



**HAL**  
open science

# Architecture and Functional Framework for Home Energy Management Systems

Kornschnok Dittawit, Finn Arve Aagesen

► **To cite this version:**

Kornschnok Dittawit, Finn Arve Aagesen. Architecture and Functional Framework for Home Energy Management Systems. 19th Open European Summer School (EUNICE), Aug 2013, Chemnitz, Germany. pp.173-184, 10.1007/978-3-642-40552-5\_16 . hal-01497014

**HAL Id: hal-01497014**

**<https://inria.hal.science/hal-01497014v1>**

Submitted on 28 Mar 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Architecture and Functional Framework for Home Energy Management Systems

Kornschnok Dittawit and Finn Arve Aagesen

Department of Telematics,  
Norwegian University of Science and Technology, Norway  
{kornschnok,finnarve}@item.ntnu.no

**Abstract.** Residential consumers contribute to a substantial amount of electricity consumption. To ensure grid stability, a Home Energy Management System (EMS) that assists household inhabitants in operating home devices to achieve optimized energy usage is needed. Previous designs of Home EMS often have some issues. First, it places high implementation overhead on device vendors to manufacture devices compatible with the system. Second, it lacks the ability to scale when new device types and external variables are introduced. This paper proposes an architecture model and a functional framework for Home EMS that are simple and scalable. The framework defines functions of Home EMS components. A policy-based reasoning mechanism is proposed as part of the framework for the optimization of home energy usage. In addition, the inter-working of Home EMS components is presented. It relies on a semantic web language for data representation to provide meaningful information for reasoning and harmonization with existing standards.

**Keywords:** energy management system, smart home, smart grid, demand side management, policy-based reasoning

## 1 Introduction

Today's power grid is the result of over hundred years of evolution [1]. At present, there is a vision to transform the present grid into what is denoted as the *Smart Grid*. In Smart Grid, distributed generation (DG) is promoted, consumers are offered more choices and real-time information, and information and communication technology is tightly integrated for better management. The market is also changing to become more deregulated with the entry of smaller power providers and the possibility of consumers to provide power. As a research area, Smart Grid is considered a multidisciplinary area that spans over multiple domains. This paper looks at the residential demand side of the power grid.

In the EU, residential electricity consumption in 2010 accounted for around one-third of the total consumption [2]. This ratio indicates the importance and the influence of households on the power grid. In other words, electricity usage by households affects grid stability as a whole. This can be seen especially during peak periods when power demand rises and causes a strain to the power grid. One

way of ensuring grid stability is the adjustment of consumer energy consumption pattern. To facilitate the adjustment process, assistance to consumers will be needed to decide on how to operate home devices to achieve optimized energy usage. New commercial products are being pushed out that include some energy management functions and there have been many attempts to design a central system to manage home devices. However, the problems with most these product and system designs are 1) it does not look at the hollistic view of the system and was designed to solve a specific problem with no concern on the other components that exist on the power grid 2) it requires specially manufactured devices in order to be compatible with the system which puts a lot of implementation overhead to device vendors and prevents market entry of the system and 3) it lacks scalability when new device types and external variables are introduced. Although it is not possible in practice to design a system to handle new unknown problems, it is possible to design a framework that will serve as a solid and scalable foundation for future solutions.

This paper proposes an architecture model and a functional framework for Home Energy Management System (EMS), a system in a house that assists household inhabitants in operating home devices to achieve optimized energy usage. The architecture model looks at a hollistic view of the system and takes into account the major entities on the power grid which are the Distribution System Operators (DSOs) and Energy Service Providers (ESPs). It was designed to be simple and scalable to enable rapid deployment and also to be integratedly easily with existing standards, namely Sensor Web Enablement [3] and IEC61970-501 [4] which is suggested for use in the power industry. Home EMS components, component functions, and Home EMS inter-working framework have been defined. A policy-based reasoning mechanism is also proposed for energy management as part of the functional framework.

