European Commission

Study on Semantic Assets for Smart Appliances Interoperability

D-S4: FINAL REPORT

A study prepared for the European Commission
DG Communications Networks, Content & Technology
by:

TNO innovation for life

Digital Agenda for Europe

This study was carried out for the European Commission by

TNO innovation for life

Authors:

Laura Daniele

laura.daniele@tno.nl

Frank den Hartog

frank.denhartog@tno.nl

Jasper Roes (Project manager)

jasper.roes@tno.nl

Internal identification

Contract number: 30-CE-0610154/00-11

SMART number: 2013/01077

DISCLAIMER

By the European Commission, Directorate-General of Communications Networks, Content & Technology.

The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the Commission. The Commission does not guarantee the accuracy of the data included in this study. Neither the Commission nor any person acting on the Commission's behalf may be held responsible for the use which may be made of the information contained therein.

Version	Date	Comments
0.9	9 February 2015	For Expert Group review
1.0	11 March 2015	For public release

Abstract

About two thirds of the energy consumed in buildings originates from household appliances. Nowadays, appliances are often intelligent and networked devices that form complete energy consuming, producing, and managing systems. Reducing energy consumption is therefore a matter of managing and optimizing the energy utilization on a system level. These systems need standardized interfaces on a sensor and device level. Many of the required standards already exist, but a common architecture does not, resulting in a market which is too fragmented and powerless. We have designed a reference ontology for these appliances, and this deliverable presents the results of the study. We have identified 47 different semantic assets that describe various properties of smart appliances in residential environments. We translated 23 of the 47 semantic assets into ontologies in the Web Ontology Language (OWL). In the 23 ontologies created we identified 20 recurring concepts that we used as initial building blocks for the Smart Appliances Reference (SAREF) ontology. We mapped the recurring concepts from the assets to the SAREF ontology to allow for translations between the 23 ontologies that we created. Next to that we showed how the SAREF ontology can be applied in relation to the ETSI M2M Architecture.

Abstract – French

Environ deux tiers de la consommation d'énergie à l'intérieur des bâtiments est à imputer aux appareils domestiques. De nos jours, les appareils sont souvent des entités intelligentes et en réseau, formant des systèmes complets de production, de gestion et de consommation d'énergie. Réduire consommation d'énergie est par conséquent une question de gérer et d'optimaliser l'utilisation de l'énergie au niveau des systèmes. Afin de rendre des extensions possibles, ces systèmes nécessitent des interfaces standardisés au niveau senseur et appareil. Beaucoup de ces standards nécessaires existent déjà, mais il manque une architecture commune, ce qui résulte en un marché trop fragmenté et sans réel pouvoir. Nous avons élaboré une ontologie référence pour ces appareils, et ce rapport présente les résultats de notre étude. Nous avons identifié 47 éléments sémantiques différents qui décrivent les propriétés variées des appareils intelligents en environnement résidentiel. Nous avons traduit 23 de ces 47 éléments en ontologies web (Web Ontology Language, OWL). Parmi les 23 ontologies ainsi créées, nous avons identifié 20 concepts récurrents, concepts initiaux que nous avons ensuite assemblé pour former l'ontologie référence des appareils intelligents (Smart Appliances Reference, SAREF. Nous avons ensuite relié les concepts des caractéristiques à l'ontologie SAREF afin de permettre des traductions entre les ontologies. En sus, nous avons montré de quelle manière l'ontologie SAREF peut être appliquée, en relation avec l'architecture ETSI M2M.

Executive summary

About two thirds of the energy consumed by buildings originates from the residential sectors and thus household appliances. Household appliances or home appliances are electrical/mechanical machines which accomplish some household functions. Nowadays, appliances are not stand-alone systems anymore. They are often highly intelligent ("smart") and networked devices, that form complete energy consuming, producing, and managing systems. Reducing the use of energy and production of greenhouse gasses is therefore not only a matter of increasing the efficiency of the individual devices, but also managing and optimizing the energy utilization on a system level. The systems will therefore inevitably consist of devices and sensors from different vendors, and open interfaces enabling further extensions. The interfaces need to be properly standardized and offer external access on a semantic level both to any manageable and controllable function of the system as a whole, and to any device that is part of the system.

