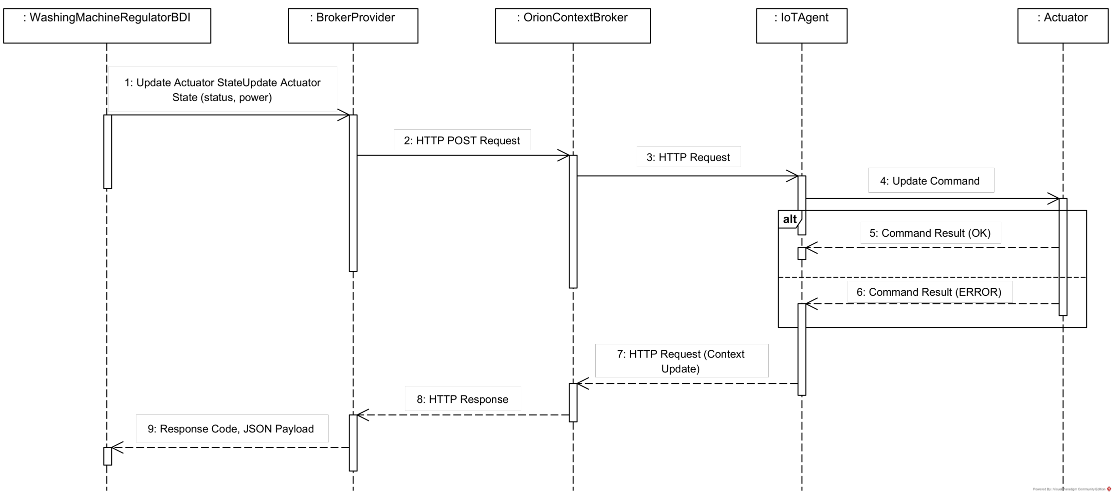


Figure 3.10: Sequence diagram: update actuator state interaction.

```
POST /v2/entities/washing-machine-room-1/attrs HTTP/1.1
Accept: application/json
Fiware-Service: home
Content-Type: application/json
User-Agent: Jersey/2.11 (HttpUrlConnection 1.8.0_282)
Cache-Control: no-cache
Pragma: no-cache
Host: localhost:1026
Connection: keep-alive
Content-Length: 48
----
{
   "status":{
      "value":"off"
   },
   "power":{
      "value":"0"
   }
}
```

Listing 3.2: HTTP request captured by Wireshark, which corresponds to the actuator status update. Updates are included in the payload of the request, which is represented below the dashed line according to the JSON format.

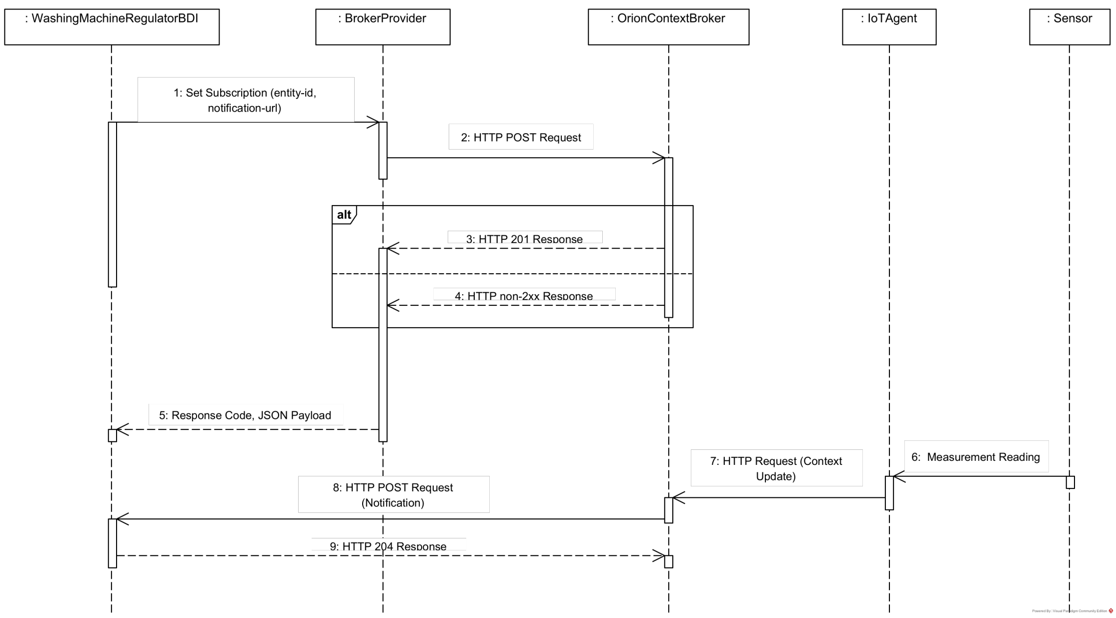


Figure 3.11: Sequence diagram: subscription setup and notification interactions.

```
POST /v2/subscriptions HTTP/1.1
Accept: application/json
Fiware-Service: home
Content-Type: application/json
User-Agent: Jersey/2.11 (HttpUrlConnection 1.8.0_282)
Cache-Control: no-cache
Pragma: no-cache
Host: localhost:1026
Connection: keep-alive
Content-Length: 153
----
{
   "subject":{
      "entities":[
         {
            "id":"room-1"
         }
      ],
      "condition":{
         "attrs":[
            "leaking"
         ]
      }
   },
   "notification":{
      "http":{
         "url":"http://172.17.0.1:8050/wmr/stateChange/"
      }
   }
}
```


Listing 3.3: HTTP request captured by Wireshark, which corresponds to a subscription set-up. The JSON payload informs the Broker about which notifications are of interest to the initiator of the request; in particular, the initiator is interested in being notified whenever a change occurs in the *leaking* attribute of the *room-1* entity. In addition, the notification should be sent to the host with IPv4 address 172.17.0.1 (corresponding to the address of the machine in the Docker network), port 8050 (that of the Jetty web server), and endpoint wmr/stateChange/, with wmr/ being the endpoint assigned by the server to the WashingMachineRegulatorBDI class agent and stateChange/ the route connecting to a state change handler.

```
POST /wmr/stateChange/ HTTP/1.1
User-Agent: orion/2.5.2 libcurl/7.29.0
Host: 172.17.0.1:8050
Fiware-Service: home
Fiware-Servicepath: /
Accept: application/json
Content-Length: 450
Content-Type: application/json; charset=utf-8
Fiware-Correlator: ab0b40ba-c94c-11eb-a7c8-0242ac120104; cbnotif=1
Ngsiv2-AttrsFormat: normalized
----
{
   "subscriptionId":"60c0fea2b14d7bc55e9226e7",
   "data":[
      {
         "id":"room-1",
         "type":"Room",
         "TimeInstant":{
            "type":"DateTime",
            "value":"2021-06-09T18:01:08.924Z",
            "metadata":{

            }
         },
         "leaking":{
            "type":"Integer",
            "value":"1",
            "metadata":{
               "TimeInstant":{
                  "type":"DateTime",
                  "value":"2021-06-09T18:01:08.924Z"
               }
            }
         }
      }
   ]
}
```

Listing 3.4: HTTP request captured by Wireshark, which corresponds to a notification issued by the Broker. The JSON payload includes the identifier of the subscription that triggered the notification and, most importantly, the state of the attribute under tracking. A *leaking* attribute with value *1* tells the agent that a leakage is currently underway.

3.5 Summary

This chapter has given an account of the proposed HEMS under three important perspectives: external, internal and architectural. The first two perspectives encompass the models that describe the multiagent system. The first one, a ResMAS-based model, examined the external interactions and roles of the agents and paved the way for in-depth modelling of agents' beliefs, goals and plans, which were gathered as a unified AgentSpeak(L) model. Finally, the architectural model introduced the software components developed as part of the dissertation project, and presented a reference deployment of the implemented artifacts. Finally, the chapter provided an in-depth description of the communication mechanisms supported by the distributed system at hand.

From the present chapter it is important to highlight the modular and layered design that characterises this architecture, making it highly adaptable to other smart-home oriented applications. New agents can be added to the multiagent system and benefit from the inheritance mechanism provided by the capabilities that have already been implemented. Agents' plans can also be extended to allow them to react to other unforeseen events that become known through FIWARE's notification mechanism. The two previous features create new opportunities for the development of software modules that can be freely shared and thus re-used by developers to create ready-to-use packaged applications. At the same time, this architecture does not impose any specific deployment of software artifacts, thus allowing developers and end-users to distribute them among nodes according to their requirements and constraints, which is especially important when deployment closer to actuator devices is not possible given usual processing and storage constraints, as mentioned in the previous chapter.

Chapter 4

Simulation Results and Analysis

This chapter aims at presenting the results and conclusions that stem from the simulations performed on the implemented Home Energy Management System. It begins by introducing the simulation methodology, the prerequisites and the metrics that have been evaluated, all under Section 4.1. The following sections introduce the results and the conclusions of the simulations, according to a specific organisational structure. Accordingly, the first sections relate to varying regulation criteria (Comfort, Bill, Green), among which further distinction is drawn between simulations concerning different energy tariffs. Afterwards, results of simulations referring to scenarios with unpredictability and with photovoltaic panels of different capacities are discussed. The chapter ends with Section 4.7, in which the results of all simulations are summarised and the main conclusions drawn from them are highlighted.

4.1 Assessment Method

In order to evaluate the feasibility and potential benefits of the proposed energy management system, several case studies were simulated and analysed. Particularly, these studies were conducted with the aim of answering the following questions.

- Can the proposed HEM solution help consumers select the tariff that best fits their energy consumption profiles?
- Can the proposed HEM solution contribute to raising awareness of consumers regarding their energy demands?
- Can the proposed HEM solution contribute to ease the decision making burden on the consumer side, regarding the scheduling of appliances?
- Can the proposed HEM solution readjust the daily energy consumption pattern of the Smart Home due to the occurrence of unforeseen events? What adjustments have to be made to the initial scheduling in order to react to such events?

Table 4.1: Parameters used to generate the PV's hourly-power dataset on the Renewables.ninja simulation tool.

Latitude	41.1494512
Longitude	-8.6107884
Month	July
Capacity (kW)	4
System Loss (fraction)	0
Tracking	0
Tilt (º)	35
Azimuth (º)	180

- Can the proposed HEM solution achieve a good balance between the overall comfort satisfaction and the satisfaction dictated by the selected regulation policy?

These studies were based on a Smart Home scenario composed of appliances of different categories - Interruptible, Uninterruptible, Curtailable - following the model introduced in the previous chapter. The characteristics of each of these appliances are defined according to the specific features of each category. For example, while Uninterruptible Appliances are characterised by a fixed power-consumption value, Curtailable Appliances are described by a power-consumption range. Table B.1 summarises the characteristics of these appliances, which are kept unchanged across all simulations. Furthermore, to fully qualify the Curtailable appliances, it becomes necessary to define additional parameters that allow the calculation of the power required to achieve a certain level of characteristic variable, as given by the AgentSpeak(L) model introduced earlier on. As a result, the bulbs that compose the Lighting System have a corresponding Luminous Efficacy level, while the Sound System has a corresponding Sensitivity Rating level. As for the Electric Vehicle, the simulations consider that it presents a charge value of 50% at the start of the day-ahead negotiation process. It is also important to note that no appliances were dropped from any simulation, meaning that a single Smart Home environment was considered throughout the simulations.

The Smart Home scenario also integrates a photovoltaic panel (PV) with a capacity of 4kW, meaning that the Smart Home can auto-generate a maximum of 4kW of power, that is, under ideal conditions (*peak sun*). The PV's hourly-power dataset was generated with the Renewables.ninja [1] simulation tool, using the parameters listed in Table 4.1 as reference. The result of this simulation was built into the HEM agent's knowledge base in the various case studies and is depicted in Figure 4.1.

Despite the availability of the photovoltaic panel in the Smart Home, its energy production capacity is not sufficient to meet the consumption needs of all the appliances that have already been introduced. Thus, the consumer has to rely on an energy supplier for the purchase of the remaining energy required to meet the total energy demand. Typically, energy providers advertise various

[1] Renewables.ninja

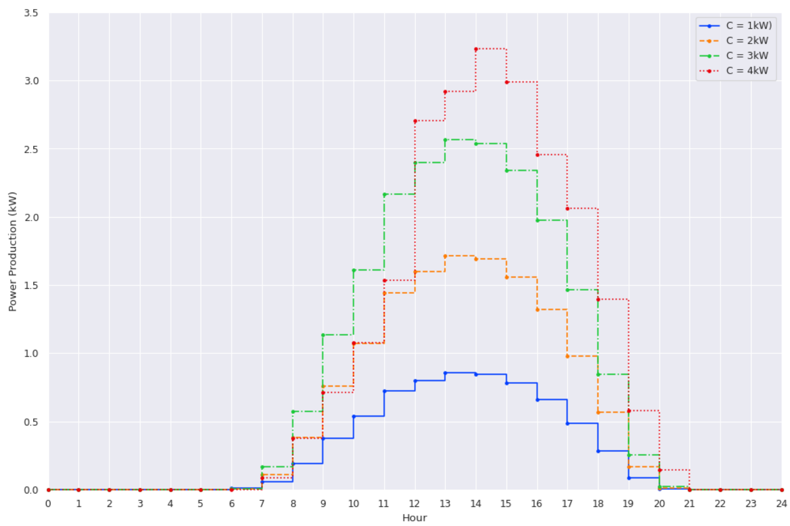


Figure 4.1: Hourly-power production of photovoltaic panels of different capacities, as considered in the simulations.

energy tariffs. As a general rule, these tariffs are characterised by the price of 1 kWh consumed per hour, as is the case for the tariffs of EDP, an energy company operating in Portugal. The tariffs used in the simulations, which are plotted in Figure 4.2, are based on the tariffs advertised by this energy supplier [2], being therefore: Simple, Dual-Rate and Triple-Rate.

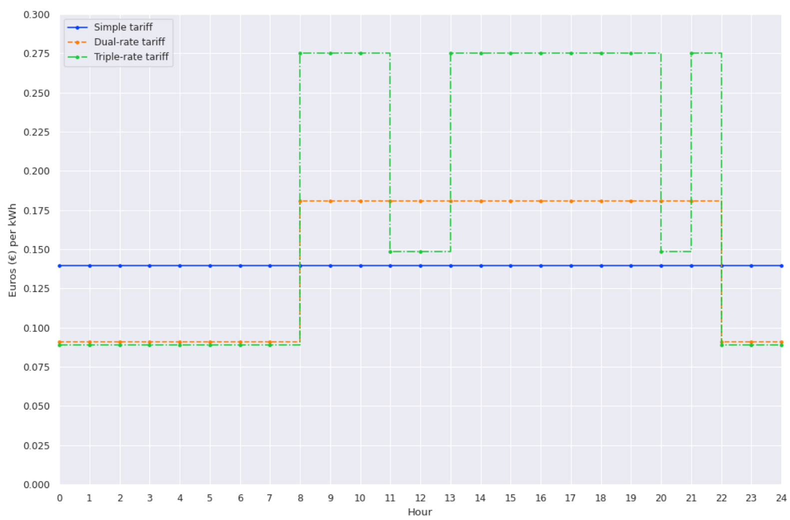


Figure 4.2: Hourly-tariffs considered in the simulations: Simple, Dual-Rate and Triple-Rate.

Additional restrictions are imposed on the Smart Home environment. First, the maximum peak power supported is set to 6kW. This information is included, together with the hourly prices of energy and the PV's hourly power production capacity, in the HEM agent's knowledge base. In addition, the latter agent is tailored to decrease the requested satisfaction level by 5% among consecutive iterations of the negotiation protocols (Anticipative and Emergency). However, in

[2]EDP's tariffs

order for the negotiations to be successful, the requested satisfaction can only be lowered up to a 50% satisfaction level, which is set as the critical satisfaction level of all the appliances, among all the simulations.

The variability of the case studies is ensured by changing the parameters over which the consumer has total control, namely the ones listed below.

- Regulation Policy (Green, Bill, Comfort)
- Tariff (Simple, Dual-Rate, Triple-Rate)
- Appliances' critical satisfaction levels
- Restrictions on the operation of household appliances
  - End-of-operation restrictions (Uninterruptible Appliances)
  - Period-of-operation (Curtailable Appliances)
  - Characteristic variables' restrictions (Curtailable Appliances)
  - Charge-level restrictions (Electric Vehicle)
  - Allowed charging-period (Electric Vehicle)
- HEM Agent's critical satisfaction level

The system's deployment follows a similar structure to the one of the deployment diagram depicted in Figure 3.8, under Chapter 3. As mentioned in this chapter, the distribution of the components among nodes is highly adaptable. In fact, given the unavailability of real-world devices, the execution of the simulations implied the emulation of all components related to actuators and sensors, using Docker containers. In sum, all components were deployed on the same machine, but were executed in isolation via Docker containers. The multiagent system, on the other hand, was executed in a single Java Virtual Machine (JVM). In addition, in order to answer the research questions, a set of evaluation metrics was established as follows.

- Peak Load: $P_{peak} = max\{P^t_{load} : t \in \mathbb{Z} \wedge t \in [0,23]\}$
- Total Energy Consumption: $E_{load} = \sum_{t=0}^{23} P^t_{load}$
- Energy taken from Grid: E_{grid} (3.19)
- Energy taken from PV: $E_{PV} = E_{load} - E_{grid}$
- Energy wasted from PV: $W_{PV} = \sum_{t=0}^{23}(P^t_{PV}) - E_{PV}$
- HEM's Satisfaction: satisfaction achieved by the HEM Agent (S_{HEM})
- Average Appliance Satisfaction: $\overline{S_{AS}}$ (3.21)
- Total Bill (euros): B (3.18)

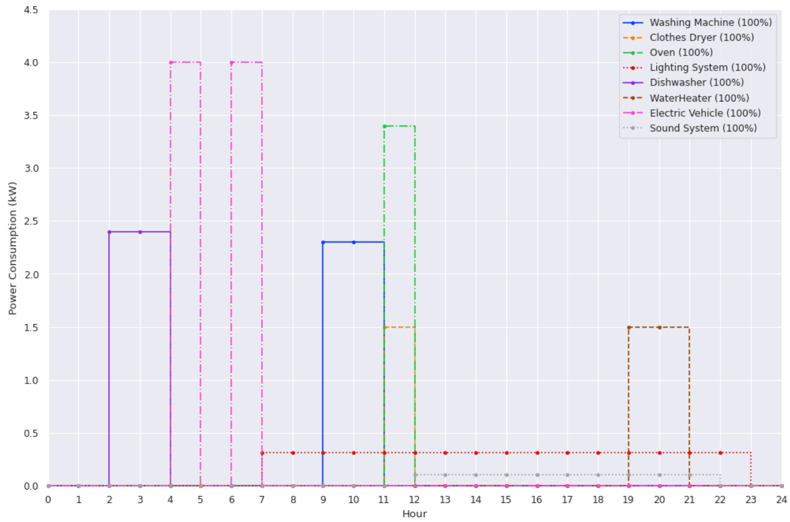


Figure 4.3: Appliances' scheduling under the comfort regulation scenario.

In addition, the scheduling results are further depicted through three different types of plots. The first, of which Figure 4.3 is an example, shows the power-consumption profile of each of the appliances over one day, as well as the satisfaction of the respective agent, which is presented in the legend. Each profile is shown in the form of a step function, indicating the power consumed at each 1-hour interval. From this plot, it is possible to derive the second plot, shown in Figure 4.4. The sum of the power consumed by each appliance at every hour leads to a step function portraying the total power being consumed at each moment in time. The orange-patterned areas depicted in the latter plot correspond to periods of a surplus of energy produced by the PV, while the blue-patterned areas represent those periods when there's a lack of PV-generated power to attend to corresponding power demands. The third type of plot, as in Figure 4.5, allows the consumer to have a better perception of the distribution of both expenses and power consumption. These are presented in the form of cumulative distribution functions where $y \in [0,1]$, as the left axis suggests. The green bars, on the other hand, depict the amount of money paid by the consumer for energy purchased from the grid at the respective time. In short, the three plots, together with the metrics mentioned above, served as the object of analysis for the undertaken simulations, resulting in the findings that follow in the next section.

4.2 Comfort Regulation Scenario

As of the first simulation, attention was drawn to the Comfort regulation policy. This proved to be a good starting point to gather *baseline* results. With this simulation, it was intended to observe whether it would be possible for the HEM Agent to achieve 100% satisfaction, thus reflecting a maximum satisfaction of all scheduler agents. A priori, this scenario may not be achievable by exceeding the maximum electrical-circuit power that may flow at any given time (set to 6kW).

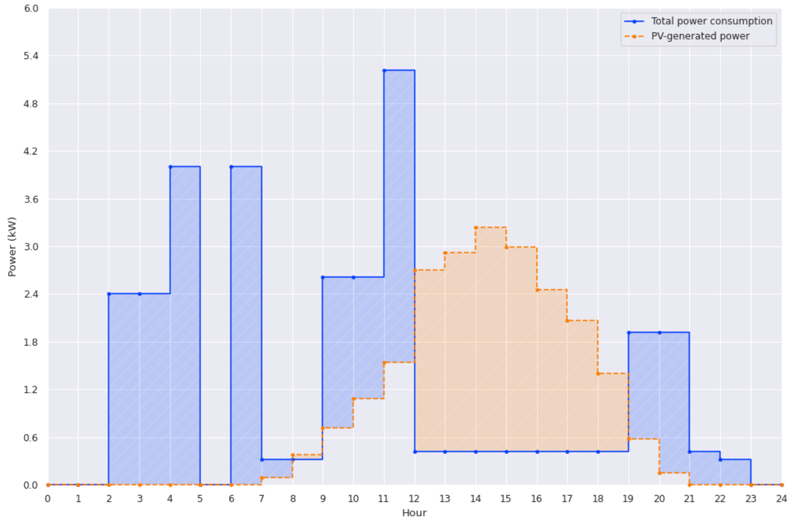


Figure 4.4: Total power consumption against PV-generated power under the comfort regulation scenario.

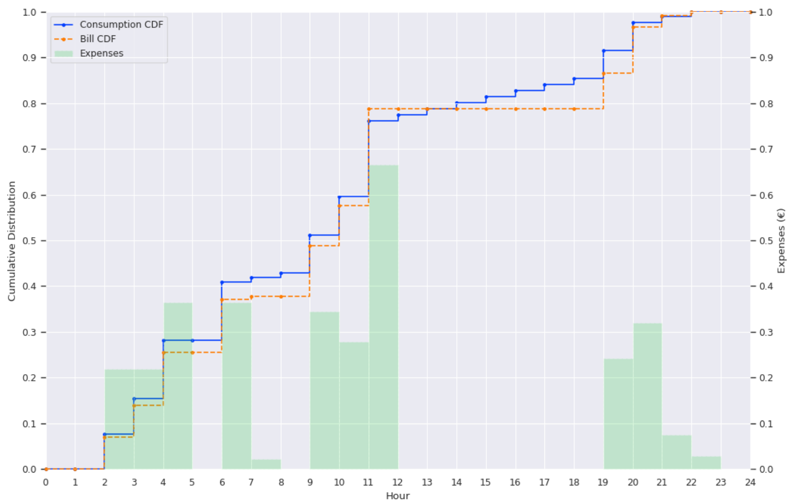


Figure 4.5: Cumulative distribution functions and hourly spendings under the comfort regulation scenario.

Table 4.2: Results of the Comfort Scenario simulation.