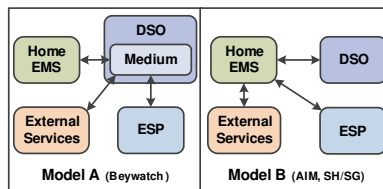
The remaining part of this paper is organized as follows. Section 2 discusses present technology and related works. Section 3 presents the proposed architecture and functional framework. Energy management in Home EMS is discussed in Section 4. Home EMS inter-working framework is presented in Section 5. Section 6 summarizes and concludes.

## 2 Present technology and related works

With the increasing concerns in energy conservation and energy efficiency, device vendors are responding to this issue by developing products that facilitate energy savings in households and are more energy efficient. Most of the released products provide the basic function to control the status of home appliances. For example, a consumer may control appliance statuses from the comfort of his mobile device and PC or set a timer to turn the appliances on or off [5] [6]. Some vendors provide a way for consumers to monitor energy consumption for an individual appliance with a specific apparatus [7]. Modern washing machines have a delay start function that allows consumers to set the start time for the washing machine so as to avoid peak periods. As useful as they are, these devices

were designed to solve very specific problems and the functions offered are only a small subset of functions that will be incorporated into consumer houses in the future. In addition, some functions such as the delay start function still require consumers to make difficult decisions based on various information sources with no assistance which results in most of these functions left unused. With some analysis, we would see that the value of these new appliances lies in the automation and scheduling of their basic operations, so what is needed is a decision support system that aids household inhabitants in controlling and coordinating appliances to optimize energy usage, i.e., Home EMS is needed.

There have been many research activities related to Home EMS including the effort by the European Commission which funded several research projects under the Seventh Framework Programme (FP7) such as AIM [8], Beywatch [9], and Smart House/Smart Grid (SH/SG) [10]. In AIM, the inter-working between home devices and the central system was well-designed and it is the same principle used in our proposal. The difference is our use of semantic web language for device information and message-based communication for operation and service invocation as opposed to packet-based solution in AIM. Our message-based solution provides greater flexibility and negates the need to establish standardized command codes before the manufacturing of home devices. The objective of Beywatch included the design of energy efficient white goods. As a result, the appliances that were compatible with Beywatch system had to be specifically manufactured. In SH/SG, energy management was done by algorithmic systems which means substantial effort will be required to change the logic if new device types and external variables are introduced into the picture. In terms of the system architecture, two reference models can be derived from the three projects based on the connectivity among the related systems as depicted in Fig. 1. In model A used by Beywatch, Home EMS is connected to an entity that hosts a medium which is further connected to power utilities and external service providers (SPs). The entity is called Service Provider in Beywatch which implies it can be any entity. In this case, we assume it is the DSO that takes the position of that entity and thus the utilities include only ESPs. In model B used by AIM and SH/SG, Home EMS is connected to DSO, ESPs, and external SPs directly. The comparison between the two models is presented in Table 1. The proposed architecture follows model B for the benefits described and its flexibility of consumers to switch or establish connections to other external SPs.



**Fig. 1.** Common architecture models for Home EMS

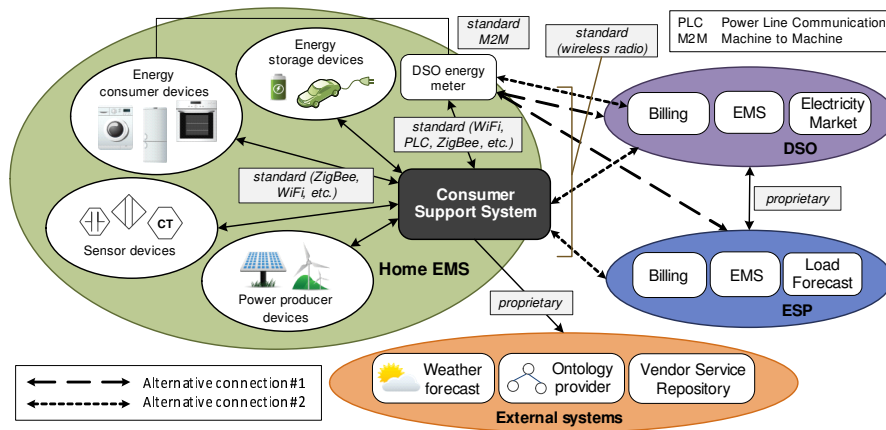
**Table 1.** Comparison between model A and B