However, the problem is not the lack of available standards. Actually, there already exist many standards, all dealing with a smaller or larger part of the problem, sometimes overlapping and competing. Various workshops and projects already explored this field and concluded that defining a useful and applicable reference data model should in principle be possible. One reference ontology of consensus could be created to cover the needs of all appliances relevant for energy efficiency, and it can be expanded to cover future intelligence requirements. The European Commission therefore issued a tender for a Study on "Available Semantics Assets for the Interoperability of Smart Appliances: Mapping into a Common Ontology as a M2M Application Layer Semantics", defining 3 tasks:

- **Task 1:** Take stock of existing semantic assets and use case assets
- **Task 2:** Perform a translation exercise of each model (or use case) to a common ontology language and a mapping or matching exercise between all the models
- **Task 3:** Propose a reference ontology and document the ontology into the ETSI M2M architecture

TNO was invited to perform this study. In Task 1 we took stock of existing semantic assets and use case assets, described their semantic coverage, and presented an initial semantic mapping. The term asset is used in this study in a broad sense, since it refers to a source that can present a project, a set of documents, an ISO standard, a working group, a committee, a paper, a homepage (of a wiki, or of any other website) that is somehow related to energy management and/or home appliances. Therefore, an asset may refer to one well-defined single ontology, but in most cases is a pointer to a set of multiple documents, several related standards and distinct articles on a web site or wiki, from which a single ontology should be derived. The Invitation to Tender already listed 29 assets. We have identified 18 more assets that needed to be included in our study given the scope as set out by the European Commission. Of these in total 47 assets we were able to short-list 23 assets which provide a good basis for further reference ontology development. The short list was composed solely based on how well the asset is covering the scope of the project and if the asset provided concrete semantic specifications, preferably in the form of XML or OWL files.

These assets have been described in terms of their:

a) Model Acronym and Full Name
b) Most relevant URL and other precise references
c) Overall description
d) Description of the semantic coverage
e) Overall description of the consensus driven process leading to the model

The other assets included have been described more briefly. The work resulted in the following short list:

- ECHONET
- FIEMSER
- UPnP
- SmartCoDE
- OMA Lightweight M2M
- SEP2
- EnOcean
- OMS
- Hydra
- KNX
- W3C SSN
- OSGi DAL
- eDIANA
- FAN
- DECT ULE
- Z-Wave
- SEEMPubs
- PowerOnt (previously SEIPF)
- FIPA
- Mirabel

In order to support the stocktaking task with an overall representation that could help the reader in visualizing the key terms used by different assets, we created a visual representation, which schematically depicts between 10 and 15 key terms for each asset. This visualization was intended to be an initial step towards the definition of a common semantics for the smart appliances domain. Although the considered assets were heterogeneous when considering their coverage we were able, using this key term visualization, to identify three main trends with a focus on 1) *devices*, *sensors* and their specification in terms of *services*, *functions* and *states*, 2) *energy* consumption information and profiles to optimize energy efficiency, and 3) *buildings* related semantic models.

The results of task 1 "take stock of existing semantic assets and use case assets" were presented in the first of a series of deliverables, namely the D-S1 Interim Study Report. D-S1 was first reviewed by the project's Expert Group, and later on discussed in the 1st stakeholders' workshop that took place in Brussels on May 27/28, 2014.

In task 2 we have translated the assets in the short list to corresponding OWL ontologies and we have created initial mappings among these ontologies. The purpose of these mappings was to relate the 20 assets using their most recurring concepts (core concepts), as initially identified in task 1.

Table 1 lists, in alphabetical order, the OWL ontologies created in task 2. The results of task 2 were described in the D-S2 Second Interim study report, in which for each ontology we provided a *title*, the *source* document used as a main reference to create the ontology, a *description* of the main classes and properties, and, eventually, *observations* necessary to better understand the choices underlying the ontology design and suggestions for its future extension. Analogously to D-S1, D-S2 was also first reviewed by the project's Expert Group, and later on discussed in the 2nd stakeholders' workshop that took place at the ETSI premises in Sophia Antipolis on October 15, 2014.