Peak Load (kW)	5.2130
Total Energy Consumption (kW)	31.3280
Energy taken from Grid (kWh)	23.9730
Energy taken from PV (kWh)	7.3550
Energy wasted from PV (kWh)	14.9180
HEM's satisfaction	100.0000
Average Appliance Satisfaction	100.0000
Total Bill (euros)	3.13869

For this simulation, a dual-rate tariff was adopted, together with a set of consumer preferences regarding the operation of appliances, as summarised in Table B.2.

The results obtained from this simulation are presented in Table 4.2. Figures 4.3, 4.4 and 4.5 also support the analysis of the results. It proved to be possible, given the consumer preferences introduced earlier on, to schedule the appliances in a matter that resulted in the maximum average satisfaction for all, a value that is equivalent, in the case of the selected regulation policy, to the satisfaction of the HEM Agent. In plot 4.3, which depicts the scheduling of all the appliances, the satisfaction of each of them is given in parenthesis in the legend.

The Comfort Regulation Policy is unconnected to the consumer's budget and also to the energy produced by the PV. For this reason, it is valid to assume that a scheduling resulting from this policy would attain the highest daily energy expenditure. In fact, when considering a dual-rate tariff, the costumer would have to be willing to spend around 3 euros a day to maximise its comfort, following an hourly distribution of these expenses as given in Figure 4.5.

The energy generated by the PV is also not well exploited, as only about 33% of the energy it generates is used to support the home's energy needs. Figure 4.4 provides a visual representation of this phenomenon. The remaining percentage corresponds to wasted energy, given that an energy storage system was not considered for this scenario. Thus, around 77% of the energy needs of the home would have to be satisfied by energy purchased from the provider.

4.3 Bill Regulation Scenario

This section outlines the results of the simulations performed under equal scenarios, which differ only on the tariff being employed: either Simple, Dual-Rate or Triple-Rate.

4.3.1 Simple Tariff

Through a cost-based regulation policy, the consumer may establish a contract covering one of three tariffs, as described in Section 4.1. In the first simulation of this type of policy, the Simple

Table 4.3: Results of the Simple Tariff Scenario simulations.

	Simple Tariff (C.S. 50%)	**Simple Tariff (C.S. 20%)**
Peak Load (kW)	5.32400	5.21300
Total Energy Consumption (kW)	31.43600	31.32800
Energy taken from Grid (kWh)	21.15700	23.97300
Energy taken from PV (kWh)	10.27900	7.35500
Energy wasted from PV (kWh)	11.99400	14.91800
HEM's satisfaction	50.77119	44.21882
Average Appliance Satisfaction	69.00000	100.00000
Total Bill (euros)	2.95373	3.34687

tariff was employed, such that the energy price remains at 0.13961€ per kWh throughout the day. Moreover, the consumer restrictions are set to 50% critical satisfaction, and the budget interval to $[0,6]$ euros, meaning that the consumer is willing to spend a maximum of 6 euros for the energy bought from the provider in the scenario of greatest compromise, that is, the one in which the critical satisfaction is zero.

As a result, the HEM Agent's satisfaction from the day-ahead negotiation with the Scheduler agents is set close to the critical satisfaction, as given in Table 4.3. This time, the scheduling of the appliances is more efficient in the sense that, compared to the previous scenario, about 50% of the energy generated by the panel is effectively used. Still, around 67% of the home's energy needs are supported by the energy purchased from the provider; therefore, the question that arises from this scenario is whether there's a scheduling that allows this consumption to be minimised. This problem is addressed by another regulation policy, the Green one, which is analysed in the following simulations.

Additionally, the previous scenario was adjusted, changing only the critical satisfaction of HEM Agent from 50% to 20%. As a result, the same appliance scheduling was obtained as in the comfort scenario where all appliances reach 100% satisfaction. However, a difference is noticed between this Comfort scenario and the new one, since the former considers a Dual-Rate tariff, while the present one considers a Simple tariff. It can be concluded that, according to the consumption preferences defined, the consumer would benefit from choosing a dual-rate tariff, as this would enable him/her to save around 20 cents. However, if the consumer is willing to compromise on comfort, for an average appliance satisfaction value of 69%, he/she will further save 18 cents, reaching a daily expenditure of less than 3 euros.

4.3.2 Dual-Rate Tariff

Currently, consumers can establish dual tariff contracts and schedule the operation of their electrical appliances (in general, those of higher consumption) for hours when the price of energy is

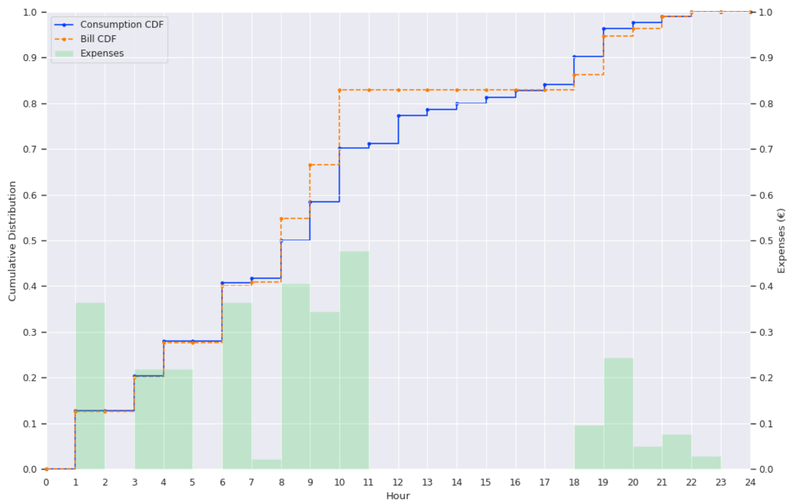


Figure 4.6: Cumulative distribution functions and hourly spendings under a scenario with Bill regulation, 50% critical satisfaction and Dual-Rate tariff.

lower. Taking as inspiration EDP's tariffs, simulations were also performed regarding this type of tariff, considering the following prices: 0.18090 euros/kWh and 0.09110 euros/kWh for peak and off-peak hours, respectively. All the other simulation parameters (including consumer preferences and appliance characteristics) were preserved in order to find out which tariff best fits the preferences of this type of consumer. As in the Simple Tariff scenarios, simulations were carried out with 50% and 20% critical satisfaction values.

The results show that the simulation with 20% critical satisfaction yields the same scheduling of appliances as in the results of the comfort regulation policy. However, as the Bill regulation policy is the one under analysis, this scheduling results in a satisfaction of around 48% for the HEM Agent as opposed to 100% under the comfort criteria.

The results obtained with a higher satisfaction value prove to be more interesting, namely because they show that by adopting a dual-rate tariff it is possible to achieve a scheduling that yields not only in lower daily costs for the consumer (2.90859 euros, as opposed to 2.95373 euros), but also in greater comfort. This reasoning leads to conclude that the latter result totally dominates those obtained under simple tariff conditions.

It is also worth noticing that the present case study, despite taking less advantage of the energy generated by the PV, still enables the lowest daily costs. The reason behind this result has to do with the off-peak prices. According to EDP's recommendations, consumers should consider this pricing strategy whenever the off-peak period alone accounts for more than 40% of the total energy consumption. The present simulation proves that it is possible to achieve a power consumption profile that satisfies this condition, as Figure 4.6 suggests. As a result, the Dual-Rate tariff proves to be monetarily beneficial. Nevertheless, the results of the Simple tariff simulation already pointed to the potential benefit of the second tariff, given that the same condition could already be observed under that scenario.

Table 4.4: Results of the Dual-Rate Tariff Scenario simulations.

	Dual-Rate Tariff (C.S.: 50%)	**Dual-Rate Tariff (C.S.: 20%)**
Peak Load (kW)	4.00000	5.21300
Total Energy Consumption (kW)	31.43600	31.32800
Energy taken from Grid (kWh)	22.70400	23.97300
Energy taken from PV (kWh)	8.73200	7.35500
Energy wasted from PV (kWh)	13.54100	14.91800
HEM's satisfaction	51.52345	47.68844
Average Appliance Satisfaction	71.12500	100.00000
Total Bill (euros)	2.90859	3.13869

4.3.3 Triple-Rate Tariff

According to EDP [3], the triple-rate tariff is best-suited for those consumers who do not need to purchase energy from the grid in periods when energy prices are the highest, and who schedule more than 20% of their energy consumption for the off-peak period, when the price of energy is at its lowest (0.0889 euros/kWh).

Again, new simulations proved that the consumer under consideration can still lower his/her daily expenses by scheduling the energy needs according to the latter tariff. When setting the HEM Agent's critical satisfaction to 50%, the MAS derives a final schedule that is equal to the one obtained under the Simple tariff scenario, also with 50% satisfaction. This time, however, the

[3] EDP's recommendations

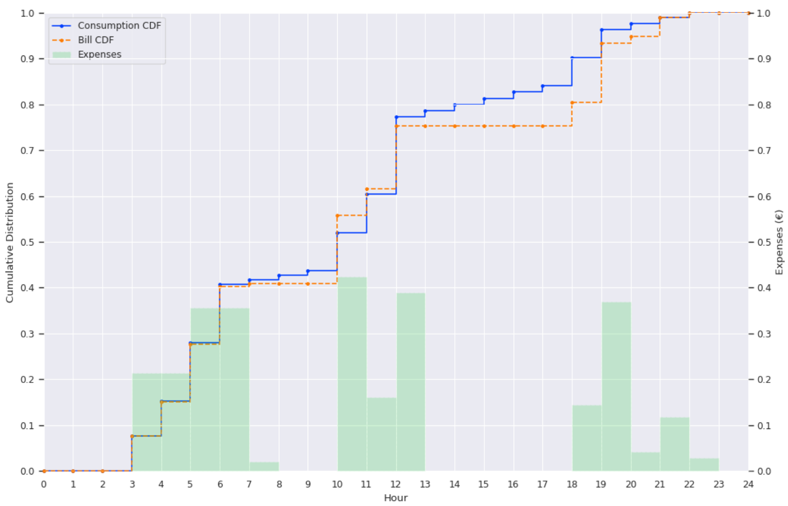


Figure 4.7: Cumulative distribution functions and hourly spendings under a scenario with Bill regulation, 50% critical satisfaction and Triple-Rate tariff.

Table 4.5: Results of the Triple-Rate Tariff Scenario simulations.

	Triple-Rate Tariff (C.S.: 50%)	**Triple-Rate Tariff (C.S.: 20%)**
Peak Load (kW)	5.32400	5.21300
Total Energy Consumption (kW)	31.43600	31.32800
Energy taken from Grid (kWh)	21.15700	23.97300
Energy taken from PV (kWh)	10.27900	7.35500
Energy wasted from PV (kWh)	11.99400	14.91800
HEM's satisfaction	52.80658	42.97388
Average Appliance Satisfaction	69.00000	100.00000
Total Bill (euros)	2.83161	3.42157

consumer would only have to spend 2.83161 euros instead of 2.95373 euros. In addition, it can also be observed that the consumer does not need to stop consuming in the peak period. Period [8, 11] will count for 15% of the total bill, period [13, 20] will count for about 20% and period [21, 22] for less than 5% (Figure 4.7). By changing the critical satisfaction level, again, to 20%, the average appliance satisfaction also increases to 100%. However, the selected tariff is no longer beneficial, as suggested by the HEM Agent's satisfaction level (42.97388 %) which is the lowest level compared to the levels attained by previous simulations under the same conditions.

4.4 Green Regulation Scenario

This section now describes the simulation results obtained with a *Green* regulation policy, together with a Dual-Rate tariff. This time, the interval of admissible regulation policy output had to be adjusted to reflect the amount of energy that the consumer is willing to acquire from the provider. Thus, the latter interval is set to $[0, 30]$ kWh, across all simulations.

Earlier results, gathered from a simulation with critical satisfaction value set to 30% yielded very interesting insights from the standpoint of this solution's contribution towards sustainability. Up to the point where the simulations now referred to would have been carried out, the results portraying to the daily energy consumed from the grid had reached a minimum of 21.15 kWh, while the *Green* simulation yields a total of 20.982 kWh. Interestingly, the total bill reached with the new scheduling is even lower than the one attained under the *Bill* regulation policy in the previous simulations, following all three tariffs. When compared to the results of the Dual-Rate Bill strategy, the new scheduling could be interesting to the consumer if he/she was willing to give up a small percentage of comfort (in fact, just 4%).

When the consumer lowers the critical satisfaction value of the HEM Agent by 5%, as in the second simulation, the average appliance satisfaction increases, as expected, leading to an increase in the energy consumed from the grid. After reflection on this matter, it was concluded

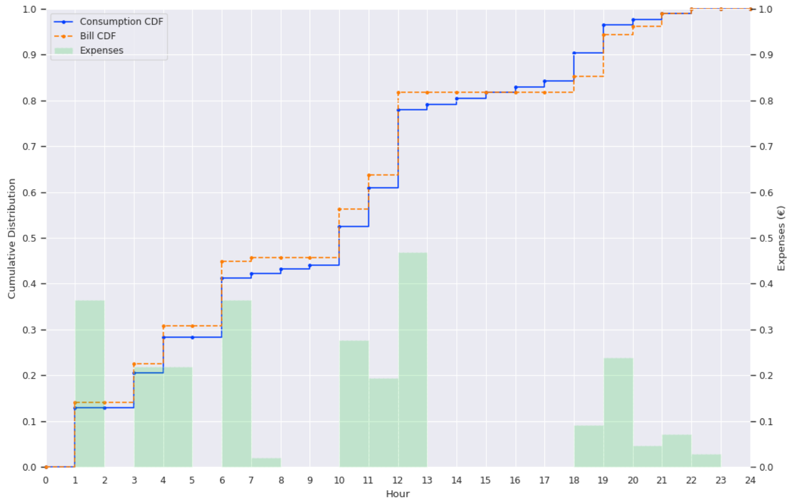


Figure 4.8: Cumulative distribution functions and hourly spendings under a scenario with Green regulation, 30% critical satisfaction and Dual-Rate tariff.

that a different result could be obtained if the energy from the panel were better exploited, given that there is still an energy surplus from this source. However, it turns out that it is not possible to obtain this result since consumer preferences regarding the hours of operation of domestic appliances are quite strict. The fact that curtailable appliances are practically the only ones to reduce their satisfaction also fosters this finding.

Plots such as in Figure 4.8 prove to be of great benefit to the consumer. By analysing the patterned areas of these plots it is possible to have a better perception of possible changes in consumer preferences conducive to more sustainable scheduling according to the Green policy. Thus, it is possible to emulate the behaviour of the consumer when confronted with the scheduling resulting from the first Green simulation (when the critical satisfaction is set to 30%). Starting from the latter scheduling of appliances one may notice that there is a predominance of the orange-patterned area in the $[13, 18]$h period, which means that the energy generated by the PV is not being well exploited in that period. Based on this observation, it is assumed that the consumer makes changes to his/her preferences, as shown in Table 4.6. In addition, the HEM Agent's critical satisfaction is raised to 40%.

The results depicted in Table 4.7 reveal that changes in the consumer's preferences do have a meaningful impact on three critical values: the amount of energy wasted from the PV, the energy taken from the grid and the total energy consumption of the Smart Home. This result is encourag-

Table 4.6: Changes applied to the consumer's preferences in the case of the Green regulation scenario.

	Critical Satisfaction	**EET**	**RET**	**LET**
Clothes Dryer	50	14	17	18
Electric Water Heater	50	14	15	18

Table 4.7: Results of the Green Scenario simulations.

	Green Scenario (C.S.: 30%)	Green Scenario (C.S.: 25%)	Green Scenario (changed preferences)
Peak Load (kW)	5.29700	5.32400	4.00000
Total Energy Consumption (kW)	31.10600	31.43600	30.91000
Energy taken from Grid (kWh)	20.98200	21.15700	17.98600
Energy taken from PV (kWh)	10.12400	10.27900	12.92400
Energy wasted from PV (kWh)	12.14900	11.99400	9.34900
HEM's satisfaction	30.06000	29.47667	40.04667
Average Appliance Satisfaction	67.12500	69.00000	67.62500
Total Bill (euros)	2.59888	2.62874	2.05798

ing as it proves that the proposed solution has the potential to raise consumers' awareness of their energy consumption profiles and the respective impacts those profiles have on the output of the various regulation strategies. In particular, it proves that this solution has the ability to foster more sustainable energy consumption behaviour.

4.5 Reaction to unpredictability

This section outlines the results of the simulations considering unpredictable events. In particular, they illustrate two different types of events, that is, those that last for a certain period of time (*enduring events*) and those that provoke an effect to which Appliance Scheduler agents can react immediately (*one-shot events*).

4.5.1 Enduring Events

The scenario that served as a starting point for the simulations of enduring events comprised the following parameters: a Dual-Rate tariff combined with a Bill regulation-policy, and critical satisfaction set to 20 %. Under this scenario, a day-ahead scheduling had already been obtained via the Anticipative mechanism (as given in Figure 4.9), and results had been collected in Table 4.4. Both results would serve as a benchmark for the ones obtained through the Emergency mechanism.

As explained under Chapter 3, Scheduler agents are able to trigger the negotiation protocol via the Emergency mechanism when they realise that their energy-consumption profile, obtained via the Anticipative mechanism, has been compromised due to unexpected events that occur throughout the day. To illustrate this scenario, the occurrence of a leakage in the room where the washing machine is located was selected as an unexpected event. In this simulation, the leakage starts at 7:30h and ceases at 15:00h, which directly interferes with the time stipulated for running the washing machine [9, 11]h. For this reason, the WMR Agent requests a new negotiation at 3 pm, starting from new satisfaction parameters provided by the consumer (Table 4.8).

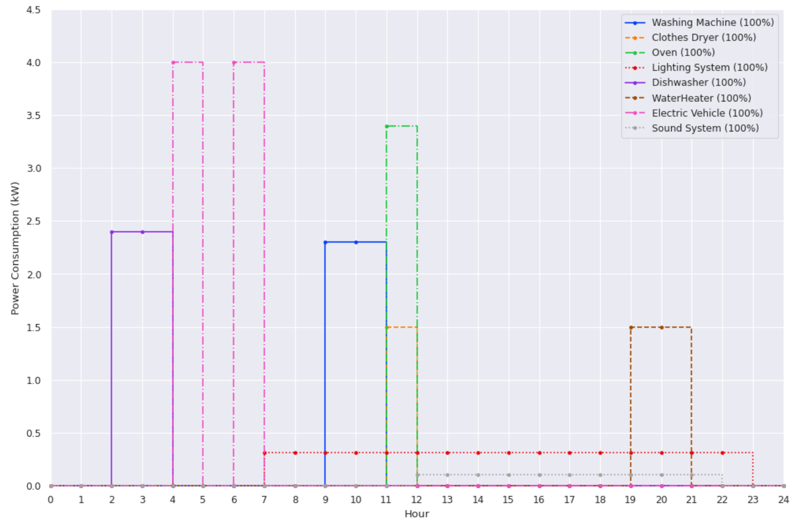


Figure 4.9: Appliances' scheduling under the scenario with Bill regulation, 20% critical satisfaction and Dual-Rate tariff.

Interestingly, agents are effectively able to reach a new agreement, however, this is not always possible and may lead the consumer to give in on his/her preferences in regards to either the Scheduler Agents or the HEM Agent. In this scenario, however, rescheduling is a viable option but incurs less satisfaction for the HEM agent since rescheduling forces an extra payment of 0.83214€ in relation to the costs initially foreseen by the Anticipative mechanism. Clearly, the extra costs could be reduced taking better advantage of the energy generated by the PV during the [16,19]h period, but this result enables Scheduler Agents to attain their maximum comfort value. Additionally, it is important to take into account that in relation to the initial scheduling, no other appliance saw its satisfaction being decreased. However, under more strict scenarios, this is rarely the case.

In an attempt to force the system to obtain a third scheduling, following the previous one, a second unforeseen leakage was triggered, this time affecting the water-heater in the period [17.5, 20]h. This time, Table 4.9 reveals that the third scheduling comes even closer to the satisfaction threshold defined by the consumer (20%), which corresponds to a 24% decrease in satisfaction compared to the day-ahead scheduling. In contrast with the result of the first emergency negotiation, agents are also unable to maximise their satisfaction, with WMR and WHR seeing their

Table 4.8: Periods of occurrence of simulated unexpected events (leakage) and changes in consumer preference towards the washing machine (first event) and the water-heater (second event).

	Start of Event	End of Event	Updated EET	Updated RET	Updated LET
First unexpected event	7:30h	15:00h	18	22	24
Second unexpected event	17:30h	20	22	24	24

Table 4.9: Results of each Emergency negotiation triggered by two unexpected events.

	First Unexpected Event	Second Unexpected Event
Peak Load (kW)	5.21300	5.21300
Total Energy Consumption (kW)	35.92800	41.24000
Energy taken from Grid (kWh)	28.57300	33.88500
Energy taken from PV (kWh)	7.35500	7.35500
Energy wasted from PV (kWh)	14.91800	14.91800
HEM's satisfaction	33.81944	23.49558
Average Appliance Satisfaction	100.00000	86.25000
Total Bill (euros)	3.97083	4.59027

satisfaction reduced by half, while curtailable appliances also give way in their degree of satisfaction, albeit less.

4.5.2 One-Shot Events: Consumer changes preferences

Another unexpected event that has not yet been mentioned is that of changing consumer preference. As in the latter scenarios, agents should react to changes in consumer preference with respect to the appliance they operate and seek re-scheduling if necessary. This is expected to happen when the agent recalculates its satisfaction value as a function of the current scheduling and the new satisfaction criteria, thus resulting in a value that is lower than its critical satisfaction value. To test this reaction mechanism, a new simulation took place, in which the consumer changes his preferences at 7:30 am, as summarised in Table 4.10. Again, the baseline scenario is the same as in the other simulations conducted within this section.

Remarkably, the new results, shown in Table 4.11, reveal that HEM Agent's satisfaction increases with the second negotiation, in contrast to what had happened in all other simulations of unexpected events. At first, it contradicts what initial intuition may suggest, given that preferences force the water-heater to operate at a period of higher energy consumption, as the plots of the initial scheduling show. Nevertheless, the new scheduling allows taking greater advantage of the energy generated by the panel, thus relieving the dependence on energy from the provider. This results in a reduction of 3 cents in daily energy expenses, and a consequent increase in satisfaction on the part of the HEM Agent. However, the associated costs in terms of reduced comfort (around

Table 4.10: Changes applied to the consumer's preferences midday.

	Critical Satisfaction	EET	RET	LET
Water Heater	50	10	12	14

Table 4.11: Results of the scenario with midday preference change.

	Midday preference change scenario
Peak Load (kW)	5.21600
Total Energy Consumption (kW)	31.43300
Energy taken from Grid (kWh)	23.83100
Energy taken from PV (kWh)	7.60200
Energy wasted from PV (kWh)	14.67100
HEM's satisfaction	48.12106
Average Appliance Satisfaction	75.87500
Total Bill (euros)	3.11274

24%) may not please the consumer, who is left to decide whether to keep the original schedule or replace it with the newer one.

4.6 Varying PV Capacity Scenarios

As a final study, new simulations were carried out, this time varying only the capacity of the photovoltaic panel. Starting from a scenario in which no panel is integrated into the Smart Home, it evolves on to a scenario with a panel of 1kW capacity, which undergoes a successive increase in capacity by 1 unit of measurement among consecutive simulations, up to a maximum value of 4 kW. This results in a range of capacities that are quite common in the domestic environment. These simulations were performed in order to understand if the HEM system could help the consumer to decide (1) if he/she should buy a panel or not and (2) in the first case, even in the selection of a panel adjusted to his/her needs.

Table 4.12: Results of simulations with varying PV capacity.