| Model A   | Model B  |
|---|--|
| <p><i>Advantages:</i></p> <ul style="list-style-type: none"> <li>- Avoid the need to have communication infrastructure between Home EMS and ESPs</li> <li>- All consumers with the same DSO have equal access to external services</li> </ul> | <p><i>Advantages:</i></p> <ul style="list-style-type: none"> <li>- Promotes competition among the vendors of Home EMS to implement access to a variety of external services</li> <li>- Potentially less delay</li> </ul> |
| <p><i>Disadvantages:</i></p> <ul style="list-style-type: none"> <li>- There may be delay for data between Home EMS and ESPs</li> <li>- DSO has more responsibilities</li> </ul>   | <p><i>Disadvantages:</i></p> <ul style="list-style-type: none"> <li>- Need communication infrastructure between Home EMS and ESPs</li> </ul>   |

### 3 Home EMS architecture and component functions

#### 3.1 Overview

A Home EMS consists of various components, each responsible for different tasks and manufactured by different vendors. An overview of the architectural components and logical connections in Home EMS and external systems is illustrated in Fig. 2. There are three main component types residing in the house: Consumer Support System (CSS), home devices (e.g. energy consumer devices, sensor devices, power producer devices, energy storage devices.), and a DSO energy meter. Home EMS is connected to external systems including DSO, ESPs, and other service providers. The connection to DSO and ESPs can be via the meter (alternative #1) or CSS (alternative #2). In the latter, the meter also sends electricity usage data to DSO for billing. Section 3.2 and 3.3 explains the functions of the two most important component types, CSS and home devices.



**Fig. 2.** Home EMS architecture

### 3.2 Consumer Support System functionality

CSS is a computing device in Home EMS that is responsible for energy management, which is the monitoring, controlling, and coordinating of home devices with the help of data obtained from various external systems and home devices in order to optimized energy usage. The proposed functions required by CSS in order to perform energy management are grouped into five main function groups: basic function, policy/goal/constraint management, service and device configuration, monitoring and diagnosis, and user interface. These functions are realized by a combination of Extended Finite State Machines (EFSMs) and a policy-based reasoning machine (see Sect. 4 for more details). Each function group performs sub-functions as shown in Fig. 3. The data maintained by CSS is stored in five different repositories based on the type of data. The details of each function group and repository are given.

**Basic function (BSC)** BSC performs normal operation and suspended operation. Normal operation refers to an operation mode where CSS performs actions based on the policies, goals, and constraints defined in the system. Suspended operation refers to an operation mode where energy management control actions are taken entirely by the consumer. This could occur from system failure or from deactivation by the consumer.

**Policy/Goal/Constraint management (PGC)** PGC maintains policies, goals, and constraints in the system. This includes the specification, update, and (re-)validation actions. (Re-)validation applies only to policies. The actions of specifying and updating policies, goals, and constraints could be initiated by the consumer through the CSS user interface or by CSS itself.

| Basic Function  | Policy/Goal/Constraint Management   | Service & Device Configuration   | Monitoring & Diagnosis  | User Interface   |
|---|---|--|---|--|
| <ul style="list-style-type: none"> <li>▪ Normal operation</li> <li>▪ Suspended operation</li> </ul> | <ul style="list-style-type: none"> <li>▪ Policy specification &amp; update</li> <li>▪ Policy (re-)validation</li> <li>▪ Constraint specification &amp; update</li> <li>▪ Goal specification &amp; update</li> </ul> | <ul style="list-style-type: none"> <li>▪ Home device (de-)registration</li> <li>▪ External service (de-)registration</li> <li>▪ Home device state and event subscription</li> <li>▪ External system data subscription</li> <li>▪ User-defined service creation &amp; update</li> </ul> | <ul style="list-style-type: none"> <li>▪ Home device state monitoring</li> <li>▪ Home device liveness monitoring</li> <li>▪ Home energy usage monitoring</li> <li>▪ External system data monitoring</li> <li>▪ Fault diagnosis</li> <li>▪ Statistics calculation</li> </ul> | <ul style="list-style-type: none"> <li>▪ Home device user interface</li> <li>▪ External system user interface</li> <li>▪ Home energy usage user interface</li> <li>▪ User-defined service configuration user interface</li> <li>▪ Policy/goal/constraint user interface</li> </ul> |

Fig. 3. CSS functions

**Service and device configuration (SDC)** SDC mainly handles the registration and deregistration of home devices and external services. It also maintains subscription of data and event from home devices and external systems. In addition, it creates and updates consumer-defined services from the consumer inputs given through the user interface.