Table 1. Ontologies described in this study

Acronym	Source	URL
DECT ULE	'HF-Protocol', 'HF-Service', 'HF-Interface', 'HF-Profile', version1.0, 23 January 2014	https://sites.google.com/site/smartappliancesproject/ontologies/dect_ule-ontology
ECHONET	ECHONET Specifications Appendix 'Detailed Requirements for ECHONET Device Objects' Release C, 31 May 2013	https://sites.google.com/site/smartappliancesproject/ontologies/echonet-ontology
eDIANA	'D2.2-A Ontology for Device Awareness', 30 November 2009	https://sites.google.com/site/smartappliancesproject/ontologies/ediana-ontology
EnOcean	'EnOcean Equipment Profiles (EEP)', Version 2.6, 17 December 2013	https://sites.google.com/site/smartappliancesproject/ontologies/enocean-ontology
FAN	'Interface description: Interface report' , Version 1.0 (final), 7th January, 2014	https://sites.google.com/site/smartappliancesproject/ontologies/fan-ontology
FIEMSER	'D5 FIEMSER Data Model', February 2011	https://sites.google.com/site/smartappliancesproject/ontologies/fiemser-ontology
FIPA	'FIPA Device Ontology Specification', document number SC00091E , 3 December 2002	https://sites.google.com/site/smartappliancesproject/ontologies/fipa-ontology
HYDRA	'Deliverable D6.6 Updated MDA Design Document', version 1.0, 20 August 2009	https://sites.google.com/site/smartappliancesproject/ontologies/hydra-ontology
KNX	• 'KNX System Specifications Interworking Datapoint Types', Version 01.09.01, 18 September 2014 • 'KNX Advanced Course-Interworking_E1209b'	https://sites.google.com/site/smartappliancesproject/ontologies/knx-ontology
MIRABEL	'D7.5 MIRABEL-ONE: Initial draft of the MIRABEL Standard', version1.0, 22 December 2011	https://sites.google.com/site/smartappliancesproject/ontologies/mirabel-ontology
OMA LW M2M	'OMA Lightweight Machine-to-Machine Technical Specification', candidate version 1.0, 10 December 2013	https://sites.google.com/site/smartappliancesproject/ontologies/oma-lightweight_m2m-ontology
OMS	'Open Metering System Specification Vol.2 – Primary Communication Issue 4.0.2', 27 January 2014	https://sites.google.com/site/smartappliancesproject/ontologies/oms-ontology
OSGi DAL	'RFC-196 OSGi Alliance Device Abstraction Layer, Draft', 30 January 2014	https://sites.google.com/site/smartappliancesproject/ontologies/osgi_dal-ontology
PowerOnt	• Politecnico di Torino, e-Lite research group webpage (http://elite.polito.it/dogont) • D. Bonino, F. Corno, 'DogOnt - Ontology Modeling for Intelligent Domotic Environments'.	https://sites.google.com/site/smartappliancesproject/ontologies/dogpower-ontology
SEEMPubs	'Deliverable D5.1, Data Format Definition', version 1.0, 30 September 2012	https://sites.google.com/site/smartappliancesproject/ontologies/seempubs-ontology

SEP2	'Zigbee Alliance/HomePlug Alliance Smart Energy Profile 2 Application Protocol Standard, ZigBee Public Document 13-0200-00, April 2013'	https://sites.google.com/site/smartappliancesproject/ontologies/sep2-ontology
SmartCoDE	'Deliverable D1.1.2 -Model of local energy resource cluster', 31 December 2012	https://sites.google.com/site/smartappliancesproject/ontologies/smartcode-ontology
UPnP	'UPnP Device Architecture 1.1', 15 October 2008	https://sites.google.com/site/smartappliancesproject/ontologies/upnp-ontology
W3C SSN	W3C Semantic Sensor Network Incubator Group webpage (http://www.w3.org/2005/Incubator/ssn/ssnx/ssn)	https://sites.google.com/site/smartappliancesproject/ontologies/w3c_ssn-ontology
Z-Wave	'Z-Wave Technical Basics - Chapter 4: Application Layer', 1 June 2011	https://sites.google.com/site/smartappliancesproject/ontologies/z-wave-ontology

These ontologies were also made available online at the smart appliances website[1]. This website provides a page for each of the 20 semantic assets in the short list, with the URL to download the corresponding ontology and a human-readable explanation to describe its main classes and properties. Each page also includes a tab for posting comments (available when logged on to the website with a Google-account) to allow the “owners” of the corresponding assets to validate whether the meaning they originally intended for their assets is actually reflected in our ontologies. These ontologies were for us a means to create the reference ontology, but they were not the final result of this project. Therefore, changes or extensions on these ontologies, together with new mappings to the reference ontology that may emerge in the future, should be realized by the interested stakeholders once the project is ended.

Based on interaction with the stakeholders during the second workshop we added the follow assets to the short list after task 2 was performed.