	No PV	**1kW**	**2kW**	**3kW**	**4kW**
Peak Load (kW)	5.98400	5.98800	5.98100	5.25900	5.27300
Total Energy Consumption (KW)	30.44400	30.54800	30.36600	30.63600	30.81200
Energy taken from Grid (KWh)	30.44400	25.82000	23.13800	20.79400	20.82600
Energy taken from PV (KWh)	0	4.72800	7.22800	9.84200	9.98600
Energy wasted from PV (KWh)	0	1.96900	6.14800	10.22300	12.28700
HEM's satisfaction	28.21878	42.08830	50.06976	57.08155	57.12875
Average Appliance Satisfaction	51.87500	53.75000	50.00000	55.62500	61.50000
Total Bill (euros)	4.30687	3.47470	2.99581	2.57511	2.57228

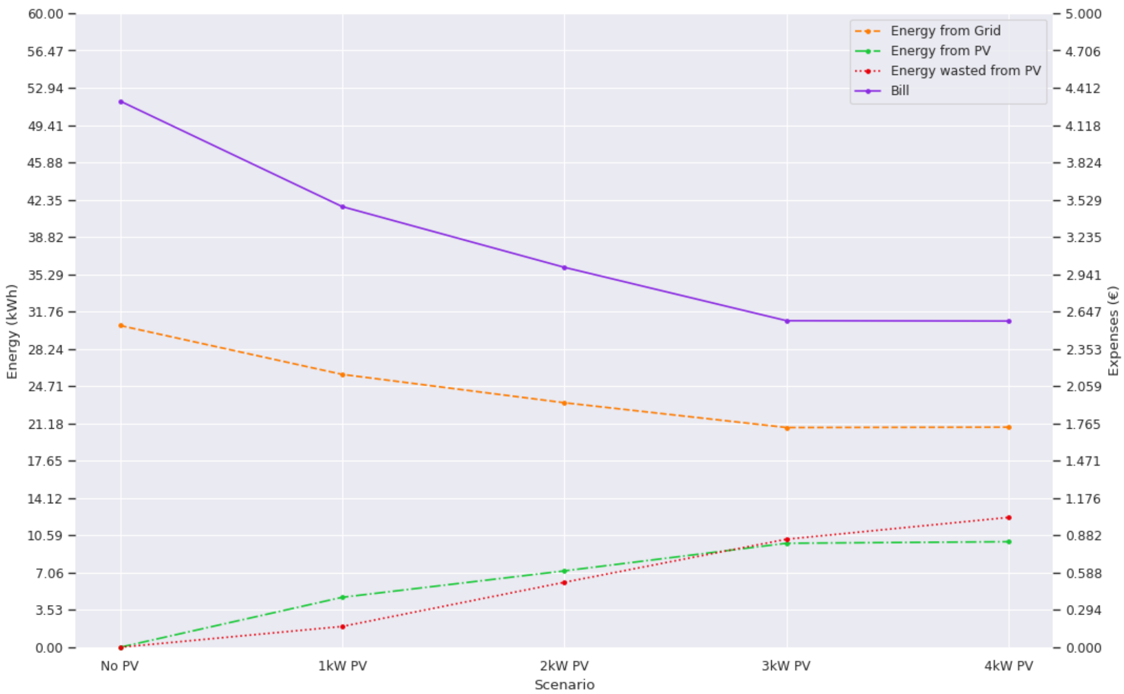


Figure 4.10: Comparison of results among simulations with varying PV capacity.

In these simulations, the input parameters of the first ones were preserved, namely, the characteristics of the appliances (Table B.1) and the consumer preferences (Table B.2). The focus was placed on evaluating the results of the day-ahead negotiation process only, so the Emergency Mechanism was not triggered. Furthermore, in contrast to previous simulations, the critical satisfaction value in all simulations was set to the maximum value that allowed to derive a feasible scheduling solution. As a result, this decision made it possible to assess the maximum savings that the consumer could achieve by switching from one panel to a higher capacity one. It thus follows that the regulation policy established was the *Bill* one, in association with a Dual-Rate tariff.

The results of the metrics under analysis were gathered in Table 4.12, however, their analysis is made simpler via the plot depicted in Figure 4.10. Immediately, a striking result suggests that the increase in panel capacity leads to successively greater savings, as shown by the uppermost line in the plot. This outcome is also expressed by the HEM Agent's satisfaction, which is presented in the table mentioned above. The underlying cause of this result becomes clear from the analysis of the amount of energy extracted from the grid; in fact, the plot shows that increasingly higher capacities lead to increasingly lesser energy purchases, as more energy is harnessed from the PV. This trend also undergoes successive attenuation, i.e. the savings obtained between consecutive simulations are always diminishing, which suggests that the savings resulting from the replacement of a 3kW PV by a 4kW or even a 5kW one may not be worth the investment. As of the overall comfort, Table 4.12 shows that there is a trend towards increased comfort in parallel with increased savings, however, this increase cannot be guaranteed, as the 2 and 3 kW simulations prove. Finally, one may also observe a successive increase in the panel's wasted energy. Effectively, with the increase of available PV energy, it will be expected that the consumer will change his/her scheduling preferences, allowing better use of the energy surplus; however, this was not analysed in these simulations, thus leading to this effect.

4.7 Summary

In this chapter the results of several simulations conducted on the implemented HEMS were introduced. These simulations explored the impact of the selected regulation policy, tariff, consumer preferences and even the PV's capacity on various metrics relating to comfort, sustainability and expenses. A critical analysis of these results calls into question whether or not having a large set of parameters to be adapted by the homeowner may constitute a barrier to the system's adoption. Nevertheless, the benefits brought up by the system, namely in terms of easing the decision making burden, suggest that this drawback is insignificant.

Among these simulations, the ones considering varying tariffs in the scope of a Bill regulation policy show that adapting the critical satisfaction value ($C.S.$) of the HEM Agent leads to the decrease of expenses, namely of 4% maximum decrease of expenses with $C.S. = 50$, and 6% when $C.S. = 20\%$. These results may not sound surprising at first as they revolve around a daily scheduling of appliances, however adding these savings over a longer period of time, such as a month, would result in significant savings for the homeowner. In addition, it was also observed that the reduction of expenses generally comes at the cost of decreased comfort, but this is not always the case. To overcome this problem, the system could help the consumer to (1) select a tariff that is best suited to his/her preferences or (2) adapt or loosen these preferences according to the tariffs under contract so that the system can achieve a schedule that takes more advantage of off-peak hours or increases the use of energy generated by the PV.

As of the Green regulation scenarios, the most encouraging results revealed that raising homeowner's awareness of energy needs can lead to the decrease in (1) the amount of energy wasted from the PV (2) the amount of energy taken from the grid and (3) the total energy consumption of the Smart Home. Furthermore, the homeowner can attain these benefits with little impact on comfort.

The results concerning the emergency mechanism have primarily shown that reacting to unforeseen events (1) doesn't necessarily lead to lower comfort among successive negotiations and (2) often leads to the requisition of more energy from the grid between successive re-negotiations, thus raising daily energy expenses. However, if the homeowner is aware of the distribution of the PV's power over the day, this can be exploited and result in lower expenses when compared to that of the day-ahead scheduling of appliances. Moreover, the simulations have also demonstrated that the system can help the homeowner in selecting a PV with capacity suited to his/her needs, which has reinforced the potential of the system to serve as a decision support tool.

Chapter 5

Conclusions and Future Work

This work presented an alternative HAS architecture to those proposed to date, which suffer from several problems. On the one hand, these systems tend to be strict, making it difficult to integrate new applications and services. It also happens that several of the proposed HASs do not contemplate access to remote services, focusing on the development of isolated control systems that do not benefit from data gathered outside the home environment. On the other hand, several architectures are still being proposed whose components portability is dubious or even unfeasible, raising scalability concerns. However, among these three missing requirements, interoperability is still the most challenging one, extending to other IoT sub-domains besides Smart Homes.

Bearing in mind the previous problems, this work introduced an alternative HAS architecture based on the integration of a multiagent system layer with the FIWARE middleware. Autonomy and control are guaranteed by the multiagent system that supports various deployment architectures of software agents. As the MAS distributes computational resources and capabilities across a network of interconnected agents, it is possible to overcome the problems inherent to centralised systems, particularly in terms of poor scalability coupled with hard-to-serve requirements imposed by intensive computation tasks. Moreover, it also allows for greater adaptability, which can be achieved by adding agents to the system offering new services (with the potential to be deployed across various computational nodes) or by adding new capabilities to existing agents. Through the implementation of agents as software components, the HAS becomes extensible and flexible, meaning it can be adapted according to the needs and constraints of each Smart Home. On the other hand, the FIWARE middleware - whose component architecture resides in the lower layers of the HAS - makes it possible to overcome the interoperability problem, thus guaranteeing the upper layer abstraction over IoT devices. Furthermore, the middleware provides several services to the multiagent system, namely data access, command flow and subscription setup.

Following the architecture proposal, this work introduced a HEMS model compatible with this architecture, with the aim of illustrating a system for resource management. The model was decomposed into three complementary sub-models. Firstly, the multiagent system was specified from an external viewpoint, thus providing a clear description of the resources, markets, participating agents and their interactions, following the ResMAS methodology. Then, the multiagent system was broken down into individual BDI agent classes that were examined according to their beliefs, goals and plans and described in accordance with the AgentSpeak(L) semantics. Finally, the architecture of the HEMS was presented in the form of several UML diagrams: class diagram, component diagram, deployment diagram and several sequence diagrams. These, in turn, served as reference for the implementation and deployment of the system.

Once the system was implemented, a realistic case study was developed, involving the specification of the scenario, needed datasets, metrics to be evaluated and tests of interest. These tests were then conducted via simulation, and the results were collected according to the metrics established beforehand. Analysis of the simulation results led to findings that confirmed the contribution of the HAS solution towards comfort and sustainability, which posed the main motivation for this study. Among them, the proof that the system can aid the consumer in decision making, raise homeowners' awareness as of their resource consumption profiles, and provoke behavioural changes leading to more sustainable consumption patterns stand out as the most significant and encouraging findings.

5.1 Main Contributions

In summary, the contributions of this work fall into three domains, as outline below.

- **Scientific** Following the Systematic Mapping Study, this work identified a research gap in the domain of HASs, which also served to reinforce some of the challenges that had been pointed out by previous works. A proposal for a reference HAS architecture was also presented, and simulations were performed on a system built according to this architecture; from these simulations resulted encouraging findings, which proved the contribution of the proposed HASs towards comfort and sustainability. Interestingly, this work appears to be among the first to integrate the MAS paradigm with the FIWARE middleware.

- **Applicational** The HAS architecture was put to the test through the design and implementation of a Home Energy Management System.

- **Technological** This work delivered a set of software modules, which can easily be adapted so that the multiagent layer provides new services, namely in terms of resource management.

In addition, these contributions have encouraged the writing of scientific articles, among which one has already been accepted, as follows [Martins et al., 2021].

> Martins, S., Rossetti, R. J. F., and Kokkinogenis, Z. (2021). A Resource-Based Model for Home Energy Management Using Multi-Agent Systems. In IEEE International Smart Cities Conference (ISC2).

5.2 Limitations and Lessons Learned

Despite the achievement of all the objectives defined during the first phase of the dissertation, this work has some limitations, which are worth mentioning. Firstly, the method for validating the architecture was based on simulation, and these simulations were not performed on a proper simulation software that would allow for more extensive testing (e.g. in terms of varying scenarios, evaluation metrics and features of IoT devices, just to name a few); precisely, this was due to the fact that there is still no such tool that allows simulating and adapting the components of this architecture at all layers, i.e., from the multiagent system to the IoT devices. On the other hand, looking at the architecture itself, it becomes clear that the FIWARE Broker constitutes a single point of failure, which implies that the functioning of the HAS as a whole is dependent on this component. Further dependencies are noticeable between the Broker and the software agents, namely in what concerns the tight coupling between the context data semantics and the agents' semantic interpretation. This becomes apparent if one changes the original semantics of the context data, as agents will be unable to both extract data from subscriptions as well as set new ones or even deploy appliance-directed commands. In this regard, sticking to a well-defined data model, such as the FIWARE data models [1], can alleviate this problem but it does not serve as an all-in-one solution. Finally, it would also be desirable for the implemented HEMS system to take into account the existence of preference hierarchies, so that the consumer could define which agents' satisfaction should take precedence over others.

As for future lessons, this work revealed that there are still many challenges as to the agentification of the IoT. On the one hand, it stems from this work that there's still underdevelopment of standards for the interoperability of multiagent systems with web services - especially with RESTful web services, which have become predominant in recent years. Exposing agents as web-services and enabling agents to consume cloud-based services requires (1) software libraries that provide straightforward integration of web-services with MAS (2) modelling languages that allow for the full representation of MAS-based systems without the need to represent those over different models, as happened in this work (3) software engineering methodologies that cover the whole software development process, including web-service integration (4) IoT semantics standards to solve the semantic interoperability problem that arises when agents access cloud-based services. Finally, the complexity involved with modelling deliberative agents suggests that the adoption of systems of BDI agents in the scope of HASs may be called into question. As an alternative, reactive architectures may be explored instead - when agents simply map sensory input to action selection - as well as hybrid ones.

5.3 Further Developments

This work as raised many opportunities for further development. These include both developments aimed at overcoming the limitations mentioned in the previous section, as well as those that would

[1] FIWARE data models

expand the proposed architecture and/or the already deployed HEMS. Among all of them stand out those listed below.

- Model and implement as part of the HEMS a set of agents whose information available at the time of the negotiation process is imprecise. In the presented multiagent system, all agents were able to compute with precision the value of power required to achieve a certain satisfaction value. In cases where this is not possible, as in the case of air conditioning systems for which the characteristic variable (room temperature) is influenced by uncontrollable factors, it is necessary to calculate predicted set points. In this sense, a possible alternative would be to apply machine learning methods to make this prediction.

- Implement a model that predicts the power generated by the photovoltaic panel and include this feature as part of the HEMS Agent component. The model can take as input meteorological data acquired from RESTful web services.

- Develop a HEMS application serving as an interface for monitoring and control of the system by the homeowner. Various human-computer interfaces can be explored as part of this work.

- Develop new regulation policies to be implemented as part of the HEMS Agent. In particular, it would be interesting to explore regulation policies that take into account preference hierarchies (e.g. *lightning system* over *washing machine*) , as outlined in the previous section.

- Deploy new systems in parallel with the HEMS. For example, one could develop a Water Management System or a Security Management System as part of the global HAS. This work is quite challenging and may involve the study of control hierarchies given the potential for intersection of the sets of appliances involved in these sub-systems. At the same time, as a result of more services being integrated into the HAS, one may expect agents' satisfaction, short-term and long-term objectives to conflict. The latter hypothesis is worth investigating, possibly using the method proposed by [Liu et al., 2013].

5.4 Future Work

This work has brought together a number of recommendations for future work. Firstly, it would be interesting to address the need for a simulation platform that covers all layers of the proposed architecture. This can be achieved by developing a Jadex-based API that integrates with the FI-WARE middleware and allows the agents to be distributed by different nodes in order to evaluate the efficiency of the communication protocols (e.g. agent-agent and agent-broker communication latency). Interested researchers could also focus on multi-resolution simulations as in MATSim [2], from the home-level to the neighbourhood, city and district levels and consequently evaluate how

[2]MATSim

the heterogeneity of regulation policies at these different levels impact various indicators, namely the sustainability indicators outlined by the United Nations; in this line of thought, it is worth considering both social and technical dimensions of Smart Cities so as to support the analysis of social coordination policies among households, following, for example, social simulation meta-models [Cunha et al., 2020]. Taken altogether, these ideas suggest a great potential for simulation-based projects.

Another interesting research opportunity lies in exploring Game Theory models for the coordination between various HEM Agents and energy providers. The present study has focused on the proposal-based m_2 market and considered the m_1 market to work over aggregated demand and fixed tariffs. In this sense, future work could target the m_1 market and explore new models that consider dynamic pricing strategies based on coordination and negotiation among households at different levels, as explained previously. Still in this line of thought, interactions between Appliance Scheduler agents and other agents belonging to exogenous systems raise several security concerns. Even if such interactions do not occur, but agents are to be distributed among different hardware nodes, attackers may capture context data or even take control over appliances. As a result, future studies could focus on exploring security vulnerabilities of multiagent systems in the scope of the proposed HAS architecture.

Moreover, the prospects raised by this work as of the behavioural changes resulting from the adoption of the HEMS suggest that a recommendation system for power management would be extremely valuable. As an example, one such recommendation system could suggest preference trade-offs leading to better use of the energy generated by a photovoltaic panel or lower energy expenses.

Finally, three other ideas arise in isolation from this work. Firstly, with the proliferation of machine learning and the consequent increase in the value of data, it follows that contextual data collected by Context Brokers presents itself as an artefact of great value. As a result, this observation calls for research on systems for data transactions between Smart Homes and interested parties. In particular, researchers could aim at designing a system that would allow Context Brokers to expose data-queries on context data in exchange for a monetary payoff that would be recorded into a Blockchain network; under this setting, security and trust considerations would have to be properly addressed, so one suggestion would be to follow the guidelines provided by the Framework for Identity and Data Sharing (FIDS) [Pentland et al., 2016], which has been delivered by the MIT to the Commission on Enhancing National Cybersecurity. Secondly, as this work suggests, research is also needed to provide a methodology for the development of multiagent systems over IoT. Together with new design models (that capture both the features of multiagent systems and those of IoT-based systems), as well as design patterns, these methodologies would accelerate the development of new projects and foster new developments within the MAS community. Lastly, interested researchers could also attempt to develop new standards for modelling Smart Homes based on Building Information Modelling (BIM), which is valuable to represent their IoT infrastructures, as argued by previous studies [Carneiro et al., 2018].

Appendix A

BDI Model - AgentSpeak(L)

Listing A.1: Belief Base Model

```
%%%%%%%%%%%%%%%%%% Appliance Scheduler %%%%%%%%%%%%%%%%%%

% Status: current status of the appliance (on/off)
status(Status).

% Room: number of the room where the agent-controlled appliance is located.
room(Room).

% Hour: current  hour, as perceived by the agent [0 ,  23]
hour(Hour).

% Profile: list where each value represents the scheduled power
% consumption for the corresponding period of the day, upon
% acceptance % by a HEM agent. |Profile| = 24
dailyPowerProfile(Profile).

% CriticalSatisfaction: critical satisfaction value provided
% by the consumer [0, 100]
criticalSatisfaction(CriticalSatisfaction).

% CurrentSatisfaction: current satisfaction value computed
% given the value of the corresponding characteristic
% variable [0,100]
currentSatisfaction(CurrentSatisfaction).

% Fact of arity 0 indicating that agent is satisfied
satisfied.

% Fact of arity 0 indicating that agent may turn on the appliance
mayTurnOn.

% Fact of arity 0 indicating that the appliance could not work when expected
```

```
couldNotWork.

% Proposal of the agent in response to a CFP issued by a HEM Agent
% CNPId: negotiation ID
% Proposals: the response of the agent to the request (a list of lists)
% StartHour: first hour of the period involved in the negotiation
% EndHour: last hour of the period involved in the negotiation
% Satisfactions: the satisfactions achieved with each profile
%   included in the Proposal (a list)
proposal(CNPId, Proposals, StartHour, EndHour, Satisfactions).

%%%%%%%%%%%%%%%%%% Leakage Detector %%%%%%%%%%%%%%%%%%

% Room: the number of the room in which a leakage has been detected
leaking(Room).

%%%%%%%%%%%%%%%%%% Gas Detector %%%%%%%%%%%%%%%%%%

% Room: the number of the room in which toxic gases have been detected
gasPresent(Room).

%%%%%%%%%%%%%%%%%% UninterruptibleLoadRegulator %%%%%%%%%%%%%%%%%%

%%% Rules

satisfaction(0, AET, EET, RET, LET) :-
    not(AET >= EET & AET <= LET).

satisfaction(Satisfaction, AET, EET, RET, LET) :-
    AET >= EET &
    AET <= RET &
    Satisfaction = (100 / (RET - EET)) * (AET - EET);

satisfaction(Satisfaction, AET, EET, RET, LET) :-
    AET > RET &
    AET <= LET &
    Satisfaction = (100 / (RET - LET)) * (AET - LET)

%%%

% Power: power consumed by the appliance
consumedPower(Power).

% Duration: duration of a full operation cycle (in hours)
cycleDuration(Duration).

% Hour: actual end time of the operational cycle, as computed by
% the agent during the negotiation phase
actualEndTime(Hour).
```

```
% Hour: earliest operational cycle end time, as provided by the consumer
earliestEndTime(Hour).

% Hour: latest operational cycle end time, as provided by the consumer
latestEndTime(Hour).

% Hour: required operational cycle end time, as provided by the consumer
requiredEndTime(Hour).

%%%%%%%%%%%%%%%%%% CurtailableLoadRegulator %%%%%%%%%%%%%%%%%%

%%% Rules

satisfaction(0, CharacteristicVariable) :-
    minCharacteristicVar(MinValue) &
    CharacteristicVariable < MinValue.

satisfaction(Satisfaction, CharacteristicVariable) :-
    minCharacteristicVar(MinValue) &
    reqCharacteristicVar(ReqValue) &
    CharacteristicVariable >= MinValue &
    CharacteristicVariable <= ReqValue &
    Satisfaction = 100 * (CharacteristicVariable - MinValue) / (ReqValue - MinValue
        ).

satisfaction(Satisfaction, CharacteristicVariable) :-
    maxCharacteristicVar(MaxValue) &
    reqCharacteristicVar(ReqValue) &
    CharacteristicVariable <= MaxValue &
    CharacteristicVariable > ReqValue &
    Satisfaction = 100 * (CharacteristicVariable - MaxValue) / (ReqValue - MaxValue
        ).

%%%

% Power: minimum power consumed by the appliance
minimumConsumedPower(Power).

% Power: maximum power consumed by the appliance
maximumConsumedPower(Power).

% Hour: time when the appliance should turn on, as provided by the consumer
startOperationTime(Hour).

% Hour: time when the appliance should turn off, as provided by the consumer
endOperationTime(Hour).

% MinValue: minimum value that the characteristic variable can reach,
```

```
% as provided by the consumer
minCharacteristicVar(MinValue).

% ReqValue: required value of the characteristic variable,
% as provided by the consumer
reqCharacteristicVar(ReqValue).

% MaxValue: maximum value that the characteristic variable can reach,
% as provided by the consumer
maxCharacteristicVar(MaxValue).

% CharacteristicVar: current value of the characteristic variable
actualCharacteristicVar(CharacteristicVar).

%%%%%%%%%%%%%%%%%% LightingSystemRegulator %%%%%%%%%%%%%%%%%%

%%% Rules

neededPowers([FirstPower, SecondPower], RequiredSatisfaction) :-
    neededPowers([FirstPower, SecondPower], RequiredSatisfaction) &
    not(FirstPower = SecondPower).

neededPowers([Power], RequiredSatisfaction) :-
    neededPowers([FirstPower, SecondPower], RequiredSatisfaction) &
    FirstPower = SecondPower &
    Power = FirstPower.

neededPowers(FirstPower, SecondPower, RequiredSatisfaction) :-
    reqCharacteristicVar(ReqValue) &
    minCharacteristicVar(MinValue) &
    maxCharacteristicVar(MaxValue) &
    satisfaction(RequiredSatisfaction, FirstCharacteristicVariableValue) &
    satisfaction(RequiredSatisfaction, SecondCharacteristicVariableValue) &
    FirstCharacteristicVariableValue >= MinValue &
    FirstCharacteristicVariableValue <= ReqValue &
    SecondCharacteristicVariableValue >= ReqValue &
    SecondCharacteristicVariableValue <= MaxValue &
    luminousEfficacy(Efficacy) &
    bulbs(NumberBulbs) &
    FirstPower = (FirstCharacteristicVariableValue / Efficacy * NumberBulbs) &
    SecondPower = (SecondCharacteristicVariableValue / Efficacy * NumberBulbs).

%%%

% Efficacy: luminous efficacy of the bulbs
luminousEfficacy(Efficacy).

% Number: number of bulbs that are part of the lighting system
bulbs(Number).
```

```
%%%%%%%%%%%%%%%%%% SoundSystemRegulator %%%%%%%%%%%%%%%%%%

%%% Rules

neededPowers([FirstPower, SecondPower], RequiredSatisfaction) :-
    neededPowers([FirstPower, SecondPower], RequiredSatisfaction) &
    not(FirstPower = SecondPower).

neededPowers([Power], RequiredSatisfaction) :-
    neededPowers([FirstPower, SecondPower], RequiredSatisfaction) &
    FirstPower = SecondPower &
    Power = FirstPower.

neededPowers(FirstPower, SecondPower, RequiredSatisfaction) :-
    reqCharacteristicVar(ReqValue) &
    minCharacteristicVar(MinValue) &
    maxCharacteristicVar(MaxValue) &
    satisfaction(RequiredSatisfaction, FirstCharacteristicVariableValue) &
    satisfaction(RequiredSatisfaction, SecondCharacteristicVariableValue) &
    FirstCharacteristicVariableValue >= MinValue &
    FirstCharacteristicVariableValue <= ReqValue &
    SecondCharacteristicVariableValue >= ReqValue &
    SecondCharacteristicVariableValue <= MaxValue &
    sensitivityRating(Rating) &
    .pow(2, (FirstCharacteristicVariableValue - Rating) / 3, FirstPower) &
    .pow(2, (SecondCharacteristicVariableValue - Rating) / 3, SecondPower).

%%%

% Rating: sensitivity rating of the sound system
sensitivityRating(Rating).

%%%%%%%%%%%%%%%%%% InterruptbileLoadRegulator %%%%%%%%%%%%%%%%%%

% Power: power consumed by the appliance
consumedPower(Power).

%%%%%%%%%%%%%%%%%% ElectricVehicleRegulator %%%%%%%%%%%%%%%%%%

%%% Rules

satisfaction(0, PredictedChargeValue) :-
    minCharge(MinValue) &
    PredictedChargeValue < MinValue.

satisfaction(100, PredictedChargeValue) :-
    reqCharge(ReqValue) &
    PredictedChargeValue > ReqValue.
```

```
satisfaction(Satisfaction, PredictedChargeValue) :-
    minCharge(MinValue) &
    reqCharge(ReqValue) &
    PredictedChargeValue >= MinValue &
    PredictedChargeValue <= ReqValue &
    Satisfaction = 100 * (PredictedChargeValue - MinValue) / (ReqValue - MinValue).

neededchargeValue(RequiredSatisfaction, ChargeValue) :-
    minCharge(MinValue) &
    reqCharge(ReqValue) &
    ChargeValue = ((RequiredSatisfaction * (ReqValue - MinValue))/100) + MinValue &
    ChargeValue >= MinValue &
    ChargeValue <= ReqValue.

%%%

% Rate: charging rate of the vehicle (measured in %/H)
chargingRate(Rate).

% Hour: time from which the vehicle can start charging, as provided
% by the consumer
startOperationTime(Hour).

% Hour: time from which the vehicle can no longer charge, as provided
% by the consumer
endOperationTime(Hour).

% MinValue: minimum required charge level that the vehicle shall attain
% by the end of the period allowed for recharging, as provided by
% the consumer
minCharge(MinValue).

% ReqValue: required charge level that the vehicle should attain
% by the end of the period allowed for recharging, as provided by
% the consumer
reqCharge(ReqValue).

% ChargeValue: charge value at the end of the vehicle's operation
% period given the negotiated power profile
actualPredictedCharge(ChargeValue).

% CNPId: the CNPId to which the PredictedChargeValue corresponds
% PredictedChargeValue: the predicted charge level by the end of the
%   operation period considering the consumption proposal provided as
%   a response to the request with the given CNPId
predictedChargeValue(CNPId, PredictedChargeValue).

%%%%%%%%%%%%%%%%%% HomeEnergyManager %%%%%%%%%%%%%%%%%%
```

```
% Hour: current  hour, as perceived by the agent [0 ,  23]
hour(Hour).

% N: number of scheduler agents participating in the negotiations
numberSchedulerAgents(N)

% Id: identifier of the last negotiation
negotiationId(Id)

% Percentage: percentage by which the requested satisfaction decreases at each
% iteration of the negotiation
satisfactionReduction(Percentage)

% Load: maximum power load
maxLoad(Load)

% Policy: regulation policy ('bill', 'green' or 'comfort')
regulationPolicy(Policy)

% Prices: a list containing the prices of energy (hour by hour)
energyPrices(Prices)

% Power: a list containing the power generated by the PV (hour by hour)
pvPower(Power)

% CriticalSatisfaction: critical satisfaction value provided
% by the consumer [0, 100]
criticalSatisfaction(CriticalSatisfaction).

% MinOutput: minimum output of the satisfaction function
% corresponding to the defined regulation policy
minPolicyOutput(MinOutput)

% MaxOutput: maximum output of the satisfaction function
% corresponding to the defined regulation policy
maxPolicyOutput(MaxOutput)

% Profile: list where each value represents the total
% scheduled power consumption for the corresponding period
% of the day. |Profile| = 24
dailyTotalPowerProfile(Profile)

% Proposal of a AS agent in response to a CFP issued by a HEM Agent
% Mechanism: the negotiation mechanism (either Anticipative or Emergency)
% CNPId: negotiation ID
% Proposals: the response of the agent to the request (a list of lists)
% StartHour: first hour of the period involved in the negotiation
% EndHour: last hour of the period involved in the negotiation
```

```
% Satisfactions: the satisfactions achieved with each profile
%   included in the Proposal (a list)
proposal(Mechanism, CNPId, Proposals, StartHour, EndHour, Satisfactions).
```