**Monitoring and diagnosis (MON)** MON performs home device state monitoring, home device liveness monitoring, home energy usage monitoring, external system data monitoring, fault diagnosis, and statistics calculation. Home device state monitoring monitors the states of home devices. Home device liveness monitoring monitors the operating status of the device, which could at one time be normal mode, standby mode, or off mode. Home energy usage monitoring monitors energy consumption, production, and storage for the entire household and for individual home devices. External system data monitoring monitors data provided by external systems. Fault diagnosis is the analysis of faults found in Home EMS components. Statistics calculation is the collection and calculation of historical data such as electricity cost and energy consumption heuristics.

**User interface (UI)** UI is responsible for providing graphical user interfaces for the consumer to manage home devices, external system services, consumer-defined services, and policies/goals/constraints. It also displays results from consumer requests, feedbacks from executed actions, and subscribed data and event notifications. CSS must be able to render any device's and system's generic user interface based on the information of the device and system provided in the common agreed format. It may render a vendor-customized user interface if a custom template is provided. The user interface is web-based and is rendered by XSLT (<http://www.w3.org/TR/xslt>) technique.

**CSS repositories** There are five repositories in CSS: data repository (DataRep), home device repository (DevRep), service and operation repository (SOREp), policy/goal/constraint repository (PGCRep), and subscription repository (SubRep). They are accessed by EFSMs of the same name as the realized function groups. DataRep stores monitoring and statistical data and is accessible by EM and MON. DevRep stores home device manufacturing information and is accessible by SDC and MON. SORep stores information on registered home device operations and external system services and is accessible by all EFSMs. PGCRep stores policies, goals, and constraints and is accessible by EM and PGC. Finally, SubRep stores subscription information and is accessible by SDC and MON.

### 3.3 Home device functionality

A home device is any device in the house that belongs to the consumer. It can either be a energy consumer device (white goods and multimedia devices), power producer device (solar cells and wind turbines), sensor device (temperature sensor and motion sensor), or energy storage device (electric vehicles). In most cases,

