

Poster Abstract: Towards a Smart Home Framework

Muddasser Alam, Alper T. Alan, Alex Rogers, and Sarvapali D. Ramchurn
Agents, Interaction and Complexity Research Group, University of Southampton, UK.
{moody,ata1g11,acr,sdr}@ecs.soton.ac.uk

ABSTRACT

We present our Smart Home Framework (SHF) which simplifies the modelling, prototyping and simulation of *smart infrastructure* (i.e., smart home and smart communities). It provides the buildings blocks (e.g., home appliances) that can be extended and assembled together to build a smart infrastructure model to which appropriate AI techniques can be applied. This approach enables rapid modelling where new research initiatives can build on existing work.

Categories and Subject Descriptors

D.2.13 [Reusable Software]: Reuse models, libraries

1. INTRODUCTION

Smart infrastructure such as smart homes, smart buildings and smart communities are considered to be an integral part of the transition to a low-carbon economy. This infrastructure is intended to embody many technologies such as smart meters, energy-efficient appliances, renewable-energy generation units, microstorage and electric vehicles. Hence, a number of research initiatives have focused on the role and interplay of these technologies in smart infrastructure. The majority of work in this line involves similar activities such as building a model of smart building, incorporating intelligence and simulation [3, p. 132], which have resulted in a number of shared and reusable tools, particularly, a large number of simulation software, with notable examples of GridLab-D, Open-DSS, Simulink-SmartGrid and MapleSim. However, other activities have not received their due attention, especially in the modelling phase where there is an absence of shared tools and resources that can help create models of smart infrastructure. We believe that the development of such modelling resources will enable new research initiatives to build upon existing work, rather than starting from scratch. As a step towards this vision, we present a *smart home framework* (SHF): a software framework that provides the *building blocks* (e.g. appliances, generators and storage) that can be *assembled* together to build general

models of smart homes and communities. These components can be extended to construct specialised models to meet the specific needs of any particular project. This approach enables rapid, bottom-up modelling of smart homes and communities where these components and models can be shared and reused. We envision that such reuse will eventually lead to standardised models (e.g., a standard model of a washing machine or a smart home) to be shared (e.g., via online repositories) and reused across the community. In addition, the SHF comes with a software *toolkit* that helps in optimisation and visualisation of data. Finally, the SHF is aimed at enabling collaboration between researchers and is available as a free, open-source framework to all.

2. RELATED WORK

The core purpose of our SHF is to enable researchers to quickly build the models of smart homes and communities by assembling general components. An example is to build a smart home by putting together components such as a solar panel, some appliances, an electric battery and an electric vehicle or any other customised component. We note that the majority of simulation software come with *parameterised models* of common components which can be customised using the parameters values. For example, GridLab-D and MapleSim allow users to specify parameters for an appliance (e.g., *energy-baseline* (kWh) or *motor-power* (kW) for a washing machine). However, such models are tightly integrated into the simulator and there is little flexibility of extending them into new models. For example, it can be difficult to extend an existing washing machine model into one that has a built-in dryer. In contrast, the SHF provides models that can be easily extended into specialised models. This focus on modelling, rather than simulation, is the core difference between the SHF and simulation software and therefore, the SHF should be considered as a *modelling tool* and not as a replacement of the simulation software. Indeed, there may be cases where both can be used together. An example is where the SHF-based smart home models can be integrated with a simulator for very large-scale simulation.

3. SMART HOME FRAMEWORK

Our SHF architecture is based on a bottom-up, extendible and plug-in methodology. The basic components are defined in a flexible hierarchy (e.g., *SolarPanel* implements *IRenewableGenerator* extends *IOOnSiteGenerator* extends *IGenerator*, see [4] for UML diagrams) making it easier to include new code. The plug-in approach is useful to replace the existing code, for example, a new optimiser can be *plugged-in*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Buildsys'13, November 13, 2013, Rome, Italy.
Copyright 2013 ACM 978-1-4503-1170-0 ...\$15.00.

to replace the default optimiser in the toolkit.

3.1 Building a Smart Home Model

The SHF provides a rich set of interfaces, abstract and implemented classes as we summarise now. Detailed UML diagrams, tutorials and example code are available online [4].

Generation: The SHF provides classes to model (i) onsite generation and (ii) the electricity grid. Onsite generation include renewable and non-renewable (e.g., combined heat-power) generation. An energy source has certain attributes and operations depending on its type. For instance, the onsite non-renewable have an operational cost (monetary and/or carbon footprint) while the renewable have uncertain power outputs. The generation module contains implemented classes for a number of energy sources that can be readily instantiated. A new type of generator can be added by extending the appropriate interface (e.g., *IRenewableEnergySource* or *INonRenewableEnergySource*). Furthermore, this module has tariff classes (e.g., standard and off-peak) that can connect a smart home to the grid. Additional classes exist to include more sophisticated types of tariffs.