Listing A.2: Plan Base Model

```
% .power(on, Power)
    % This action allows the agent to switch the device on at a certain
    % power level (Power)

% .power(off)
    % This action allows the agent to switch off the device

% .sendEmergencyCall()
    % This action allows the agent to send an emergency
    % request message to the HEM Agent

% .updateList(StartHour, EndHour, Profile, NewPowerProfile, RenewedPowerProfile)
    % This action takes the StartHour and the EndHour corresponding
    % to a NewPowerProfile (a unidimensional list) and unifies
    % RenewedPowerProfile with a list resulting from substitution of hourly
    % power consumption values in the original Profile list by those of
    % the NewPowerProfile (for the given hours)

% .sendProposal(CNPId, PowerProfiles, Satisfactions)
    % This action allows the agent to send a proposal message to the HEM Agent,
    % where CNPId relates to the negotiation identifier. PowerProfiles is
    % a two-dimensional list containing one or more power profile proposals
    % and Satisfactions is a two-dimensional list whose values correspond to
    % satisfactions of the agent with each of the proposals

% .pow(X, Y, Result)
    % This action unifies Result with X to the power of Y

% .operationCycles(StartHour, EndHour, Satisfaction, OperationCycles,
%       Satisfactions)
    % This action unifies OperationCycles with a two-dimensional list
    % representating all possible power profiles (of given Satisfaction) that
    % the UninterruptibleLoadRegulator agent can achieve according to the [
        StartHour, % EndHour] period. Satisfactions is unified with a two-
        dimensional list
    % whose values correspond to satisfactions of the agent with each of the
    % power profiles

% .nullConsumptionProposal(StartHour, EndHour, NullProposal, MaxSatisfaction)
    % This action unifies MaxSatisfaction with 100 and NullProposal with
    % a unidimensional list of length EndHour-StartHour whereby all values
    % are set to zero
```

```
% .subList(Profile, StartHour, EndHour, SubProfile)
    % This action unifies SubProfile with the sublist of Profile in index
    % range [StartHour, EndHour]

% .lastIndexOf(Power, Profile, LastIndex)
    % This action unifies LastIndex with the index of the last occurrence of
    % Power in the Profile list

% .filledSchedule(Powers, RequiredSatisfaction, StartHour, EndHour,
%        OpStartHour, OpEndHour, Schedules, Satisfactions)
    % This action takes a two-dimensional list Powers of power values as well
    % as StartHour, EndHour, OpStartHour and OpEndHour. For each power
    % value it creates a power profile corresponding to the
    % [StartHour, EndHour] period in which the period [OpStartHour, OpEndHour]
    % is set to the corresponding power value, while all values outside
    % that period are set to 0. Each of the latter lists is included in a
    % two-dimensional list that is subsequently unified with Schedules.
    % Satisfactions is unified with a list of length equal to that of the
    % Powers list in which all values are set to RequiredSatisfaction

% .overlap(S1, E1, S2, E2)
    % This action determines whether the [S1, E1] and [S2, E2] periods
    % overlap, in which case it outputs 'true'

% .schedulePermutations(SchedulePermutations, Power, NeededChargeTime,
%      OperationStartHour, OperationEndHour, StartHour, EndHour,
%             RequiredSatisfaction, Satisfactions)
    % This action unifies SchedulePermutations and the corresponding Satisfactions.
    % SchedulePermutations is unified with all possible power profiles in which the
    % device charges for NeededChargeTime hours during the
    % [OperationStartHour, OperationEndHour] period, which is contained within
    % the [StartHour, EndHour] period. Power permutations are computed on
    % the basis of the predicted charge level if the device is set to charge
    % before StartHour. Satisfaction is unified with a two-dimensional list
    % in which all values are set to RequiredSatisfaction.

% .neededChargeTime(PredictedChargeValue, StartHour, NeededChargeTime)
    % This action unifies NeededChargeTime with the time (in hours) needed
    % for the device to achieve PredictedChargeValue considering the charge
    % level it will have at StartHour (since it may charge in the meantime)

% .allProposalsReceived(CNPId, N)
    % This action outputs 'true' whenever the HEM Agent has received N
    % replies to the negotiation with identifier CNPId

% .feasibleCombinationsForLoad(CNPId, ScheduleCombinations)
    % This action unifies ScheduleCombinations with a multi-dimensional
    % list representing all feasible combinations of the proposals received in
```

```
    % response to the negotiation with identifier CNPId. Unfeasible combinations
    % for which the sum of the power profiles of all the agents results in a
    % daily power profile for which at least one hourly power consumption
    % value exceeds the maximum aload load.

% .feasibleCombinationsForCriticalSatisfaction(ScheduleCombinations,
    FeasibleSchedules)
    % This action unifies FeasibleSchedules with a list of all
    % combinations taken from ScheduleCombinations for which the output of the
    % selected regulation function (Bill, Green or Comfort) is equal or greater to
    % the HEM Agent’s critical satisfaction level

% .bestFeasibleSchedule(FeasibleSchedules, Schedules)
    % This action unifies Schedules with a multi-dimensional list
    % corresponding to the best schedule among the FeasibleSchedules,
    % that is, that for which the output of the selected regulation criteria
    % is the highest. In case of the Emergency Mechanism the output of the
    % regulation function takes into account the whole day and not
    % the emergency period alone

% .renegotiate_consumer()
    % This action informs the consumer that no feasible shedule exists
    % and asks for preference changes

% .send_all_accept_messages(CNPId, Schedules)
    % This action sends an ’accept_proposal’ message to all Scheduler Agents,
    % thus informing them of the accepted power profile proposal (which is
    % included as part of the Schedules list)

% .update_daily_total_power_profile(Schedules)
    % This action updates the ’dailyTotalPowerProfile’ belief according to
    % the accepted power profile proposals included as part of the
    % Schedules list

%%%%%%%%%%%%%%%%%% Appliance Scheduler %%%%%%%%%%%%%%%%%%

+currentSatisfaction(CurrentSatisfaction)
    : criticalSatisfaction(CriticalSatisfaction) &
        CurrentSatisfaction >= CriticalSatisfaction
    <- +satisfied.

+currentSatisfaction(CurrentSatisfaction)
    : true
    <- -satisfied.

-satisfied
    : true
    <- .sendEmergencyCall().
```

```
+hour(Hour)
    : dailyPowerProfile(Profile) &
        .nth(Hour, Profile, Power) &
        Power > 0 &
        mayTurnOn
    <- +!power(on, Power).

+hour(Hour)
    : dailyPowerProfile(Profile) &
        .nth(Hour, Profile, Power) &
        Power > 0 &
        not(mayTurnOn)
    <- +couldNotWork.

+hour(Hour)
    : dailyPowerProfile(Profile) &
        .nth(Hour, Profile, Power) &
        Power = 0
    <- +!power(off).

+mayTurnOn
    : couldNotWork
    <- .sendEmergencyCall().

+!power(on, Power)
    : mayTurnOn
    <- .power(on, Power) ;
        +status(on).

+!power(off)
    <- .power(off) ;
        +status(off).

+status(off)
    : dailyPowerProfile(Profile) &
        .nth(Hour, Profile, Power) &
        Power > 0 &
        not(mayTurnOn)
    <- +couldNotWork.

% triggered by the HEM Agent
+accept_proposal(CNPId, ProposalIndex)
    : true
    <- ?proposal(CNPId, Proposals, StartHour, EndHour, Satisfactions) ;
        .nth(ProposalIndex, Proposals, NewPowerProfile) ;
        .nth(ProposalIndex, Satisfactions, Satisfaction) ;
        .dailyPowerProfile(Profile) ;
        .updateList(StartHour, EndHour, Profile, NewPowerProfile,
            RenewedPowerProfile) ;
```

```
        +dailyPowerProfile(RenewedPowerProfile) ;
        +currentSatisfaction(Satisfaction).


%%%%%%%%%%%%%%%%%% Leakage Detector %%%%%%%%%%%%%%%%%%

+leaking(Room)
    : room(MyRoom) &
        Room = MyRoom
    <- +!power(off) ;
        -mayTurnOn.

-leaking(Room)
    : room(MyRoom) &
        Room = MyRoom
    <- +mayTurnOn.

%%%%%%%%%%%%%%%%%% Gas Detector %%%%%%%%%%%%%%%%%%

+gasPresent(Room)
    : room(MyRoom) &
        Room = MyRoom
    <- power(off).

-gasPresent(Room)
    : room(MyRoom) &
        Room = MyRoom
    <- +mayTurnOn.

%%%%%%%%%%%%%%%%%% UninterruptibleLoadRegulator %%%%%%%%%%%%%%%%%%

+actualEndTime(Hour)
    : true
    <- !update_satisfaction.

+earliestEndTime(Hour)
    : true
    <- !update_satisfaction.

+requiredEndTime(Hour)
    : true
    <- !update_satisfaction.

+latestEndTime(Hour)
    : true
    <- !update_satisfaction.

+!update_satisfaction
    : true
```

```
    <- ?actualEndTime(AET) ;
        ?earliestEndTime (EET) ;
        ?requiredEndTime (RET) ;
        ?latestEndTime (LET) ;
        ?satisfaction(Satisfaction, AET, EET, RET, LET) ;
        +currentSatisfaction(Satisfaction).

+propose(anticipative, CNPId, StartHour, EndHour, RequiredSatisfaction)
    : criticalSatisfaction(CriticalSatisfaction) &
        RequiredSatisfaction < CriticalSatisfaction
    <-  +dailyPowerProfile([]) ;
        -couldNotWork ;
        +proposal(CNPId, [], StartHour, EndHour, []) ;
        .sendProposal(CNPId, [], []).

+propose(anticipative, CNPId, StartHour, EndHour, RequiredSatisfaction)
    : criticalSatisfaction(CriticalSatisfaction) &
        RequiredSatisfaction >= CriticalSatisfaction &
        .operationCycles(StartHour, EndHour, RequiredSatisfaction, [], [])
    <- +dailyPowerProfile([]) ;
        -couldNotWork ;
        +proposal(CNPId, [], StartHour, EndHour, []) ;
        .sendProposal(CNPId, [], []).

+propose(anticipative, CNPId, StartHour, EndHour, RequiredSatisfaction)
    : criticalSatisfaction(CriticalSatisfaction) &
        RequiredSatisfaction >= CriticalSatisfaction  &
        (not (.operationCycles(StartHour, EndHour, RequiredSatisfaction, [], [])))
    <- .operationCycles(StartHour, EndHour, RequiredSatisfaction,
            OperationCycles, Satisfactions) ;
        +dailyPowerProfile([]) ;
        -couldNotWork ;
        +proposal(CNPId, OperationCycles, StartHour, EndHour, Satisfactions) ;
        .sendProposal(CNPId, OperationCycles, Satisfactions).

+propose(emergency, CNPId, StartHour, EndHour, RequiredSatisfaction)
    : not(couldNotWork) &
        actualEndTime(AET) &
        AET <= StartHour
    <- .nullConsumptionProposal(StartHour, EndHour, NullProposal,
            MaxSatisfaction) ;
        +proposal(CNPId, NullProposal, StartHour, EndHour, MaxSatisfaction) ;
        .sendProposal(CNPId, NullProposal, MaxSatisfaction).

+propose(emergency, CNPId, StartHour, EndHour, RequiredSatisfaction)
    : not(couldNotWork) &
        hour(Hour) &
        actualEndTime(AET) &
        AET > StartHour &
```

```
        cycleDuration(Duration) &
        Hour >= (AET - Duration)
    <- ?dailyPowerProfile(Profile) ;
        .subList(Profile, StartHour, EndHour, SubProfile) ;
        ?currentSatisfaction(Satisfaction) ;
        +proposal(CNPId, SubProfile, StartHour, EndHour, Satisfaction) ;
        .sendProposal(CNPId, SubProfile, Satisfaction).

+propose(emergency, CNPId, StartHour, EndHour, RequiredSatisfaction)
    : criticalSatisfaction(CriticalSatisfaction) &
        RequiredSatisfaction < CriticalSatisfaction
    <- +proposal(CNPId, [], StartHour, EndHour, []) ;
        .sendProposal(CNPId, [], []).

+propose(emergency, CNPId, StartHour, EndHour, RequiredSatisfaction)
    : criticalSatisfaction(CriticalSatisfaction) &
        RequiredSatisfaction >= CriticalSatisfaction &
        .operationCycles(StartHour, EndHour, RequiredSatisfaction, [], [])
    <- +proposal(CNPId, [], StartHour, EndHour, []) ;
        .sendProposal(CNPId, [], []).

+propose(emergency, CNPId, StartHour, EndHour, RequiredSatisfaction)
    : criticalSatisfaction(CriticalSatisfaction) &
        RequiredSatisfaction >= CriticalSatisfaction  &
        (not (.operationCycles(StartHour, EndHour, RequiredSatisfaction, [], [])))
    <- .operationCycles(StartHour, EndHour, RequiredSatisfaction,
            OperationCycles, Satisfactions) ;
        +proposal(CNPId, OperationCycles, StartHour, EndHour, Satisfactions) ;
        .sendProposal(CNPId, OperationCycles, Satisfactions).

+accept_proposal(CNPId, ProposalIndex)
    : true
    <- ?proposal(CNPId, Proposals, StartHour, EndHour, Satisfactions) ;
        .nth(ProposalIndex, Proposals, NewPowerProfile) ;
        .nth(ProposalIndex, Satisfactions, Satisfaction) ;
        ?dailyPowerProfile(Profile) ;
        ?updateList(StartHour, EndHour, Profile, NewPowerProfile,
            RenewedPowerProfile) ;
        ?consumedPower(Power) ;
        .lastIndexOf(Power, Profile, LastIndex) ;
        +dailyPowerProfile(RenewedPowerProfile) ;
        +actualEndTime(LastIndex + StartHour + 1).

%%%%%%%%%%%%%%%%%% CurtailableLoadRegulator %%%%%%%%%%%%%%%%%%

+actualCharacteristicVar(CharacteristicVariable)
    : true
    <- !update_satisfaction.
```

```
+!update_satisfaction
    : true
    <- ?actualCharacteristicVar(CharacteristicVariable) ;
        ?satisfaction(Satisfaction, CharacteristicVariable) ;
        +currentSatisfaction(Satisfaction).

+propose(anticipative, CNPId, StartHour, EndHour, RequiredSatisfaction)
    : criticalSatisfaction(CriticalSatisfaction) &
        RequiredSatisfaction < CriticalSatisfaction
    <-  +dailyPowerProfile([]) ;
        -couldNotWork ;
        +proposal(CNPId, [], StartHour, EndHour, []) ;
        .sendProposal(CNPId, [], []).

+propose(anticipative, CNPId, StartHour, EndHour, RequiredSatisfaction)
    : criticalSatisfaction(CriticalSatisfaction) &
        RequiredSatisfaction >= CriticalSatisfaction
    <- +dailyPowerProfile([]) ;
        -couldNotWork ;
        ?neededPowers(Powers, RequiredSatisfaction) ;
        ?startOperationTime(SH) ;
        ?endOperationTime(EH) ;
        .filledSchedule(Powers, RequiredSatisfaction, StartHour, EndHour, SH,
            EH, Schedules, Satisfactions) ;
        +proposal(CNPId, Schedules, Satisfactions) ;
        .sendProposal(CNPId, Schedules, Satisfactions).

+propose(emergency, CNPId, StartHour, EndHour, RequiredSatisfaction)
    : startOperationTime(StartOperation) &
        endOperationTime(EndOperation) &
        not(.overlap(StartHour, EndHour, StartOperation, EndOperation))
    <- .nullConsumptionProposal(StartHour, EndHour, NullProposal,
            MaxSatisfaction) ;
        +proposal(CNPId, NullProposal, StartHour, EndHour, MaxSatisfaction) ;
        .sendProposal(CNPId, NullProposal, MaxSatisfaction).

+propose(emergency, CNPId, StartHour, EndHour, RequiredSatisfaction)
    : startOperationTime(StartOperation) &
        endOperationTime(EndOperation) &
        .overlap(StartHour, EndHour, StartOperation, EndOperation) &
        criticalSatisfaction(CriticalSatisfaction) &
        RequiredSatisfaction < CriticalSatisfaction
    <- +proposal(CNPId, [], StartHour, EndHour, []) ;
        .sendProposal(CNPId, [], []).

+propose(emergency, CNPId, StartHour, EndHour, RequiredSatisfaction)
    : startOperationTime(StartOperation) &
        endOperationTime(EndOperation) &
        .overlap(StartHour, EndHour, StartOperation, EndOperation) &
```

```
        criticalSatisfaction(CriticalSatisfaction) &
        RequiredSatisfaction >= CriticalSatisfaction
    <- ?neededPowers(Powers, RequiredSatisfaction) ;
        ?startOperationTime(SH) ;
        ?endOperationTime(EH) ;
        .filledSchedule(Powers, RequiredSatisfaction, StartHour, EndHour,
            SH, EH, Schedules, Satisfactions) ;
        +proposal(CNPId, Schedules, Satisfactions) ;
        .sendProposal(CNPId, Schedules, Satisfactions).

%%%%%%%%%%%%%%%%%% ElectricVehicleRegulator %%%%%%%%%%%%%%%%%%

+actualPredictedCharge(PredictedChargeValue)
    : true
    <- !update_satisfaction.

+!update_satisfaction
    : true
    <- ?actualPredictedCharge(PredictedChargeValue) ;
        ?satisfaction(Satisfaction, PredictedChargeValue) ;
        +currentSatisfaction(Satisfaction).

+accept_proposal(CNPId, ProposalIndex)
    : true
    <- ?predictedChargeValue(CNPId, PredictedChargeValue) ;
        +actualPredictedCharge(PredictedChargeValue).

+propose(anticipative, CNPId, StartHour, EndHour, RequiredSatisfaction)
    : criticalSatisfaction(CriticalSatisfaction) &
        RequiredSatisfaction < CriticalSatisfaction
    <-  +dailyPowerProfile([]) ;
        -couldNotWork ;
        +proposal(CNPId, [], StartHour, EndHour, []) ;
        .sendProposal(CNPId, [], []).

+propose(anticipative, CNPId, StartHour, EndHour, RequiredSatisfaction)
    : criticalSatisfaction(CriticalSatisfaction) &
        RequiredSatisfaction >= CriticalSatisfaction
    <- ?neededchargeValue(RequiredSatisfaction, PredictedChargeValue) ;
        .neededChargeTime(PredictedChargeValue, StartHour, NeededChargeTime) ;
        ?startOperationTime(OperationStartHour) ;
        ?endOperationTime(OperationEndHour) ;
        ?consumedPower(Power);
        .schedulePermutations(SchedulePermutations, Power, NeededChargeTime,
            OperationStartHour, OperationEndHour, StartHour, EndHour,
            RequiredSatisfaction, Satisfactions) ;
        +predictedChargeValue(CNPId, PredictedChargeValue) ;
        +dailyPowerProfile([]) ;
        -couldNotWork ;
```

```
        +proposal(CNPId, SchedulePermutations, StartHour, EndHour, Satisfactions) ;
        .sendProposal(CNPId, SchedulePermutations, Satisfactions).

+propose(emergency, CNPId, StartHour, EndHour, RequiredSatisfaction)
    : ?dailyPowerProfile(Profile) &
        ?consumedPower(Power) &
        .lastIndexOf(Power, Profile, LastIndex) &
        StartHour >= (LastIndex + 1)
    <- .nullConsumptionProposal(StartHour, EndHour, NullProposal,
            MaxSatisfaction) ;
        +proposal(CNPId, NullProposal, StartHour, EndHour, MaxSatisfaction) ;
        .sendProposal(CNPId, NullProposal, MaxSatisfaction).

+propose(emergency, CNPId, StartHour, EndHour, RequiredSatisfaction)
    : ?dailyPowerProfile(Profile) &
        ?consumedPower(Power) &
        .lastIndexOf(Power, Profile, LastIndex) &
        (LastIndex + 1) > StartHour  &
        criticalSatisfaction(CriticalSatisfaction) &
        RequiredSatisfaction < CriticalSatisfaction
    <- +proposal(CNPId, [], StartHour, EndHour, []) ;
        .sendProposal(CNPId, [], []).

+propose(emergency, CNPId, StartHour, EndHour, RequiredSatisfaction)
    : ?dailyPowerProfile(Profile) &
        ?consumedPower(Power) &
        .lastIndexOf(Power, Profile, LastIndex) &
        (LastIndex + 1) > StartHour  &
        criticalSatisfaction(CriticalSatisfaction) &
        RequiredSatisfaction >= CriticalSatisfaction
    <- +propose(anticipative, CNPId, StartHour, EndHour, RequiredSatisfaction).


%%%%%%%%%%%%%%%%%% HomeEnergyManager %%%%%%%%%%%%%%%%%%

+negotiate(CNPId, anticipative, RequiredSatisfaction, StartHour)
    : true
    <- -proposal(_, CNPId, _, _, _, _) % remove all previous proposals to that
        negotiation
        +dailyPowerProfile([]) ;
        !call(CNPId, anticipative, RequiredSatisfaction, StartHour) ;
        !retrieve_proposals(CNPId) ;
        !find_scheduling(CNPId, Schedules)
        !announce_results(CNPId, anticipative, RequiredSatisfaction, Schedules).

+negotiate(CNPId, emergency, RequiredSatisfaction, StartHour)
    : true
    <- -proposal(_, CNPId, _, _, _, _) % remove all previous proposals to that
        negotiation
```

```
        !call(CNPId, emergency, RequiredSatisfaction, StartHour) ;
        !retrieve_proposals(CNPId) ;
        !find_scheduling(CNPId, Schedules)
        !announce_results(CNPId, emergency, RequiredSatisfaction, Schedules).

+!call(CNPId, Mechanism, RequiredSatisfaction, StartHour)
    : true
    <- .df_search("participant",Participants);
        +numberSchedulerAgents(.length(Participants)) ;
        .send(Participants, tell, propose(Mechanism, CNPId, StartHour, 24,
            RequiredSatisfaction)).

!retrieve_proposals(CNPId)
    : true
    <- ?numberSchedulerAgents(N) ;
        .wait(.allProposalsReceived(CNPId, N)).

!find_scheduling(CNPId, [], []) % unfeasible problem
    : proposal(_, CNPId, [], _, _, []) % at least one agent can't find a shedule
        according to constraints
    <- true.

!find_scheduling(CNPId, Schedules)
    : not(proposal(_, CNPId, [], _, _, [])) % all agents can find a feasible
        schedule
    <- .feasibleCombinationsForLoad(CNPId, ScheduleCombinations) ;
        .feasibleCombinationsForCriticalSatisfaction(ScheduleCombinations,
            FeasibleSchedules) ;
        .bestFeasibleSchedule(FeasibleSchedules, Schedules).

% empty schedule list indicates no feasible schedule was found
!announce_results(CNPId, Mechanism, RequiredSatisfaction, [])
    : satisfactionReduction(Percentage) &
        (RequiredSatisfaction - Percentage = 0)
    <- .renegotiate_consumer().

% empty schedule list indicates no feasible schedule was found
!announce_results(CNPId, Mechanism, RequiredSatisfaction, [])
    : satisfactionReduction(Percentage) &
        (RequiredSatisfaction - Percentage > 0)
    <- ?satisfactionReduction(Percentage) ;
        +!negotiate(CNPId, Mechanism, RequiredSatisfaction - Percentage, StartHour)
            .

% empty schedule list indicates no feasible schedule was found
!announce_results(CNPId, Mechanism, RequiredSatisfaction, Schedules)
    : satisfactionReduction(Percentage) &
        (RequiredSatisfaction - Percentage > 0) &
        not(.empty(Schedules))
```

```
    <- .send_all_accept_messages(CNPId, Schedules)
        .update_daily_total_power_profile(Schedules).

% some agent requested an Emergency negotiation
+emergency()
    : true
    <- ?hour(CurrentHour) ;
        ?negotiationId(CNPId) ;
        +negotiate(CNPId + 1, emergency, 100, CurrentHour+1).
```