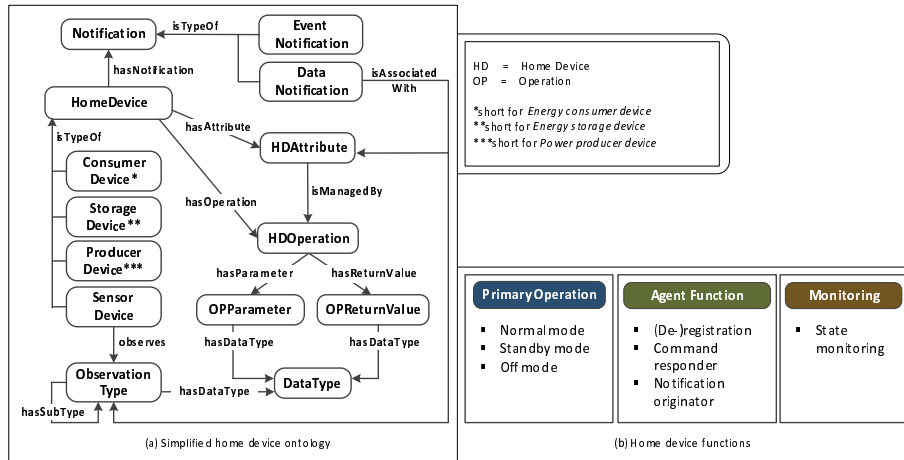


Fig. 4. Home component ontology and functions

a home device has attributes (e.g. temperature and status) and performs operations (e.g. get/set temperature, get/set status). This could be slightly different in case of a sensor. A sensor normally provides measurements on its observed environmental property. Some sensors only publish measurements to CSS hence no operations. Others might be capable of returning measurements upon request. See Fig. 4(a) for a home device ontology. The functionalities of a home device are grouped into three function groups: primary operation, agent function, and monitoring. Each function group is comprised of subfunctions as shown in Fig. 4(b). Note that for a simple device such as a sensor, some functions may not be implemented. For example, there could be only one operation mode for a sensor. In addition, a home device needs a functional component for both communicating with CSS and with itself via electrical signals to perform Primary Operation and Monitoring functions as instructed by CSS. This component is denoted as a CSS Agent and it is the entity performing Agent Function.

**Primary operation** handles home device operations in different operating modes. A device can be in normal mode where it is able to receive commands and send responses and notifications to CSS, standby mode where it is able to receive wakeup signal, or off mode. In off mode, the device is still a part of Home EMS but cannot interact with the system. It has to be turned on manually by the consumer.

**Agent function** responds to commands, and sends out subscribed notifications. It also handles the (de-)registration of the device to CSS.

**Monitoring** monitors the inherent attributes of itself, i.e. the state, and may store the state data in its internal datastore.



## 4 Home EMS energy management

### 4.1 Energy management realization

Energy management in Home EMS is the responsibility of CSS. The subfunctions performed by CSS in order to carry out energy management are realized by a combination of EFSMs and a reasoning machine (RM) as illustrated in Fig. 5. RM is a decision-making component that can be invoked by any EFSM requiring decision support, in this case  $E_{BSC}$  and  $E_{MON}$ . The logic to reach the decisions normally comes from extensive research which tends to change due to new researched solutions or the emergence of new variables. To cope with this issue, the energy management logic should be contained and formalized in a form that facilitates changes. For this matter, policy-based reasoning provides the flexibility that allows new rules to be specified without re-compilation [11]. Thus, the term *reasoning* in the reasoning machine refers to policy-based reasoning.

RM interprets policy specifications and goal specifications. A policy is a set of rules and actions. Rules are defined by variables and constraints. Variables are relevant data required in order to select actions. There are two types of variables: decision variable and dynamic variable. A decision variable is what needs to be determined by RM, e.g. the start time of a device. A dynamic variable is data that changes as a function of time and may be beyond the control scope of RM, e.g. energy and grid prices. Dynamic variables influence the values of decision variables. For example, grid price (dynamic) during certain time of day is high so the start time (decision) could be delayed to reduce cost. Constraints are imposed on decision variables which, consequently, restrict the possible actions. Each action has a cost. Goals are associated with policies and define the targets to achieve. The goal specification and costs of actions determine the selection