Storage: The SHF contains classes to model (i) onsite storage (e.g., electric batteries at homes) and (ii) mobile storage. Each storage class has different attributes (e.g., onsite storage is assumed to be available all the time). The SHF provides basic models of electric batteries in terms of their capacity, charging and discharging rates and efficiency while more sophisticated models can be created by extension.

Load: Load requirements are described in terms of appliances and their usage (i.e., *load events*). Common appliances (e.g., oven, fridge) are already implemented and more sophisticated models can be added. A load event describes a single use of an appliance and has a certain type and attributes. The supported load event types are: (i) deferrable and non-deferrable (ii) interruptible and non interruptible (iii) critical (iv) baseload (v) or combinations (e.g. a deferrable, interruptible and load event). It also supports *simple loads* to model loads in an appliance-independent way.

Miscellaneous: The SHF also supports HVAC (heating, ventilation, air conditioning) systems and electric vehicles.

3.2 Building a Smart Community

The SHF can be used to create multiple smart homes that collectively can represent a smart community. This community can be off-grid or connected to the grid. When a community model is ready it can be optimised or simulated (either using the SHF optimisation package or a third-party simulator) as a whole [1] or on an individual homes basis [2].

3.3 Additional Toolkits

The SHF comes with a toolkit that contains (i) an optimisation and (ii) a visualisation package. The optimisation package includes a default optimiser (i.e., IBM CPLEX, available free of cost to academia) that supports convex optimisation and for more sophisticated or non-convex optimisation, a third-party optimiser can be *plugged-in* to replace it. The visualisation package is essentially a customised wrapper around the JFreeChart library that can be used to visualise data. For example, it can be used to quickly plot the load events or the battery charging in a smart home. Figure 1 shows an example where a smart home model is built and optimised using the default appliances/devices and CPLEX optimiser and results are plotted via the visual toolkit. No

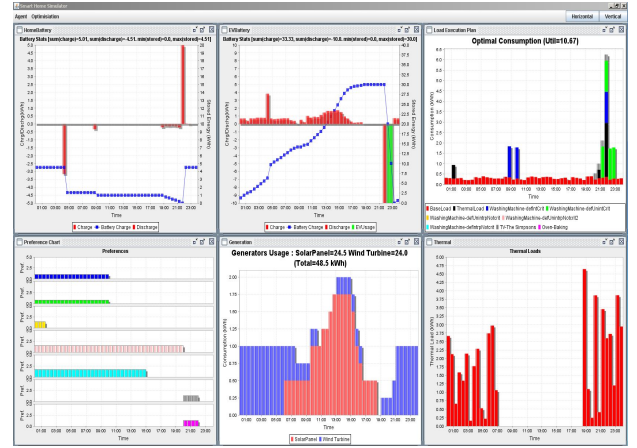


Figure 1: Graphs (battery, EV, loads, preferences, generation, HVAC) show optimal use of resources in a home. additional code was needed outside the framework.

4. APPLIED EXAMPLES AND RESOURCES

We have utilised the SHF to enables smart homes to exchange energy. For example, using the SHF we created an off-grid community where each home has a microgeneration unit, an electric battery and some load. We then showed that the cooperative exchange of energy in such communities can reduce the overall battery usage by 65% [1]. More recently, we presented a negotiation protocol to facilitate decentralised energy exchange between self-interested agents to reduce their battery usage [2]. In addition to this work, we believe that the SHF can be especially useful for research projects focused on exploring demand response, uncertainty in renewable generation, optimal energy usage for cost and carbon emissions reduction, electric vehicle charging, coalition formation for group tariffs and intelligent heating.

The SHF website [4] has the tutorials and example code to help get started with the SHF. We also provide examples to build a model of (i) a smart home and (ii) a smart community using the SHF and its toolkit. The SHF code is hosted on GitHub and is freely available under GP Licence.

5. FUTURE WORK

At present, the SHF provides minimal support to model utilities other than electricity (e.g., gas and water). Our future work will provide modelling capacity for these utilities.

6. REFERENCES

- [1] M. Alam, S. D. Ramchurn, and A. Rogers. Cooperative energy exchange for the efficient use of energy and resources in remote communities. In *International Conference on Autonomous Agents and Multiagent Systems*, Saint Paul, Minnesota, USA, 6-10 May 2013.
- [2] M. Alam, A. Rogers, and S. D. Ramchurn. Interdependent multi-issue negotiation for energy exchange in remote communities. In *Proceedings of AAAI*, Washington, USA, 14-18 July 2013.
- [3] J. C. Augusto and C. D. Nugent. *Designing smart homes: the role of artificial intelligence*, volume 4008. Springer-Verlag NY Inc., 2006.
- [4] Smart Home Framework. Available online at. www.smarthomeframework.org.