Appendix B

Simulation Conditions

Table B.1: Features of the appliances considered in the simulations.

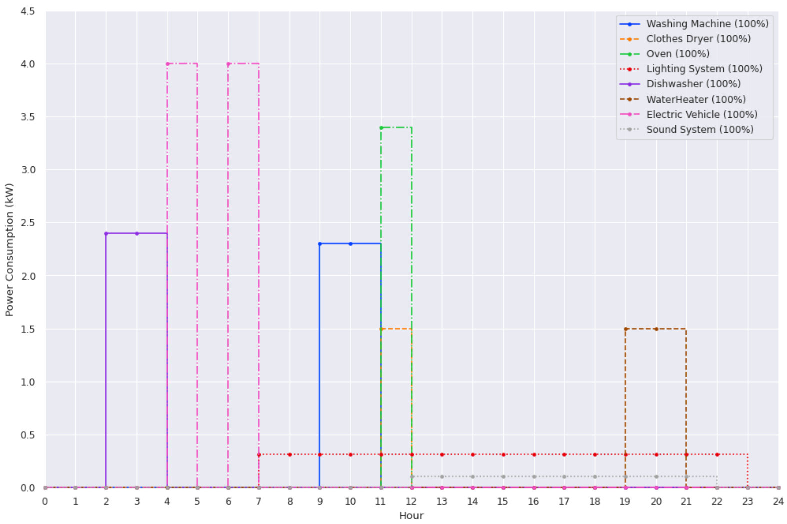
	Oven	Washing Machine	Clothes Dryer	Dishwasher	Electric Water Heater	Sound System	Lighting Set	Electric Vehicle
Appliance Category	Uninterruptible	Uninterruptible	Uninterruptible	Uninterruptible	Uninterruptible	Curtailable	Curtailable	Interruptible
Consumed Power (kW)	3,40	2,30	1,50	2,40	1,50	-	-	4
Cycle Duration (h)	1	2	1	2	2	-	-	-
Minimum Power Consumption (kW)	-	-	-	-	-	0,03	0,45	-
Maximum Power Consumption (kW)	-	-	-	-	-	0,33	0,75	-
Charging Rate (%/h)	-	-	-	-	-	-	-	20
Sensitivity Rating (dB)	-	-	-	-	-	85	-	-
Luminous Efficacy (lux)	-	-	-	-	-	-	20	-
Number of Bulbs	-	-	-	-	-	-	10	-

Table B.2: Consumer preferences regarding the operation of appliances. These were set as parameters of the simulations described in Chapter 4.

	Oven	**Washing Machine**	**Dishwasher**	**Clothes Dryer**	**Electric Water Heater**	**Sound System**	**Lighting Set**	**Electric Vehicle**
Critical Satisfaction	50	50	50	50	50	50	50	50
EET	10	8	2	11	19	-	-	-
RET	12	11	4	12	21	-	-	-
LET	14	13	7	14	22	-	-	-
Operation Period Start	-	-	-	-	-	12	7	-
Operation Period End	-	-	-	-	-	22	23	-
Min. Characteristic Var.	-	-	-	-	-	100	500	-
Best Characteristic Var.	-	-	-	-	-	105	625	-
Max. Characteristic Var.	-	-	-	-	-	110	750	-
Start of Charging Period	-	-	-	-	-	-	-	1
End of Charging Period	-	-	-	-	-	-	-	8
Min. Charge	-	-	-	-	-	-	-	80
Required Charge	-	-	-	-	-	-	-	90

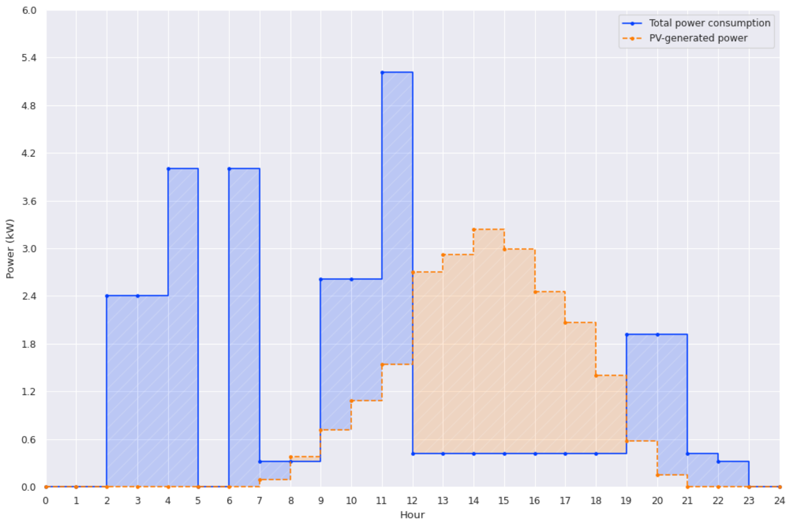
Appendix C

Simulation Results

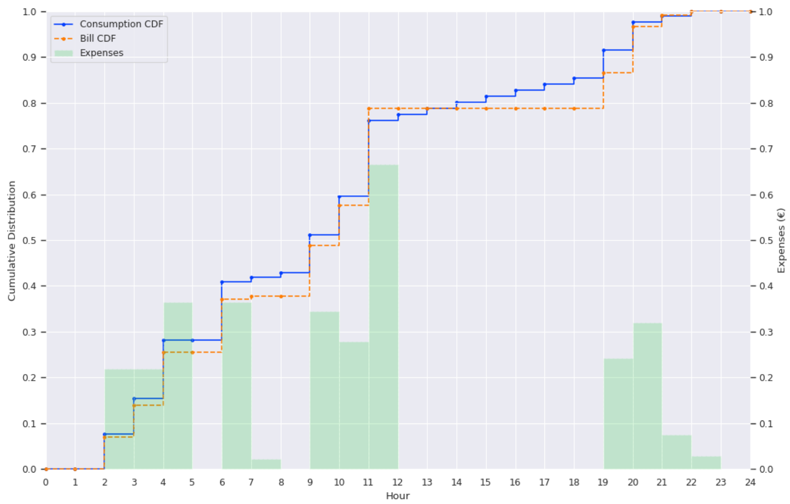


(a) Appliances' scheduling.

Figure C.1: Comfort policy: results of the simulation.

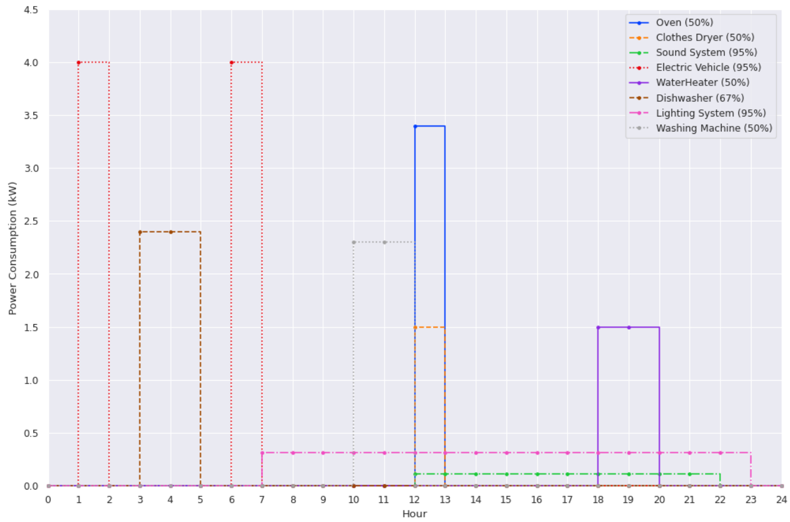


(b) Total power consumption against PV-generated power.

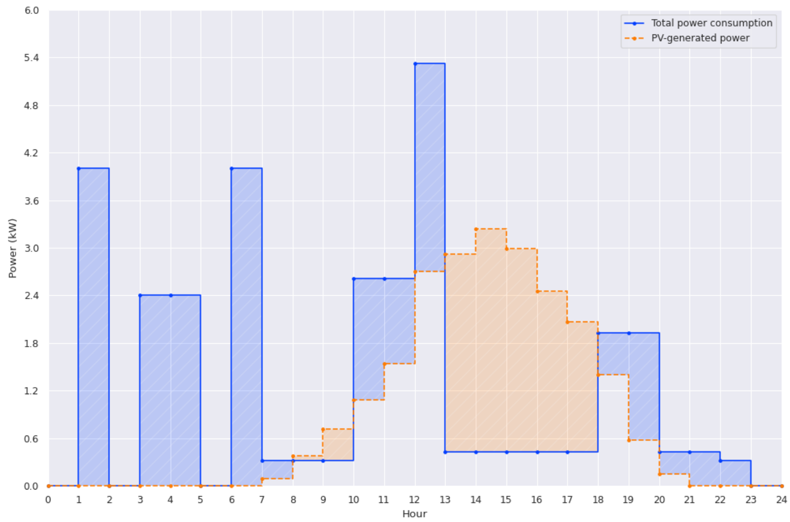


(c) Cumulative distribution functions and hourly spendings.

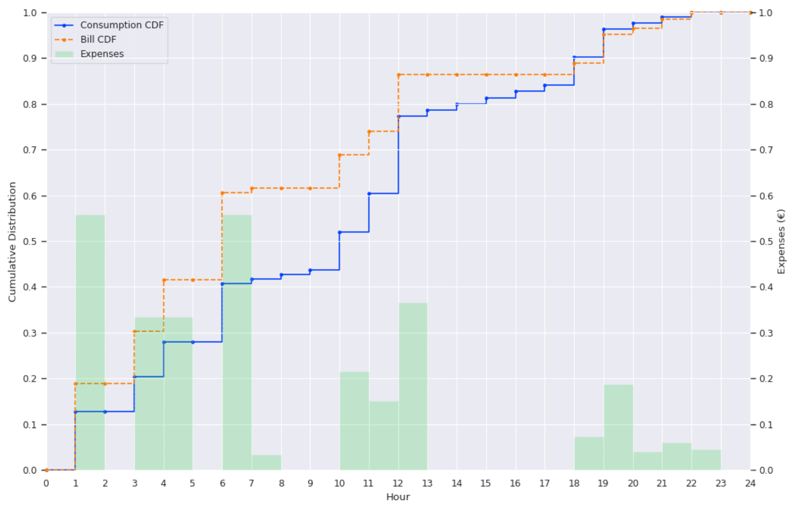
Figure C.1: Comfort policy: results of the simulation *(cont.)*.



(a) Appliances' scheduling.

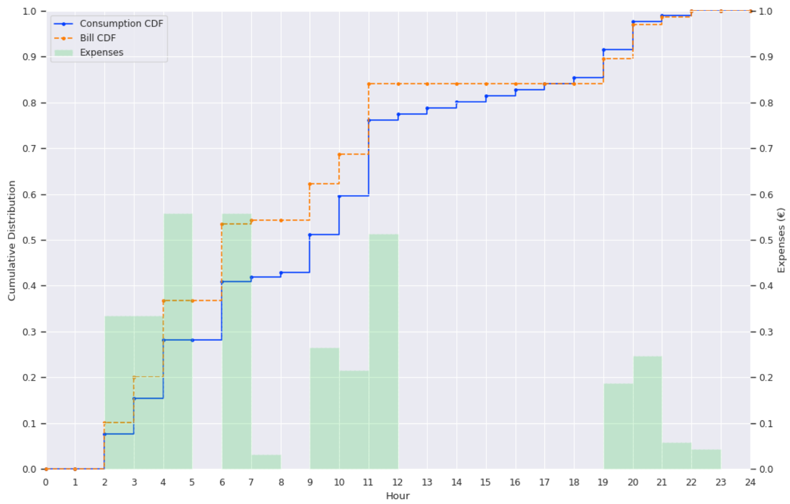


(b) Total power consumption against PV-generated power.



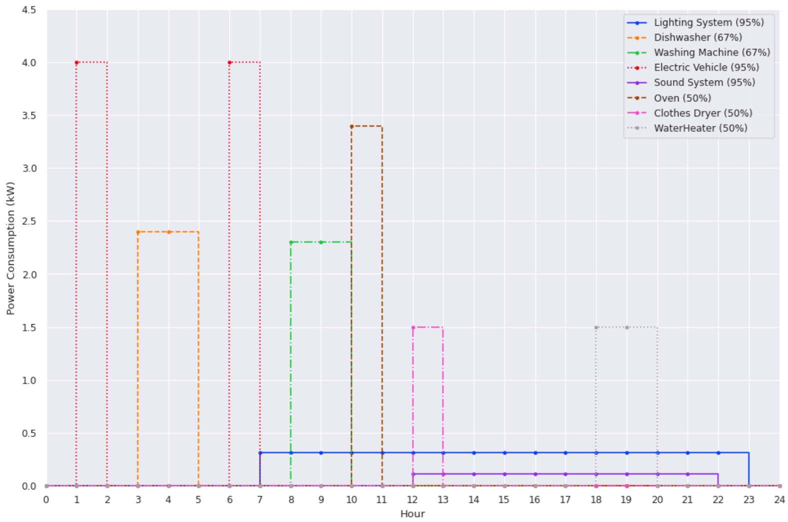
(c) Cumulative distribution functions and hourly spendings.

Figure C.2: Bill regulation with 50% critical satisfaction and Simple tariff: results of the simulation.

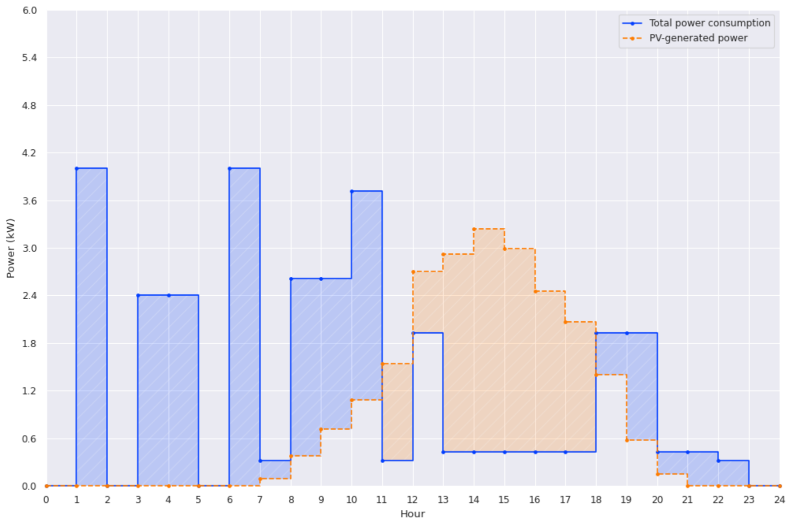


(a) Cumulative distribution functions and hourly spendings.

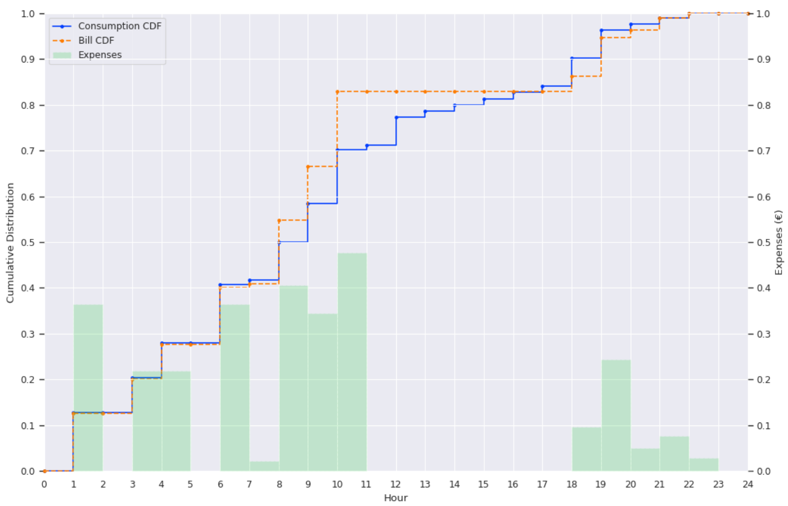
Figure C.3: Bill regulation with 20% critical satisfaction and Simple tariff: results of the simulation.



(a) Appliances' scheduling.

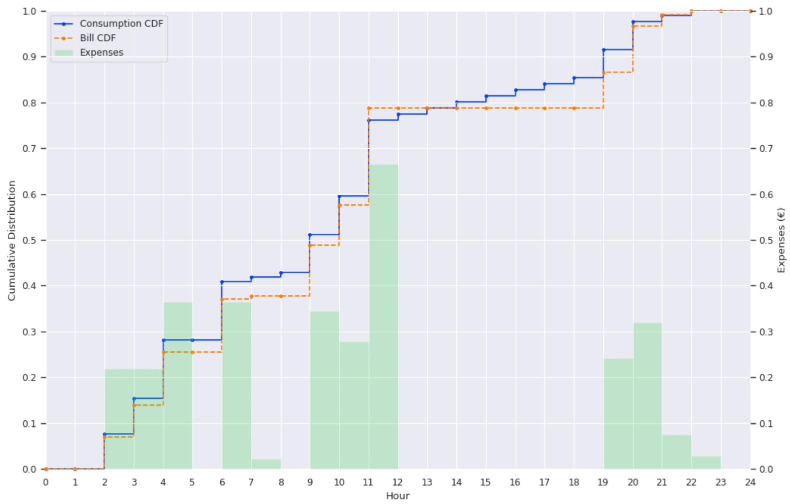


(b) Total power consumption against PV-generated power.