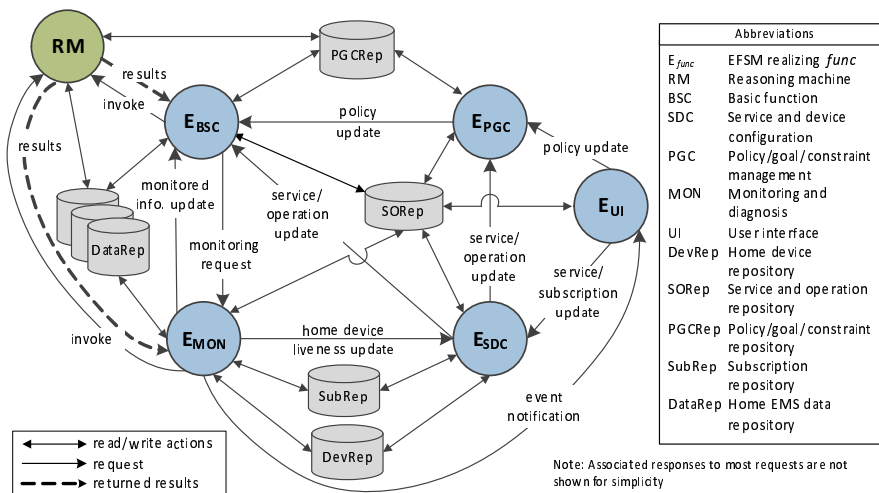


Fig. 5. ESFM and RM interaction diagram

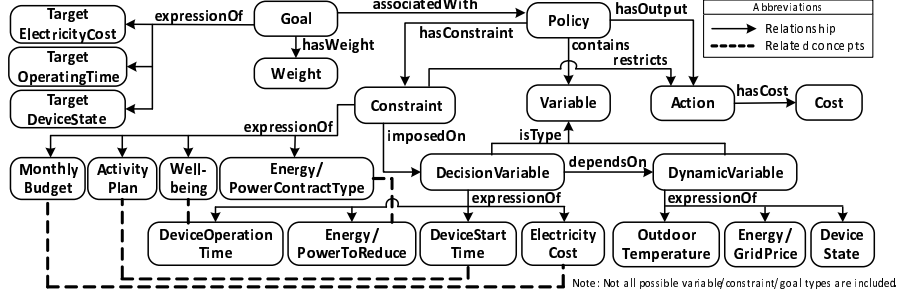


Fig. 6. Policy and goal ontology model

of actions when more than one action set is applicable. The decided actions, which are mainly operating instructions, will be executed on home devices. Fig. 6 depicts an ontology model for all related concepts. For readability, the example constraints are high-level concepts that can be formulated by decision variables.

Policies are ideally created and updated by the CSS vendor. The constraint values are set by the consumer. Apart from the policies themselves, statistics related to energy consumption heuristics of home devices can be used to adjust some estimates that will improve the decision making by RM.

#### 4.2 Policy example: optimized laundry

One policy example that operates a washing machine (*wm*) at the optimal time and further suggests the method and time to dry clothes (electric dryer (*ed*) or outdoor drying (*od*)) is described. The maximum delay between activities is one hour. The goal is to minimize the electricity cost. The constraint is the latest finish time. Two policies are described in Table 2. Dynamic variables are specified explicitly. Decision variables are the start time, dry time, and cost. The first policy describes a case when *od* is selected while the second policy describes a case when *ed* is selected. For both cases, the optimal times to start *wm*, *od*, and *ed* are selected based on the weather conditions retrieved from an external weather service and electricity prices retrieved from DSO and ESP which are used to calculate the end time and the electricity cost.

Let:

|                      |            |   |
|----------------------|------------|---|
| $t_c$                | represents | current hour                                    |
| $t_{[device],start}$ | represents | start time of the <i>device</i>                 |
| $t_{[device],end}$   | represents | end time of the <i>device</i>                   |
| $t_{latest}$         | represents | latest finish time                              |
| $t_{wash}$           | represents | washing duration (hour)                         |
| $t_{edry}$           | represents | drying duration by electric dryer (hour)        |
| $w_{[device]}$       | represents | power consumption by <i>device</i> (kW)         |
| $weather(t_1, t_2)$  | represents | a set of weather conditions from $t_1$ to $t_2$ |
| $prices(t_1, t_2)$   | represents | a set of electricity prices from $t_1$ to $t_2$ |
| $p(t_1, t_2)$        | represents | average electricity price from $t_1$ to $t_2$   |