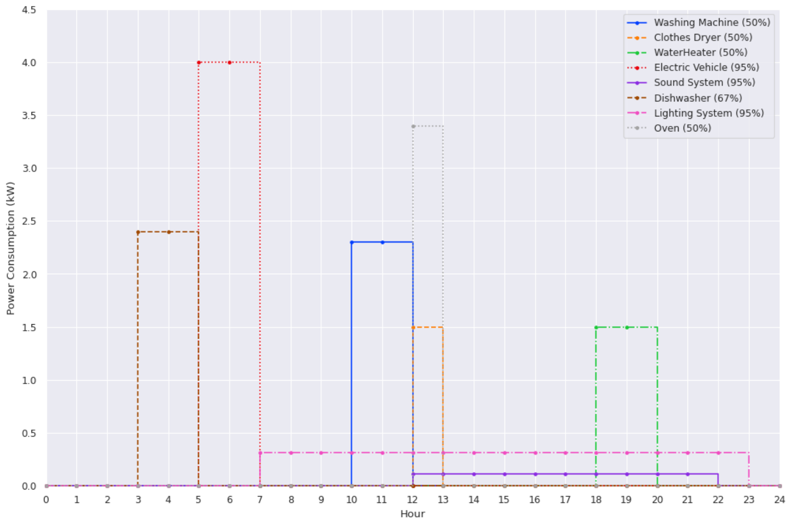
(c) Cumulative distribution functions and hourly spendings.

Figure C.4: Bill regulation with 50% critical satisfaction and Dual-Rate tariff: results of the simulation.

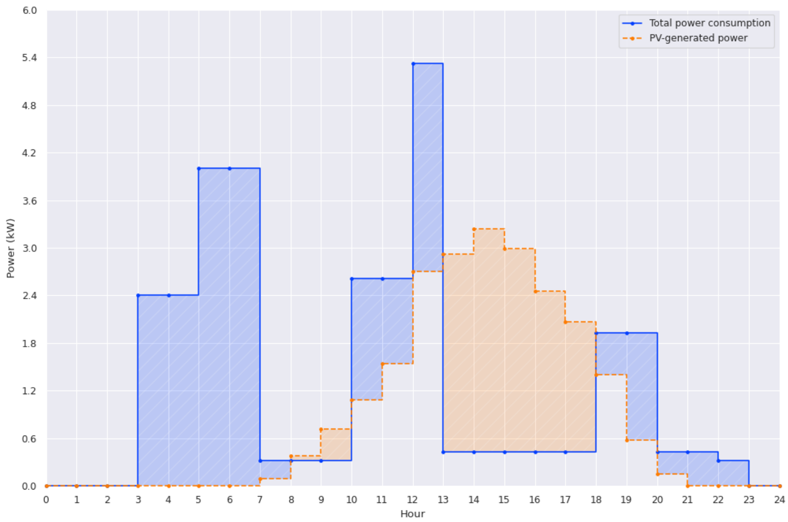


(a) Cumulative distribution functions and hourly spendings.

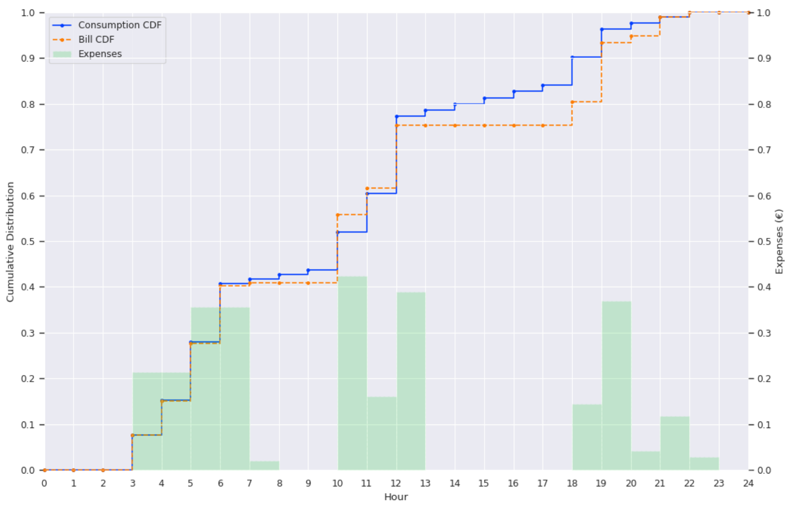
Figure C.5: Bill regulation with 20% critical satisfaction and Dual-Rate tariff: results of the simulation.



(a) Appliances' scheduling.

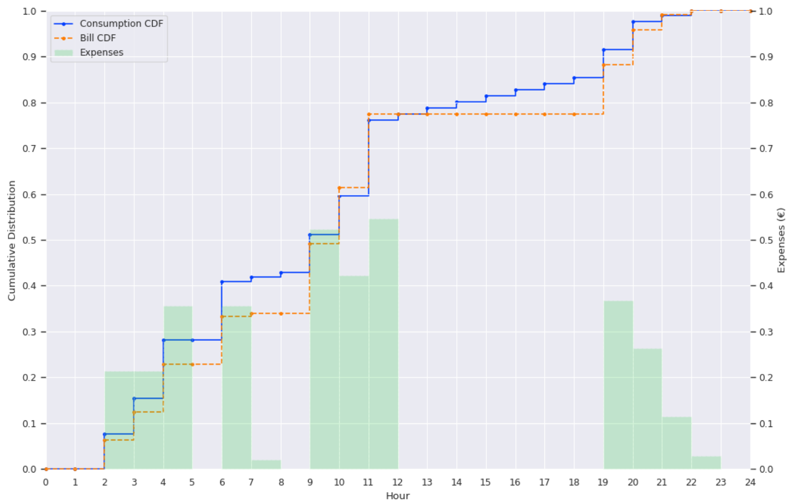


(b) Total power consumption against PV-generated power.



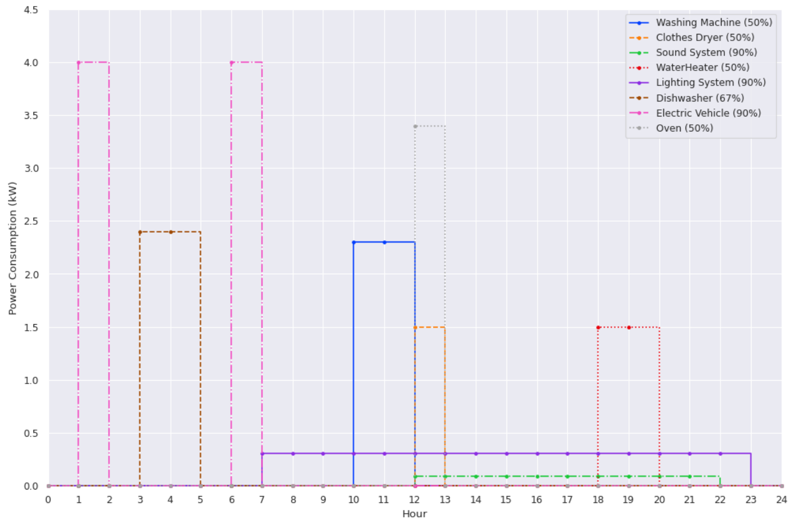
(c) Cumulative distribution functions and hourly spendings.

Figure C.6: Bill regulation with 50% critical satisfaction and Triple-Rate tariff: results of the simulation.

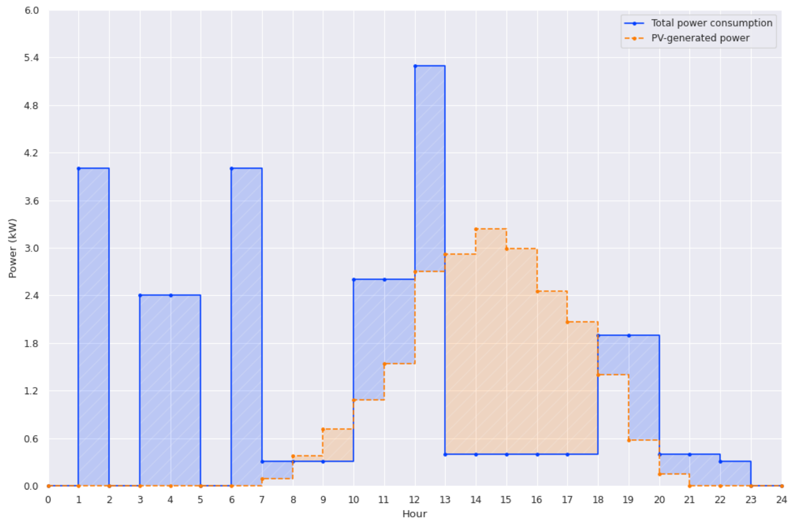


(a) Cumulative distribution functions and hourly spendings.

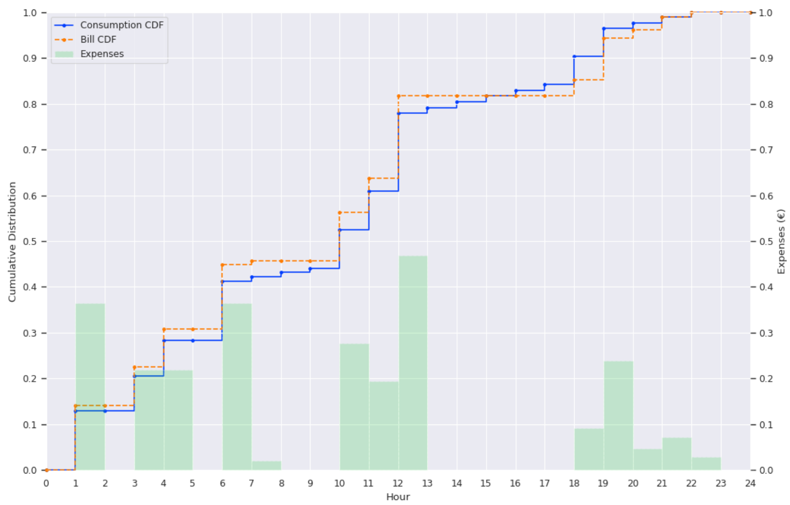
Figure C.7: Bill regulation with 20% critical satisfaction and Triple-Rate tariff: results of the simulation.



(a) Appliances' scheduling.

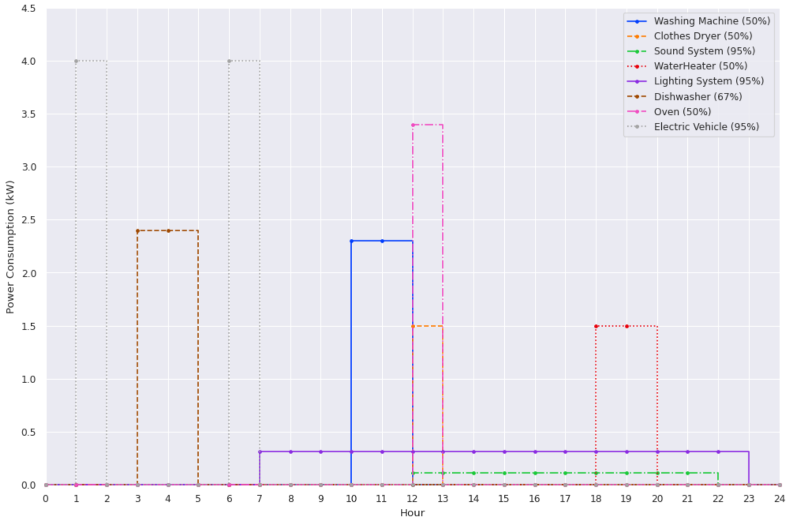


(b) Total power consumption against PV-generated power.

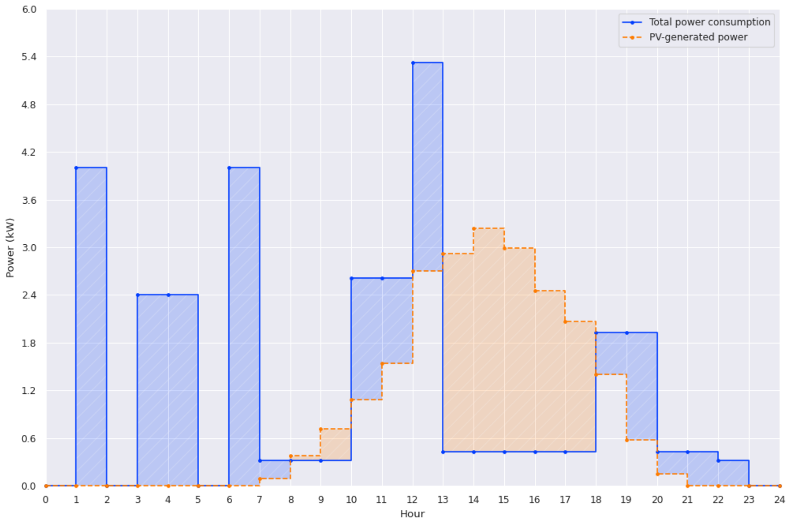


(c) Cumulative distribution functions and hourly spendings.

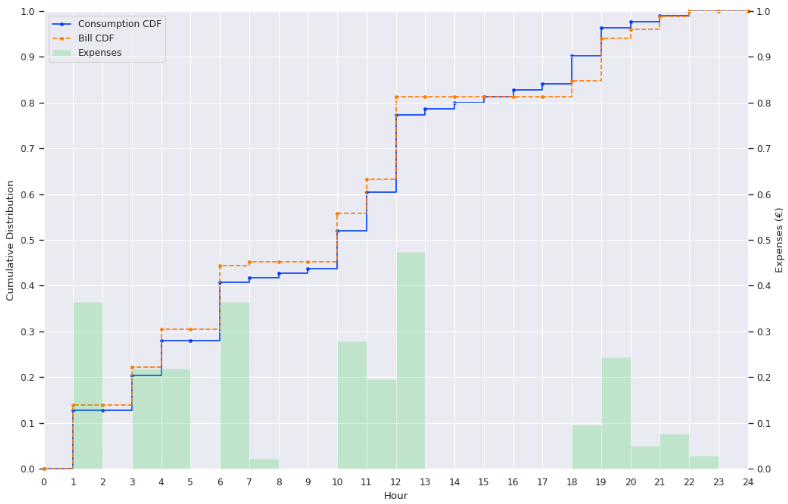
Figure C.8: Green regulation with 30% critical satisfaction and Dual-Rate tariff: results of the simulation.



(a) Appliances' scheduling.

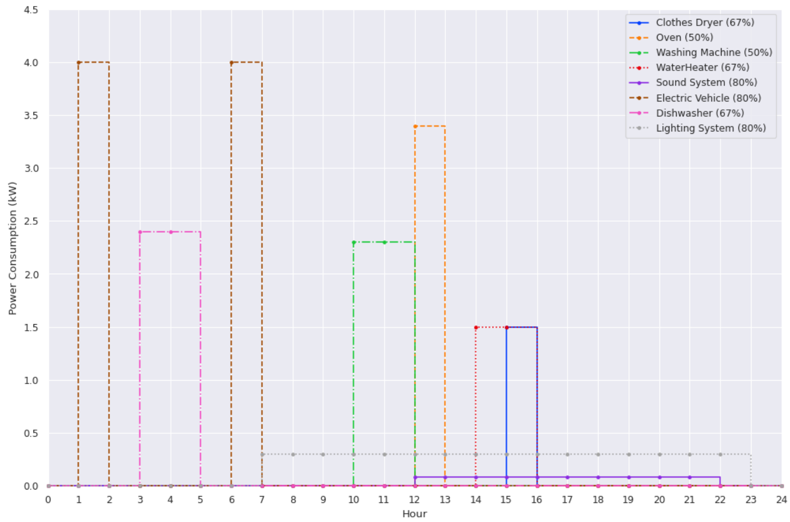


(b) Total power consumption against PV-generated power.

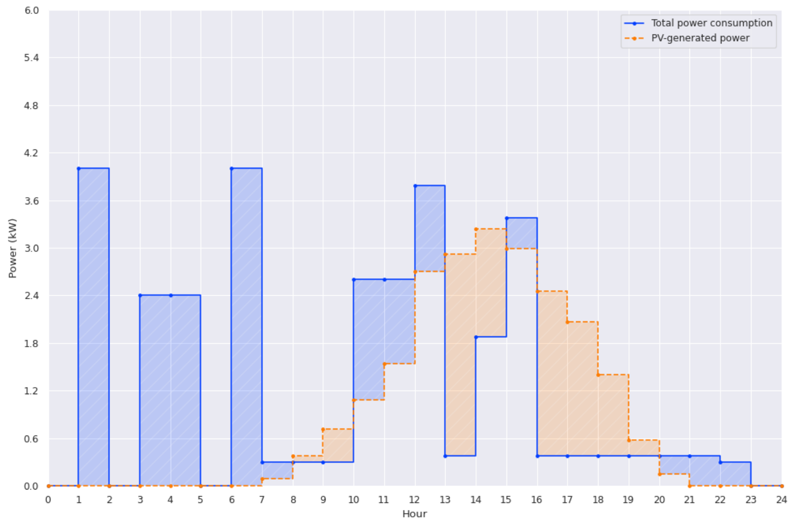


(c) Cumulative distribution functions and hourly spendings.

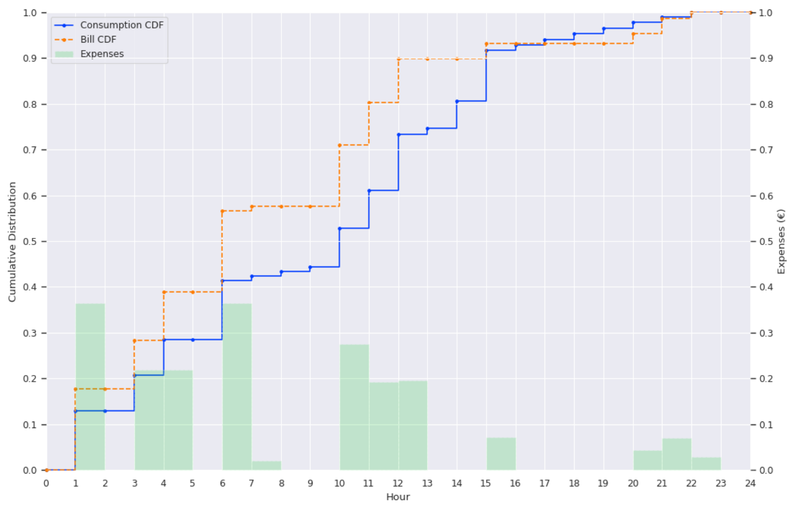
Figure C.9: Green regulation with 25% critical satisfaction and Dual-Rate tariff: results of the simulation.



(a) Appliances' scheduling.

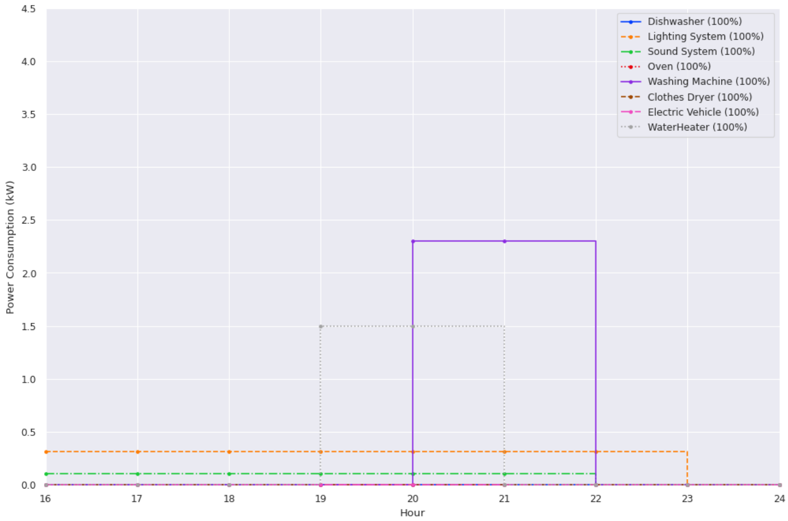


(b) Total power consumption against PV-generated power.

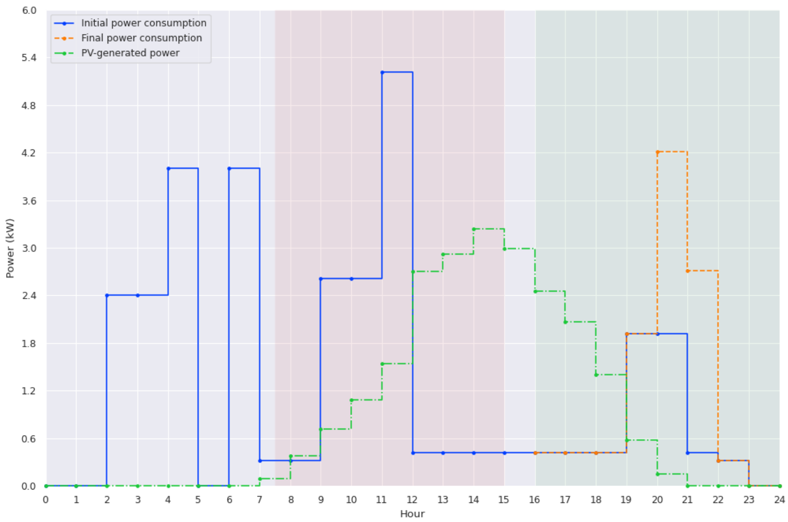


(c) Cumulative distribution functions and hourly spendings.

Figure C.10: Green regulation with changed preferences and Dual-Rate tariff: results of the simulation.

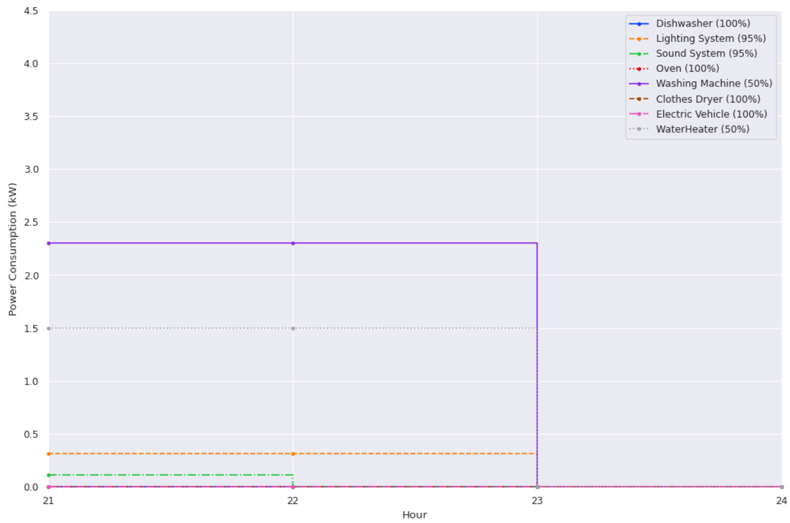


(a) Appliances' scheduling.

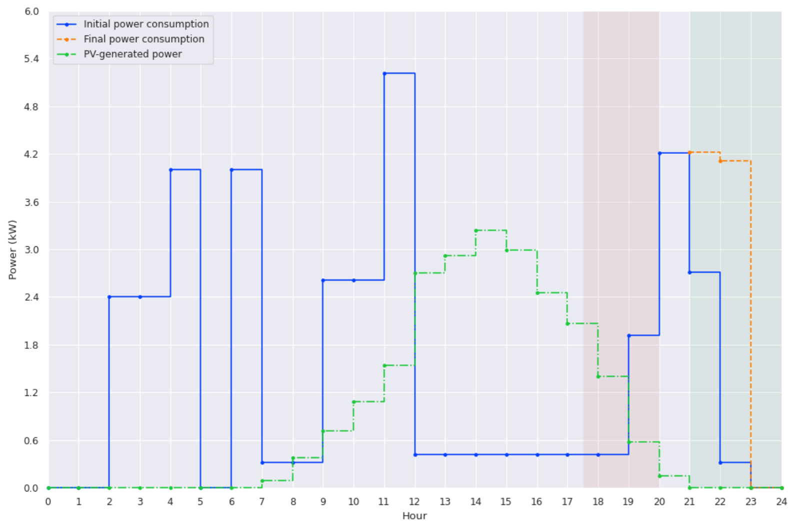


(b) Total power consumption against PV-generated power.