**Table 2.** Policy example for optimized laundry in formal notations

| # | Policy specifications   |
|---|---|
| 1 | <p><b>dynamic variables:</b> <math>weather(t_c, t_{latest})</math> and <math>prices(t_c, t_{latest})</math></p> <p><b>constraints:</b> <math>t_{od,start} + drytime(t_{od,start}) \leq t_{latest}</math></p> <p><b>actions:</b> start <math>wm</math> at <math>t_{wm,start}</math> and start <math>od</math> at <math>t_{od,start}</math> such that <math>0 \leq t_{od,start} - (t_{wm,start} + t_{wash}) \leq 1h</math> and <math>t_{od,start} + drytime(t_{od,start}) \leq t_{latest}</math> and <math>c(t_{wm,start}) &lt; c(t) \quad \forall t \in T - t_{wm,start}, T = \{t_c, t_c + 1, \dots, t_{latest}\}</math></p> <p><b>cost:</b> <math>c(t) = w_{wm} \times t_{wash} \times p(t, t + t_{wash})</math></p>  |
| 2 | <p><b>dynamic variables:</b> <math>weather(t_c, t_{latest})</math> and <math>prices(t_c, t_{latest})</math></p> <p><b>constraints:</b> <math>t_{ed,start} + t_{edry} \leq t_{latest}</math></p> <p><b>actions:</b> start <math>wm</math> at <math>t_{wm,start}</math> and start <math>ed</math> at <math>t_{ed,start}</math> such that <math>0 \leq t_{ed,start} - (t_{wm,start} + t_{wash}) \leq 1h</math> and <math>t_{ed,start} + t_{edry} \leq t_{latest}</math> and <math>c(t_{wm,start}, t_{ed,start}) &lt; c(t_1, t_2) \quad \forall (t_1 \in T - t_{wm,start}, t_2 \in T - t_{ed,start}), T = \{t_c, t_c + 1, \dots, t_{latest}\}</math></p> <p><b>cost:</b> <math>c(t_1, t_2) = (w_{wm} \times t_{wash} \times p(t_1, t_1 + t_{wash})) + (w_{ed} \times t_{edry} \times p(t_2, t_2 + t_{edry}))</math></p> |

And  $drytime(t)$  is the estimated outdoor drying duration in hours which is calculated as a function of the weather conditions from time  $t$  to  $t_{latest}$  (see (1)). The estimation is based on the heuristics related to the temperature, wind, and humidity. Infinite duration means outdoor drying is not possible.

$$drytime(t) = f(weather(t, t_{latest})) \in \{1, 2, \dots, \infty\} \quad (1)$$

## 5 Home EMS inter-working framework

An inter-working framework is here defined as an application layer framework that governs how CSS, home devices, and external systems communicate with one another. Two things are defined: 1) a common and formal format for operation and service description and 2) operation and service invocation.

Operations and services provided by home devices and external systems need to be defined in a common agreed format to be accessible and executable by CSS. The required elements to be defined are ID, name, and description, with the optional addition of parameters and return values. We propose the use of RDF/XML as a language for operation and service definitions as well as home device and external system information. Resource Description Framework (RDF) [12] is a semantic web language for describing resources on the web. There are four reasons for choosing RDF: 1) it can be expected that full ontology models for appliances will be standardized 2) RDF is good at modeling relationships in a meaningful way which can be used by CSS to make decisions 3) RDF is suggested as a data exchange format to be used by power utilities in IEC61970-501 [4] hence simpler integration with DSO and ESPs. 4) Semantic Sensor Network Ontology [13] coupled with Sensor Web Enablement [3] can be used for the description and measurement data of sensors. Fig. 7(a) shows a simplified device information file of a washing machine. RDF terms used are

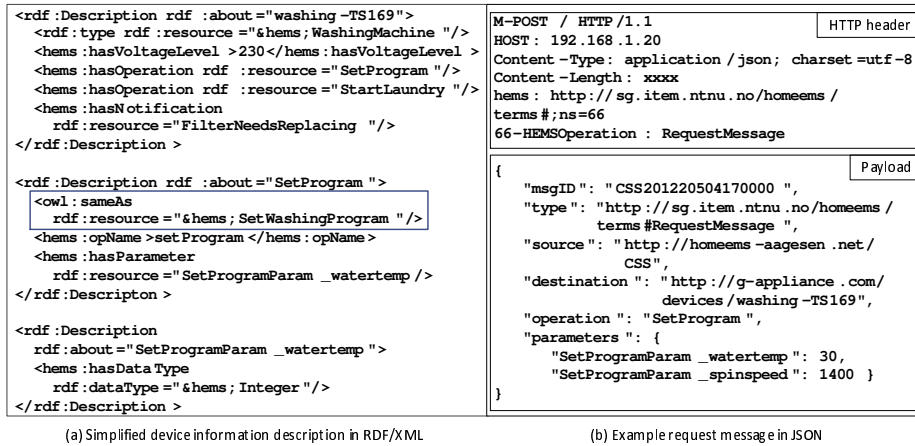


Fig. 7. Device information description and request message

in accordance to the ontology depicted in Fig. 4. The highlighted line shows how a standardized operation name can be specified. This line can be added at any time and the vendor only needs to provide the updated information file which may be pushed to CSS. What this really means is device vendors and CSS vendors can manufacture their products separately without being concerned on naming protocols because declaring that one operation is the same as the other only requires one line. This device information file contains operation descriptions along with other device information. It must be provided by a home device to CSS upon registration. Services exposed by external systems are defined in the same way but the registration is often a manual process by the vendor of the CSS at manufacture time. This is because existing web services use different methods for service execution and thus have to be handled differently. In this regard, service descriptions for external services exist only to enable consistent processing and semantic classification of available services.

Operation and service invocation between CSS and home devices or CSS and external systems is done through message exchange. There are three types of messages: request, response, and notification. All messages are included as HTTP payloads. The format of the message is JavaScript Object Notation (JSON) (<http://www.json.org>). JSON was chosen as opposed to XML-based languages because the processing needs for production is low and thus can be constructed by most devices including microcontrollers. However, any reference to device or system in a JSON-encoded message needs to be the same URI as defined in the RDF/XML device or system information file. See Fig. 7(b) for an example request message in JSON. Note that this inter-working framework only specifies the interface for the communication between Home EMS components. The translation and conversion of the received messages into electrical signals for the actual control of home devices is a necessary and inevitable process that must be done by the device vendors.

## 6 Summary and conclusion

Residential consumers contribute to a substantial amount of electricity consumption. Energy consumption pattern can affect the stability of the grid. Home Energy Management System (EMS) can assist household inhabitants in operating home devices to optimize energy usage. This paper proposes an architecture model and a functional framework for Home EMS with focus on simplicity and scalability to enable the system's rapid deployment. Energy management is carried out by a central system whose functions are realized by a combination of EFSMs and a policy-based reasoning machine. Our proposal to use policy-based reasoning offers an advantage to cope with changes in variables and simplifies incorporation of new energy management solutions as new policies. In addition, a Home EMS inter-working framework based on RDF and JSON was also proposed. Data representation with RDF provides the necessary semantics and enables harmonization with existing standards while JSON messages can be constructed by most microcontrollers. We are currently working on the construction of more energy management policies to cover broader aspects including indoor temperature adjustment and load scheduling involving more devices. A prototype system based on the proposed functional and inter-working frameworks will be implemented. The prototype will also be used as a platform to evaluate the effectiveness of the constructed policies with respect to electricity cost saved by consumers and the reduction of peak loads.

## References

1. Li, F., Qiao, W., Sun, H., Wan, H., Wang, J., Xig, Y., Xu, Z., Zhang, P.: Smart transmission grid: Vision and framework. *IEEE Transactions on Smart Grid* 1(2) (2010) 168–177
2. European Commission: Energy efficiency status report 2012 EUR 25405 EN. (2012)
3. Open Geospatial Consortium: Sensor Web Enablement, <http://www.opengeospatial.org/projects/groups/sensorwebdwg>
4. International Electrotechnical Committee: IEC 61970-501 ed. 1 Energy Management System Application Program Interface (EMS-API) - part 501: Common Information Model Resource Description Framework (CIM RDF) Schema. (2006)
5. Belkin International, Inc: Belkin WeMo Home, <http://www.belkin.com/uk/c/WSWH>
6. X10: Home automation solutions, <http://www.x10.com/automation/>
7. eGauge Systems LLC: eGauge overview, <http://www.egauge.net/overview.php>
8. AIM: A novel architecture for modelling, virtualizing and managing the energy consumption of household appliances, <http://www.ict-aim.eu/home.html>
9. Beywatch: Beywatch Home, <http://www.beywatch.eu/index.php>
10. SmartHouse-SmartGrid Consortium: Smart House/Smart Grid, <http://www.smarthouse-smartgrid.eu>
11. Aagesen, F.A., Thongtra, P.: On Capability-related Adaptation in Networked Service Systems. *Int. Journal of Computer Networks and Communications* 4(4) (2012)
12. World Wide Web Consortium: Resource Description Framework (RDF), <http://www.w3.org/RDF/>
13. W3C Incubator Group: Semantic Sensor Network XG Final Report, Available at <http://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628>