Figure C.11: Bill regulation with Dual-Rate tariff: results of the simulation following the first unexpected event.

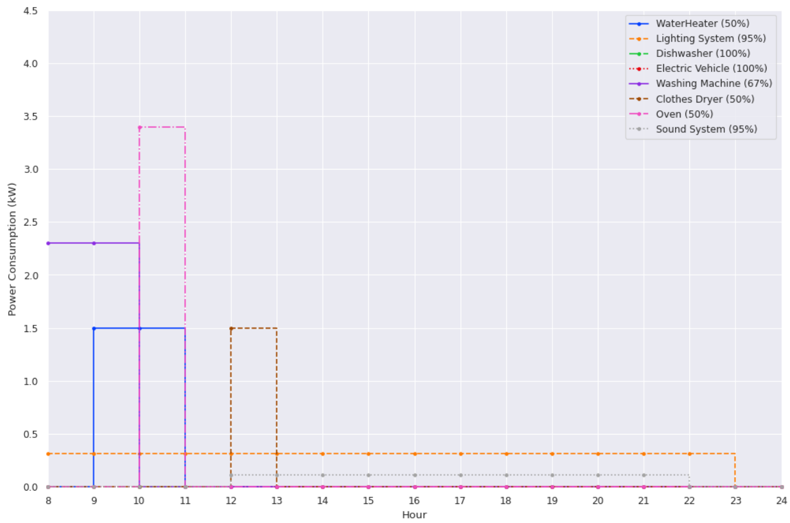


(a) Appliances' scheduling.

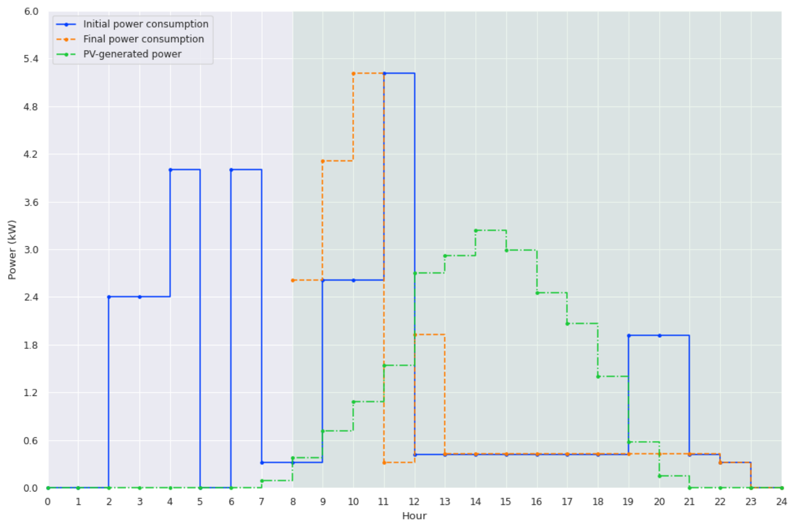


(b) Total power consumption against PV-generated power.

Figure C.12: Bill regulation with Dual-Rate tariff: results of the simulation following the second unexpected event.

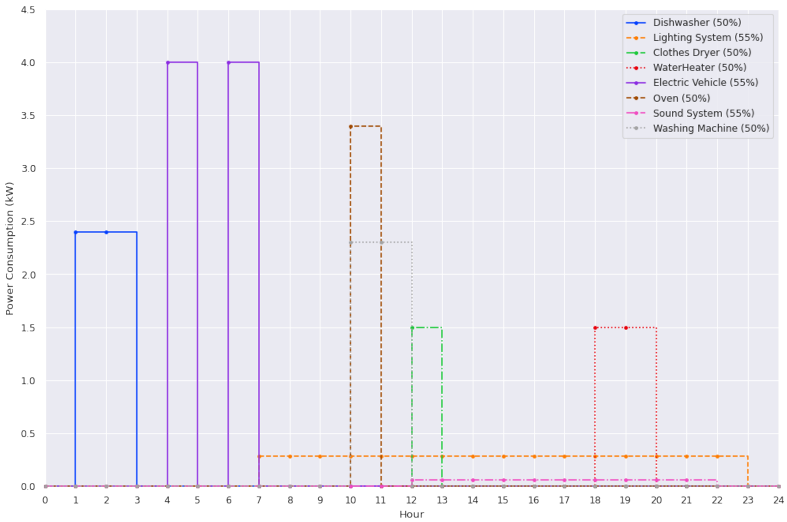


(a) Appliances' scheduling.

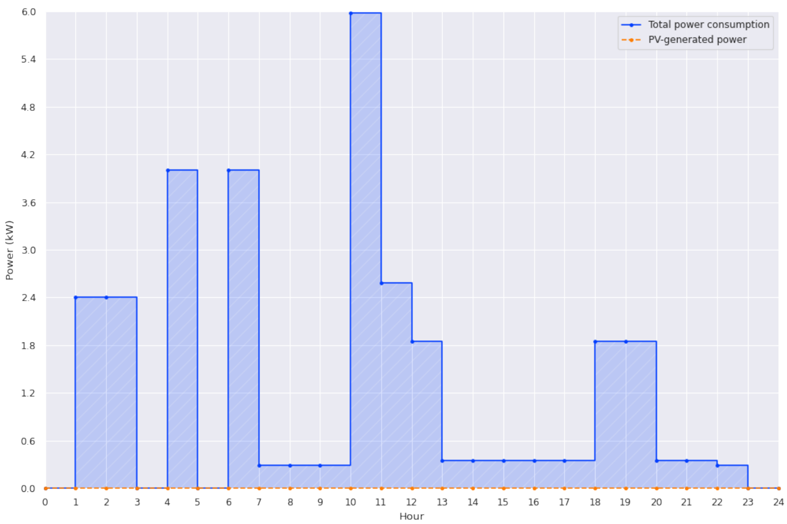


(b) Total power consumption against PV-generated power.

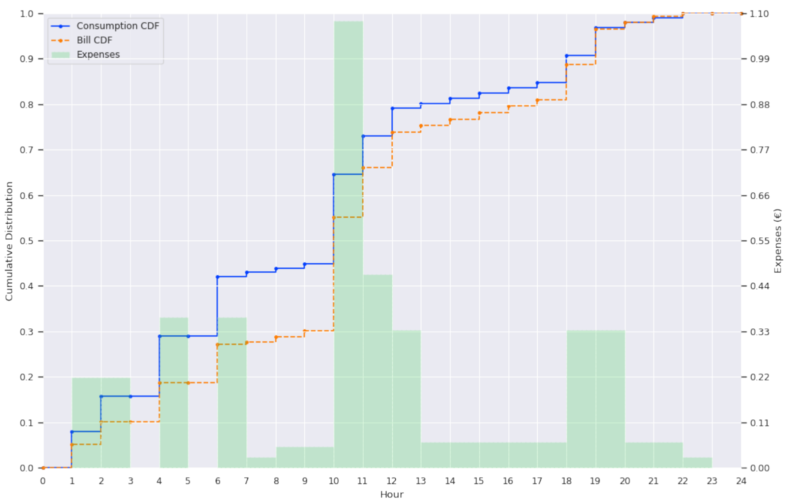
Figure C.13: Bill regulation with Dual-Rate tariff: results of the simulation following midday preference changes.



(a) Appliances' scheduling.

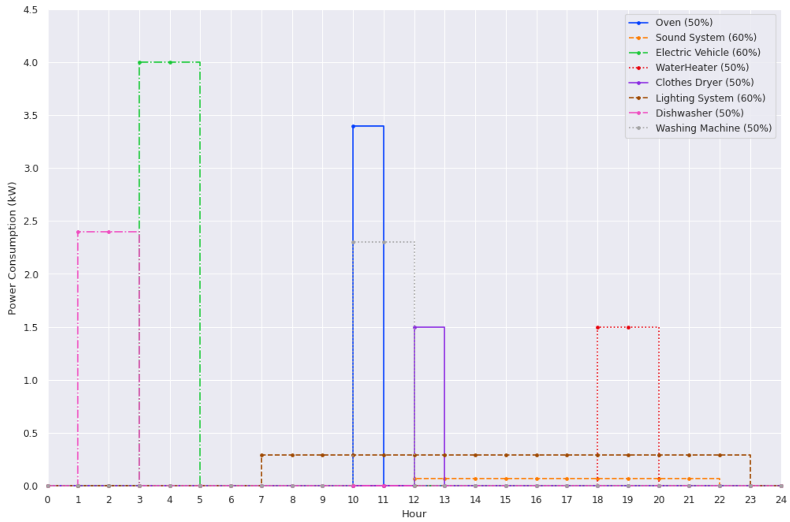


(b) Total power consumption against PV-generated power.

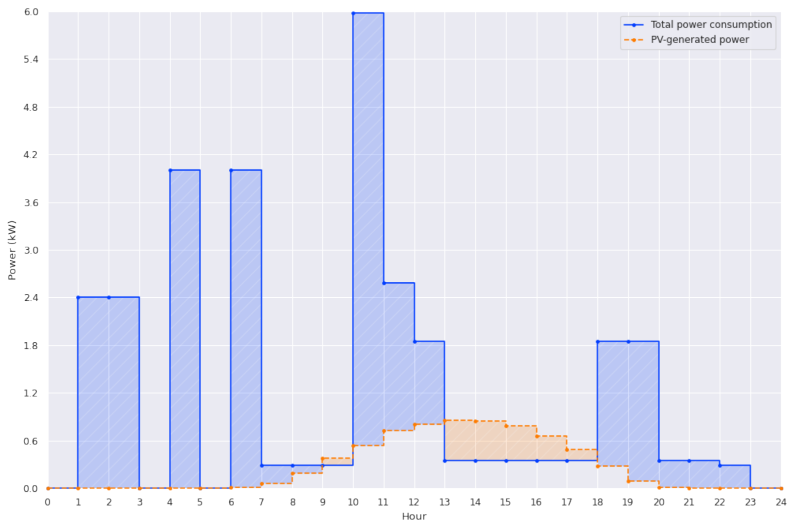


(c) Cumulative distribution functions and hourly spendings.

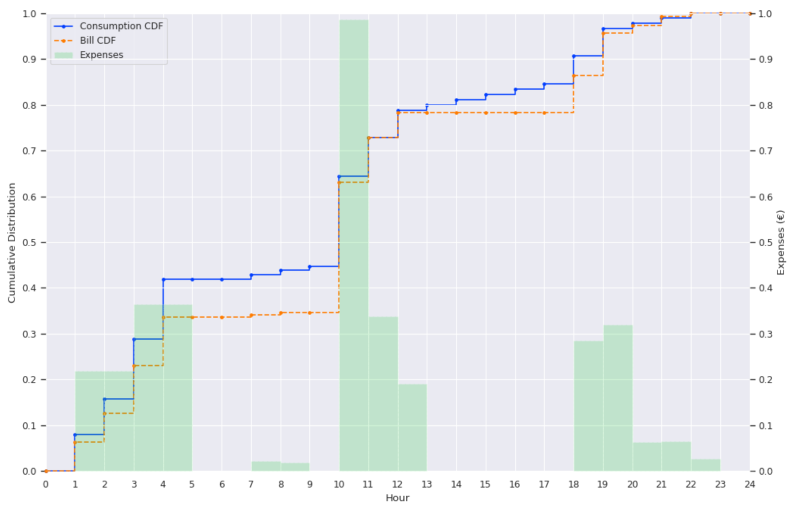
Figure C.14: Bill regulation with highest critical satisfaction value, Dual-Rate tariff and no PV: results of the simulation.



(a) Appliances' scheduling.

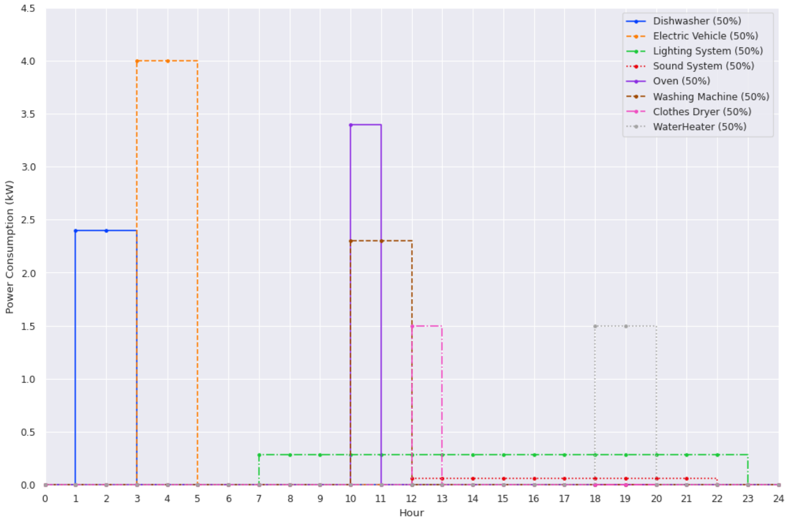


(b) Total power consumption against PV-generated power.

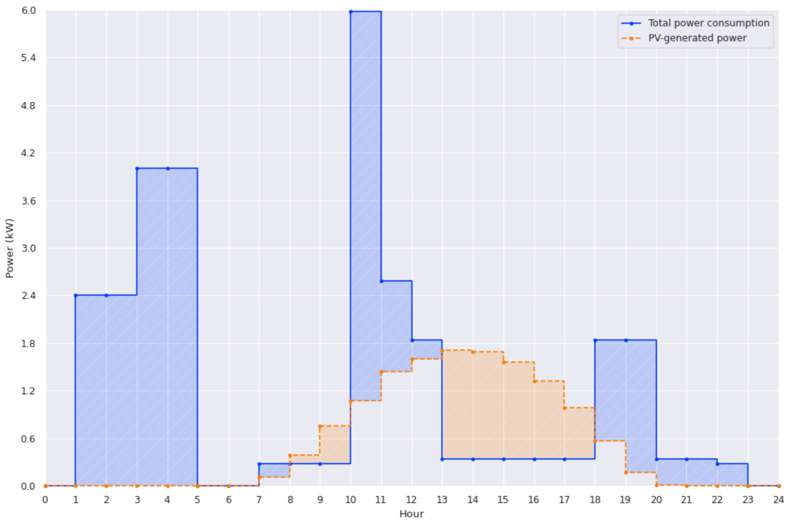


(c) Cumulative distribution functions and hourly spendings.

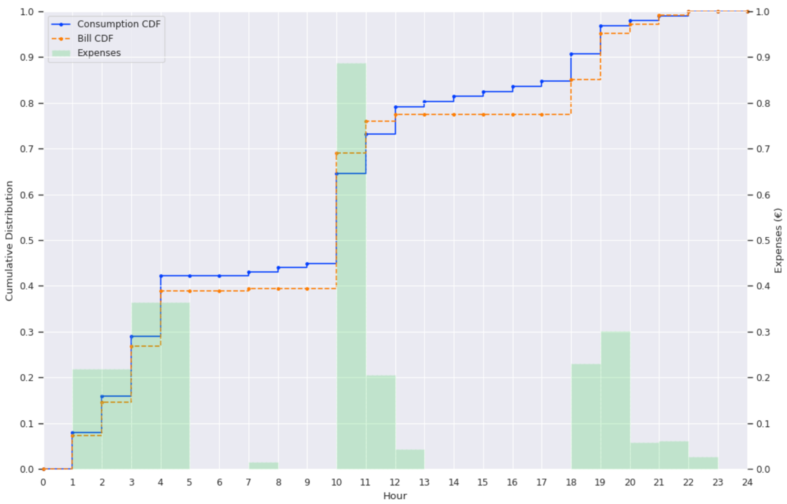
Figure C.15: Bill regulation with highest critical satisfaction value, Dual-Rate tariff and PV with 1 kW capacity: results of the simulation.



(a) Appliances' scheduling.

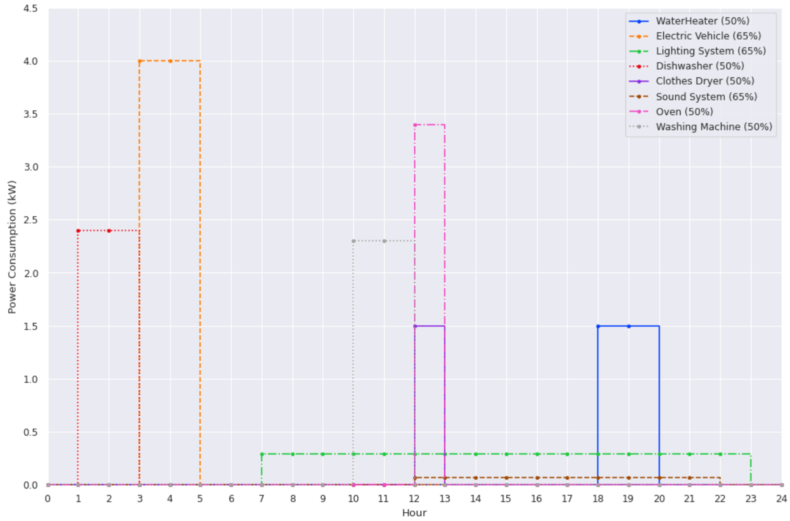


(b) Total power consumption against PV-generated power.

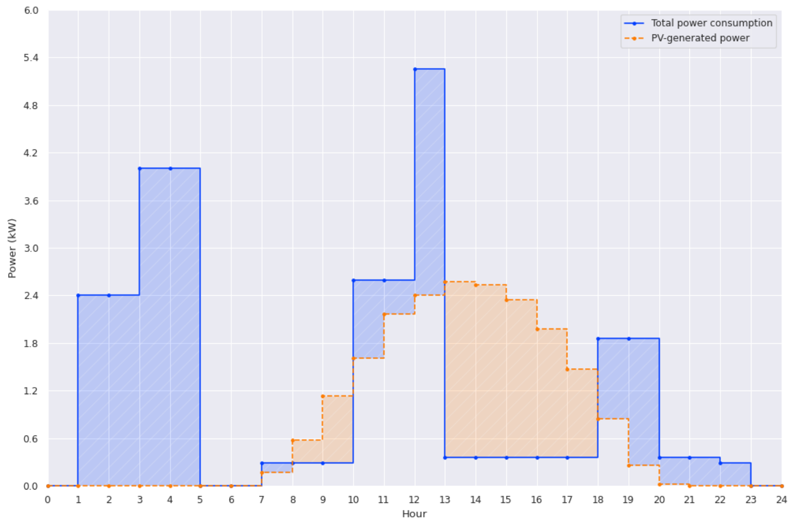


(c) Cumulative distribution functions and hourly spendings.

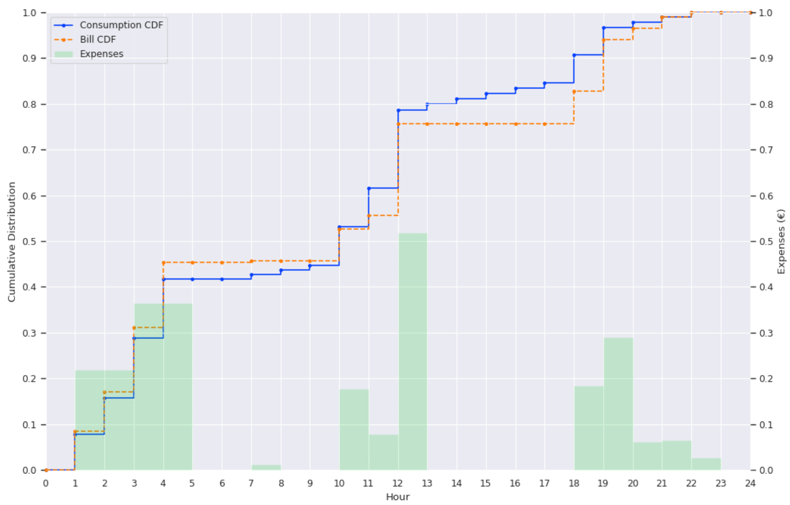
Figure C.16: Bill regulation with highest critical satisfaction value, Dual-Rate tariff and PV with 2 kW capacity: results of the simulation.



(a) Appliances' scheduling.

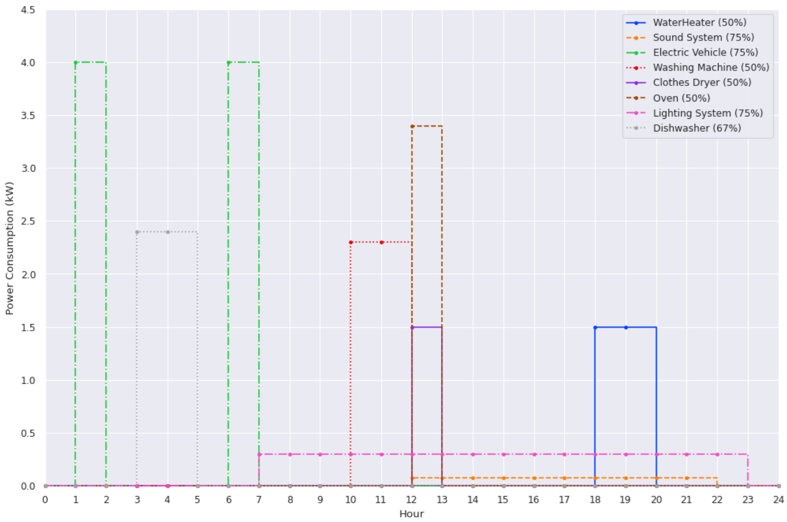


(b) Total power consumption against PV-generated power.

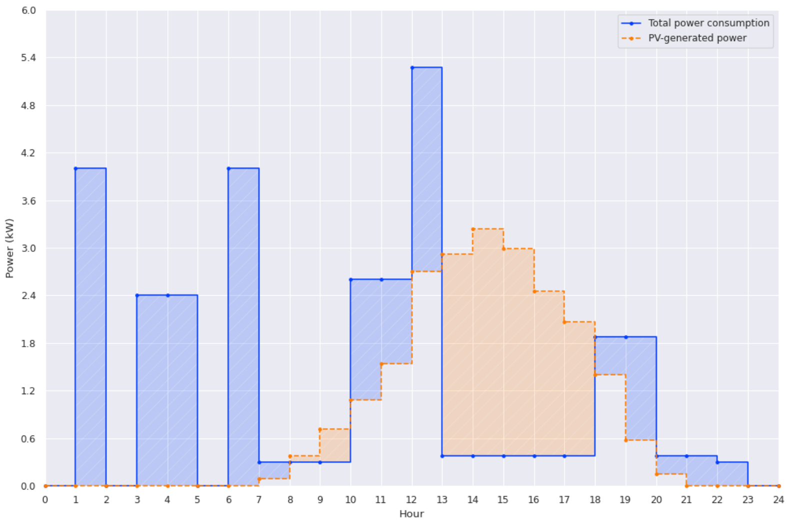


(c) Cumulative distribution functions and hourly spendings.

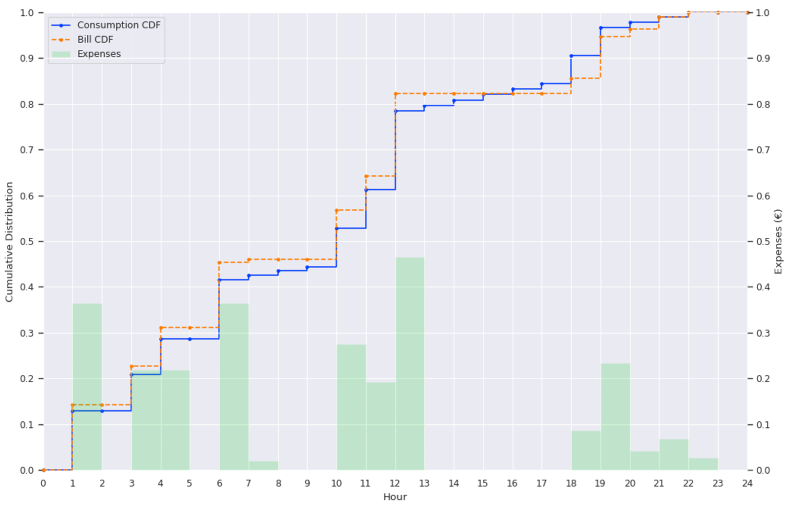
Figure C.17: Bill regulation with highest critical satisfaction value, Dual-Rate tariff and PV with 3 kW capacity: results of the simulation.



(a) Appliances' scheduling.



(b) Total power consumption against PV-generated power.



(c) Cumulative distribution functions and hourly spendings.

Figure C.18: Bill regulation with highest critical satisfaction value, Dual-Rate tariff and PV with 4 kW capacity: results of the simulation.

Bibliography

[Abras et al., 2010] Abras, S., Pesty, S., Ploix, S., and Jacomino, M. (2010). Advantages of MAS for the Resolution of a Power Management Problem in Smart Homes. In Demazeau, Y., Dignum, F., Corchado, J. M., and Pérez, J. B., editors, *Proceedings of Advances in Practical Applications of Agents and Multiagent Systems*, pages 269–278, Berlin, Heidelberg. Springer Berlin Heidelberg. DOI:10.1007/978-3-642-12384-9_32.

[Abras et al., 2008] Abras, S., Ploix, S., Pesty, S., and Jacomino, M. (2008). *A Multi-agent Home Automation System for Power Management*, pages 59–68. Springer Berlin Heidelberg, Berlin, Heidelberg. DOI:10.1007/978-3-540-79142-3_6.

[Addo et al., 2019] Addo, I. B., Thoms, M. C., and Parsons, M. (2019). The influence of water-conservation messages on reducing household water use. *Applied Water Science*, 9(5):126. DOI:10.1007/s13201-019-1002-0.

[Administration, 2020] Administration, U. E. I. (2020). Annual Energy Outlook 2020. Technical report, U.S. Energy Information Administration.

[Al-Fuqaha et al., 2015] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., and Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys Tutorials*, 17(4):2347–2376. DOI:10.1109/COMST.2015.2444095.

[Alaa et al., 2017] Alaa, M., Zaidan, A. A., Zaidan, B. B., Talal, M., and Kiah, M. L. M. (2017). A review of smart home applications based on Internet of Things. *Journal of Network and Computer Applications*, 97:48–65. DOI:10.1016/j.jnca.2017.08.017.

[Almusaylim and Zaman, 2019] Almusaylim, Z. A. and Zaman, N. (2019). A Review on Smart Home Present State and Challenges: Linked to Context-Awareness Internet of Things (IoT). *Wirel. Netw.*, 25(6):3193–3204. DOI:10.1007/s11276-018-1712-5.

[Andreas et al., 2019] Andreas, Aldawira, C. R., Putra, H. W., Hanafiah, N., Surjarwo, S., and Wibisurya, A. (2019). Door Security System for Home Monitoring Based on ESP32. *Procedia Computer Science*, 157:673–682. DOI:10.1016/j.procs.2019.08.218.

[Asare-Bediako et al., 2013a] Asare-Bediako, B., Kling, W., and Ribeiro, P. (2013a). Integrated agent-based home energy management system for smart grids applications. In *Proceedings of IEEE PES ISGT Europe 2013*, pages 1–5. DOI:10.1109/ISGTEurope.2013.6695332.

[Asare-Bediako et al., 2013b] Asare-Bediako, B., Kling, W., and Ribeiro, P. (2013b). Multi-agent system architecture for smart home energy management and optimization. In *Proceedings of IEEE PES ISGT Europe 2013*, pages 1–5. DOI:10.1109/ISGTEurope.2013.6695331.

[Asare-Bediako et al., 2012] Asare-Bediako, B., Ribeiro, P. F., and Kling, W. L. (2012). Integrated energy optimization with smart home energy management systems. In *Proceedings of 2012 3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, pages 1–8. DOI:10.1109/ISGTEurope.2012.6465696.

[Austin, 1975] Austin, J. L. (1975). *How to Do Things with Words*. Harvard University Press, Second edition.

[Barták et al., 2010] Barták, R., Salido, M. A., and Rossi, F. (2010). Constraint satisfaction techniques in planning and scheduling. *Journal of Intelligent Manufacturing*, 21(1):5–15. DOI:10.1007/s10845-008-0203-4.

[Beaudin and Zareipour, 2015] Beaudin, M. and Zareipour, H. (2015). Home energy management systems: A review of modelling and complexity. *Renewable and Sustainable Energy Reviews*, 45:318–335. DOI:10.1016/j.rser.2015.01.046.

[Bellifemine et al., 2007] Bellifemine, F. L., Caire, G., and Greenwood, D. (2007). *Developing Multi-Agent Systems with JADE (Wiley Series in Agent Technology)*. John Wiley & Sons, Inc., Hoboken, NJ, USA, First edition.

[Betts and Müller, 2014] Betts, J. and Müller, B. (2014). Engineering MAS – A Device Integration Framework for Smart Home Environments. *Fundamenta Informaticae*, 135(4):341–353. DOI:10.3233/FI-2014-1127.

[Bond and Gasser, 2014] Bond, A. H. and Gasser, L. (2014). *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[Bordini and Hübner, 2006] Bordini, R. H. and Hübner, J. F. (2006). BDI Agent Programming in AgentSpeak Using Jason. In Toni, F. and Torroni, P., editors, *Proceedings of Computational Logic in Multi-Agent Systems*, pages 143–164, Berlin, Heidelberg. Springer Berlin Heidelberg. DOI:10.1007/11750734_9.

[Bresciani et al., 2004] Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., and Mylopoulos, J. (2004). Tropos: An Agent-Oriented Software Development Methodology. *Autonomous Agents and Multi-Agent Systems*, 8:203–236. DOI:10.1023/B:AGNT.0000018806.20944.ef.

[Britz et al., 2014] Britz, J., Frey, J., and Alexandersson, J. (2014). Bridging the Gap between Smart Home and Agents. In *Proceedings of 2014 International Conference on Intelligent Environments*, page 31–38. IEEE. DOI:10.1109/IE.2014.12.

[Carneiro et al., 2018] Carneiro, J., Rossetti, R. J. F., Silva, D. C., and Oliveira, E. C. (2018). Bim, gis, iot, and ar/vr integration for smart maintenance and management of road networks: a review. In *2018 IEEE International Smart Cities Conference (ISC2)*, pages 1–7.

[Celik et al., 2017] Celik, B., Roche, R., Bouquain, D., and Miraoui, A. (2017). Coordinated neighborhood energy sharing using game theory and multi-agent systems. In *Proceedings of 2017 IEEE Manchester PowerTech*, page 1–6. IEEE. DOI:10.1109/PTC.2017.7980820.

[CISCO, 2020] CISCO (2020). Cisco Annual Internet Report (2018-2023). Technical report, CISCO.

[Cornel - Cristian et al., 2019] Cornel - Cristian, A., Gabriel, T., Arhip-Calin, M., and Zamfirescu, A. (2019). Smart home automation with MQTT. In *Proceedings of 2019 54th International Universities Power Engineering Conference (UPEC)*, pages 1–5. DOI:10.1109/UPEC.2019.8893617.

[Cunha et al., 2020] Cunha, M. E. S., Rossetti, R. J. F., and Campos, P. (2020). Modelling smart cities through socio-technical systems. In *2020 IEEE International Smart Cities Conference (ISC2)*, pages 1–8.

[de la Vega et al., 2018] de la Vega, F., Soriano, J., Jimenez, M., and Lizcano, D. (2018). A Peer-to-Peer Architecture for Distributed Data Monetization in Fog Computing Scenarios. *Wireless Communications and Mobile Computing*, 2018:1–15. DOI:10.1155/2018/5758741.

[Dorri et al., 2018] Dorri, A., Kanhere, S. S., and Jurdak, R. (2018). Multi-Agent Systems: A Survey. *IEEE Access*, 6:28573–28593. DOI:10.1109/ACCESS.2018.2831228.

[Dorri et al., 2017] Dorri, A., Kanhere, S. S., Jurdak, R., and Gauravaram, P. (2017). Blockchain for IoT security and privacy: The case study of a smart home. In *Proceedings of 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, page 618–623. IEEE. DOI:10.1109/PERCOMW.2017.7917634.

[European Commission and Technology, 2020] European Commission, Directorate-General for Communications Networks, C. and Technology (2020). Communication From The Commission To The European Parliament, The Council, The European Economic And Social Committee And The Committee Of The Regions: A European strategy for data. Technical report, CNECT.

[Feron and Monti, 2017] Feron, B. and Monti, A. (2017). Integration of space heating demand flexibility in a home energy management system using a market-based multi agent system. In *Proceedings of 2017 IEEE Power & Energy Society General Meeting*, pages 1–5. DOI:10.1109/PESGM.2017.8273810.

[Gazafroudi et al., 2017] Gazafroudi, A. S., De Paz, J. F., Prieto-Castrillo, F., Villarrubia, G., Talari, S., Shafie-khah, M., and Catalão, J. P. S. (2017). A Review of Multi-agent Based Energy Management Systems. In De Paz, J. F., Julián, V., Villarrubia, G., Marreiros, G., and Novais, P., editors, *Proceedings of Ambient Intelligence- Software and Applications - 8th International Symposium on Ambient Intelligence (ISAmI 2017)*, pages 203–209, Cham. Springer International Publishing. DOI:10.1007/978-3-319-61118-1_25.

[Harper, 2003] Harper, R. (2003). *Inside the Smart Home*. Springer-Verlag London, First edition. DOI:10.1007/b97527.

[Jabbar et al., 2019] Jabbar, W. A., Shang, H. K., Hamid, S. N. I. S., Almohammedi, A. A., Ramli, R. M., and Ali, M. A. H. (2019). IoT-BBMS: Internet of Things-Based Baby Monitoring System for Smart Cradle. *IEEE Access*, 7:93791–93805. DOI:10.1109/ACCESS.2019.2928481.

[Jennings et al., 1998] Jennings, N. R., Sycara, K., and Wooldridge, M. (1998). A Roadmap of Agent Research and Development. *Autonomous Agents and Multi-Agent Systems*, 1(1):7–38. DOI:10.1023/A:1010090405266.

[Kang et al., 2018] Kang, E. S., Pee, S. J., Song, J. G., and Jang, J. W. (2018). A Blockchain-Based Energy Trading Platform for Smart Homes in a Microgrid. In *Proceedings of 2018 3rd International Conference on Computer and Communication Systems (ICCCS)*, page 472–476. IEEE. DOI:10.1109/CCOMS.2018.8463317.

[Khanna and Kaur, 2020] Khanna, A. and Kaur, S. (2020). Internet of Things (IoT), Applications and Challenges: A Comprehensive Review. *Wireless Personal Communications*, 114(2):1687–1762. DOI:10.1007/s11277-020-07446-4.

[Kim et al., 2015] Kim, J. Y., Lee, H., Son, J., and Park, J. (2015). Smart home web of objects-based IoT management model and methods for home data mining. In *Proceedings of 2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pages 327–331. DOI:10.1109/APNOMS.2015.7275448.

[Kinny and Georgeff, 1997] Kinny, D. and Georgeff, M. (1997). Modelling and design of multi-agent systems. In Müller, J. P., Wooldridge, M. J., and Jennings, N. R., editors, *Proceedings of Intelligent Agents III Agent Theories, Architectures, and Languages*, pages 1–20, Berlin, Heidelberg. Springer Berlin Heidelberg. DOI:10.1007/BFb0013569.

[Kouzinopoulos et al., 2018] Kouzinopoulos, C. S., M. Giannoutakis, K., Votis, K., Tzovaras, D., Collen, A., Nijdam, N. A., Konstantas, D., Spathoulas, G., Pandey, P., and Katsikas, S. (2018). Implementing a Forms of Consent Smart Contract on an IoT-based Blockchain to promote user trust. In *Proceedings of 2018 Innovations in Intelligent Systems and Applications (INISTA)*, page 1–6. IEEE. DOI:10.1109/INISTA.2018.8466268.

[Leitão et al., 2020] Leitão, J., Gil, P., Ribeiro, B., and Cardoso, A. (2020). A Survey on Home Energy Management. *IEEE Access*, 8:5699–5722. DOI:10.1109/ACCESS.2019.2963502.

[Li et al., 2016] Li, W., Logenthiran, T., Woo, W. L., Phan, V., and Srinivasan, D. (2016). Implementation of demand side management of a smart home using multi-agent system. In *Proceedings of 2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 2028–2035. DOI:10.1109/CEC.2016.7744037.

[Liang et al., 2008] Liang, C.-H., Hung, W.-S., Hsieh, M.-C., Wu, C.-M., and Luo, C.-H. (2008). A Multi-agent Based Architecture for an Assistive User Interface of Intelligent Home Environment Control. In *Proceedings of 2008 Eighth International Conference on Intelligent Systems Design and Applications*, page 335–338. IEEE. DOI:10.1109/ISDA.2008.320.

[Liu et al., 2013] Liu, S., Araujo, M., Brunskill, E., Rossetti, R., Barros, J., and Krishnan, R. (2013). Understanding sequential decisions via inverse reinforcement learning. In *2013 IEEE 14th International Conference on Mobile Data Management*, volume 1, pages 177–186.

[Liu et al., 2016] Liu, Y., Qiu, B., Fan, X., Zhu, H., and Han, B. (2016). Review of Smart Home Energy Management Systems. *Energy Procedia*, 104:504–508. DOI:10.1016/j.egypro.2016.12.085.

[Lotfi et al., 2019] Lotfi, J., Abdi, F., Abbou, M. F., and Chaatit, F. (2019). Integration of the Multi Agent System Paradigm and ZigBee Smart Home Area Network. In *Proceedings of 2019 7th International Renewable and Sustainable Energy Conference (IRSEC)*, page 1–4. IEEE. DOI:10.1109/IRSEC48032.2019.9078147.

[Martins et al., 2021] Martins, S., Rossetti, R. J. F., and Kokkinogenis, Z. (2021). A resource-based model for home energy management using multi-agent systems. In *2021 IEEE International Smart Cities Conference (ISC2)*, pages 1–4, Virtual Conference. (Sept. 7–10, 2021).

[Melo et al., 2019] Melo, L., Sampaio, R., Leao, R., Barroso, G., and Bezerra, J. (2019). Python-based multi-agent platform for application on power grids. *International Transactions on Electrical Energy Systems*, 29. DOI:10.1002/2050-7038.12012.

[Mezquita et al., 2019] Mezquita, Y., Gazafroudi, A. S., Corchado, J. M., Shafie-Khah, M., Laaksonen, H., and Kamisalic, A. (2019). Multi-Agent Architecture for Peer-to-Peer Electricity Trading based on Blockchain Technology. In *Proceedings of 2019 XXVII International Conference on Information, Communication and Automation Technologies (ICAT)*, page 1–6. IEEE. DOI:10.1109/ICAT47117.2019.8938926.

[Muhammad et al., 2019] Muhammad, A., Afzal, B., Imran, B., Tanwir, A., Akbar, A. H., and Shah, G. (2019). oneM2M Architecture Based Secure MQTT Binding in Mbed OS. In *Proceedings of 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, page 48–56. IEEE. DOI:10.1109/EuroSPW.2019.00012.

[Nations, 2014] Nations, U. (2014). The 17 Sustainable Development Goals. Available at `https://op.europa.eu/s/oJNV`. Accessed 2020-12-18.

[Nations, 2021] Nations, U. (2021). Sustainable cities and human settlements. Available at https://sdgs.un.org/topics/sustainable-cities-and-human-settlements. Accessed 2020-12-18.

[Ogbeiwi, 2017] Ogbeiwi, O. (2017). Why written objectives need to be really SMART. *British Journal of Healthcare Management*, 23:324–336. DOI:10.12968/bjhc.2017.23.7.324.

[Palensky and Dietrich, 2011] Palensky, P. and Dietrich, D. (2011). Demand Side Management: Demand Response, Intelligent Energy Systems, and Smart Loads. *IEEE Transactions on Industrial Informatics*, 7(3):381–388. DOI:10.1109/TII.2011.2158841.

[Parsons and Wooldridge, 2002] Parsons, S. and Wooldridge, M. (2002). Game Theory and Decision Theory in Multi-Agent Systems. *Autonomous Agents and Multi-Agent Systems*, 5(3):243–254. DOI:10.1023/A:1015575522401.

[Passos et al., 2011] Passos, L. S., Rossetti, R. J. F., and Gabriel, J. (2011). An agent methodology for processes, the environment, and services. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 2124–2129.

[Pentland et al., 2016] Pentland, A., Shrier, D., Hardjono, T., and Wladawsky-Berger, I. (2016). Towards an Internet of Trusted Data: A New Framework for Identity and Data Sharing. Technical report, MIT.

[Raiffa, 1988] Raiffa, H. (1988). *Decision analysis : introductory lectures on choices under uncertainty*. Random House, First edition.

[Rao, 1996] Rao, A. S. (1996). AgentSpeak(L): BDI agents speak out in a logical computable language. In Van de Velde, W. and Perram, J. W., editors, *Proceedings of Agents Breaking Away*, pages 42–55, Berlin, Heidelberg. Springer Berlin Heidelberg. DOI:10.1007/BFb0031845.

[Rossetti et al., 2002] Rossetti, R. J. F., Bordini, R. H., Bazzan, A. L. C., Bampi, S., Liu, R., and Vliet, D. V. (2002). Using BDI agents to improve driver modelling in a commuter scenario. *Transportation Research Part C: Emerging Technologies*, 10(5):373–398. DOI:10.1016/S0968-090X(02)00027-X.

[Rúbio et al., 2017] Rúbio, T. R. P. M., Kokkinogenis, Z., Cardoso, H. L., Oliveira, E., and Rossetti, R. J. F. (2017). ResMAS - A Conceptual MAS Model for Resource-Based Integrated Markets. In Bajo, J., Vale, Z., Hallenborg, K., Rocha, A. P., Mathieu, P., Pawlewski, P., Del Val, E., Novais, P., Lopes, F., Duque Méndez, N. D., Julián, V., and Holmgren, J., editors, *Proceedings of Highlights of Practical Applications of Cyber-Physical Multi-Agent Systems*, pages 117–129, Cham. Springer International Publishing. DOI:10.1007/978-3-319-60285-1_10.

[Rúbio et al., 2019] Rúbio, T. R. P. M., Lopes Cardoso, H., and Oliveira, E. (2019). Agent process modelling. In Demazeau, Y., Matson, E., Corchado, J. M., and De la Prieta, F., editors, *Advances in Practical Applications of Survivable Agents and Multi-Agent Systems: The PAAMS Collection*, pages 277–280, Cham. Springer International Publishing.

[Russell and Norvig, 2020] Russell, S. J. and Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*. Pearson, Boston, Fourth edition.

[Said and Masud, 2013] Said, O. and Masud, M. (2013). Towards Internet of Things: Survey and Future Vision. *International Journal of Computer Networks*, 5:1–17.

[Sommerville, 2010] Sommerville, I. (2010). *Software Engineering*. Addison-Wesley Publishing Company, USA, Ninth edition.

[Sovacool and Furszyfer Del Rio, 2020] Sovacool, B. K. and Furszyfer Del Rio, D. D. (2020). Smart home technologies in Europe: A critical review of concepts, benefits, risks and policies. *Renewable and Sustainable Energy Reviews*, 120:109663. DOI:10.1016/j.rser.2019.109663.

[Sun et al., 2013] Sun, Q., Yu, W., Kochurov, N., Hao, Q., and Hu, F. (2013). A Multi-Agent-Based Intelligent Sensor and Actuator Network Design for Smart House and Home Automation. page 32. DOI:10.3390/jsan2030557.

[Taiwo et al., 2020] Taiwo, O., Gabralla, L. A., and Ezugwu, A. E. (2020). Smart Home Automation: Taxonomy, Composition, Challenges and Future Direction. In Gervasi, O., Murgante, B., Misra, S., Garau, C., Blečić, I., Taniar, D., Apduhan, B. O., Rocha, A. M. A. C., Tarantino, E., Torre, C. M., and Karaca, Y., editors, *Proceedings of Computational Science and Its Applications – ICCSA 2020*, pages 878–894, Cham. Springer International Publishing. DOI:10.1007/978-3-030-58817-5_62.

[Tom et al., 2020] Tom, R. J., Sankaranarayanan, S., and Rodrigues, J. J. (2020). Agent negotiation in an IoT-Fog based power distribution system for demand reduction. *Sustainable Energy Technologies and Assessments*, 38:100653. DOI:10.1016/j.seta.2020.100653.

[van Moergestel et al., 2013] van Moergestel, L., Langerak, W., Meerstra, G., van Nieuwenburg, N., Pape, F., Telgen, D., Puik, E., and Meyer, J.-J. (2013). Agents in Domestic Environments. In *Proceedings of 2013 19th International Conference on Control Systems and Computer Science*, page 487–494. IEEE. DOI:10.1109/CSCS.2013.57.

[Vlachostergiou et al., 2016] Vlachostergiou, A., Stratogiannis, G., Caridakis, G., Siolas, G., and Mylonas, P. (2016). User Adaptive and Context-Aware Smart Home Using Pervasive and Semantic Technologies. *Journal of Electrical and Computer Engineering*, 2016:1–20. DOI:10.1155/2016/4789803.

[Winikoff and Padgham, 2004] Winikoff, M. and Padgham, L. (2004). *The Prometheus Methodology*, pages 217–234. Springer US, Boston, MA. DOI:10.1007/1-4020-8058-1_14.

[Wooldridge, 2009] Wooldridge, M. (2009). *An Introduction to Multiagent Systems*. Wiley, Chichester, UK, Second edition.

[Wooldridge and Jennings, 1995] Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: theory and practice. *The Knowledge Engineering Review*, 10(2):115–152. DOI:10.1017/S0269888900008122.

[Wooldridge et al., 2000] Wooldridge, M., Jennings, N. R., and Kinny, D. (2000). The Gaia Methodology for Agent-Oriented Analysis and Design. *Autonomous Agents and Multi-Agent Systems*, 3(3):285–312. DOI:10.1023/A:1010071910869.

[Xu et al., 2018] Xu, Q., He, Z., Li, Z., and Xiao, M. (2018). Building an Ethereum-Based Decentralized Smart Home System. In *Proceedings of 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, page 1004–1009. IEEE. DOI:10.1109/PADSW.2018.8644880.

[Yokoo and Hirayama, 2001] Yokoo, M. and Hirayama, K. (2001). Algorithms for Distributed Constraint Satisfaction: A Review. *Autonomous Agents and Multi-Agent Systems*, 3:185–207. DOI:10.1023/A:1010078712316.

[Zhang et al., 2018] Zhang, J., Zhou, Z., Li, S., Gan, L., Zhang, X., Qi, L., Xu, X., and Dou, W. (2018). Hybrid computation offloading for smart home automation in mobile cloud computing. *Personal and Ubiquitous Computing*, 22(1):121–134. DOI:10.1007/s00779-017-1095-0.