- ZigbeeHA
- CENELEC EN50491
- Adapt4EE

In task 3 we created a first version of the Smart Appliances REFerence (SAREF) ontology that explicitly specifies the recurring core concepts in the smart appliances domain, the main relationships between these concepts, and axioms to constrain the usage of these concepts and relationships. SAREF is based on the fundamental principles of r*euse and alignment* of concepts and relationships that are defined in existing assets, *modularity* to allow separation and recombination of different parts of the ontology depending on specific needs, e*xtensibility* to allow further growth of the ontology, and *maintainability* to facilitate the process of identifying and correcting defects, accommodate new requirements, and cope with changes in (parts of) the SAREF ontology. We subsequently mapped SAREF on the ETSI M2M Architecture, and found that there is a good correlation between the ETSI M2M Architecture and SAREF’s function-related device categories. The correlation with energy-related and building-related device categories is still minimal. The results of

1 https://sites.google.com/site/smartappliancesproject/ontologies

task 3 were presented in the *D-S3 Third Interim Study Report*. D-S3 was also reviewed by the project's Expert Group, and later on discussed in the 3rd stakeholders' workshop that took place in Brussels on February 10, 2015.

The final result of the smart appliances study carried out by TNO is the *D-S4 Final Study report* , which consists of the three interim deliverables produced during the study, updated with the feedback collected from the stakeholders and the expert group. D-S4 was officially presented at the 4th stakeholders' workshop that took place on the 1st April 2015 in Brussels. D-S4 will be officially passed to ETSI Smart M2M for further development into, as is currently foreseen, a Technical Specification. After the end of the project, it is up to the industry to maintain and extend the reference ontology as needed. We recommend that this process is supported by ETSI. The Horizon 2020 call ICT 30 is a perfect way of supporting research and development activities.

The Smart Appliances REFerence (SAREF) ontology is available online at http://ontology.tno.nl/saref. The Turtle version of the SAREF ontology can be downloaded at http://ontology.tno.nl/saref.ttl, and can be opened with any ontology editor, such as TopBraid Composer, Protégé and NeOn.

Executive summary – French

Environ deux tiers de la consommation d'énergie à l'intérieur des bâtiments est à imputer aux appareils domestiques. Les appareils domestiques ou ménagers sont des machines électriques/mécaniques réalisant des fonctions ménagères. De nos jours, les appareils ne sont plus des systèmes indépendants. Ce sont souvent des entités intelligentes et en réseau, formant des systèmes complets de production, de gestion et de consommation d'énergie. Réduire la consommation d'énergie et la production de gaz à effet de serre est par conséquent non seulement une question d'augmenter l'efficacité de chaque appareil, mais aussi de gérer et d'optimaliser l'utilisation de l'énergie au niveau des systèmes. Les systèmes seront par conséquent inévitablement constitués d'appareils et de senseurs provenant de différents vendeurs, et d'interfaces ouverts permettant les extensions. Les interfaces doivent être correctement standardisés et offrir des accès externes au niveau sémantique, non seulement concernant toute fonction gérable et contrôlable du système dans son ensemble, mais aussi de tout appareil en tant qu'élément du système.

Cependant, le problème n'est pas le manque de standards disponibles. Il existe en fait beaucoup de standards, tous en relation avec une petite ou une grande part du problème, parfois même en doublon ou concurrent. Des workshops et projets variés ont déjà exploré ce domaine et ont conclu que définir un modèle de données de référence utile et applicable devrait être en principe possible. Le consensus pourrait être créé autour d'une ontologie de référence qui couvrirait les besoins des appareils concernant l'efficacité énergétique et qui pourrait être étendue à des exigences d'intelligence futures. La Commission Européenne a par conséquent lancé une étude ayant pour thème "Eléments sémantiques pour l'interopérabilité des appareils intelligents disponibles: intégration dans une ontologie commune comme sémantique de la couche Application machine-à-machine', étude qui définit trois activités:

- **Activité 1:** Faire le point sur les éléments de sémantiques existants et les éléments de cas utilisateurs.
- **Activité 2:** Mener un exercice de traduction de chaque modèle (ou cas utilisateur) en une langue commune d'ontologie et un exercice de combinaison ou comparaison de tous les modèles.
- **Activité 3:** Proposer une ontologie référente et documenter cette ontologie dans le cadre de l'architecture ETSI M2M.

TNO a été invité à réaliser cette étude. Dans le cadre de la première activité, nous avons fait le point sur les éléments de sémantique et de cas utilisateurs existants, décrivant leur couverture sémantique et présentant une vue d'ensemble sémantique initiale. Le terme 'élément' est utilisé au sens large dans cette étude, incluant une source présentant un projet, un ensemble de documents, un standard ISO, un groupe de travail, un comité, un article, une page web (ou un wiki, ou tout autre site), tout élément qui est relié d'une manière ou d'une autre au thème de gestion énergétique et/ou aux appareils domestiques. Par conséquent, un élément peut faire référence à une ontologie bien définie unique, mais dans la majorité des cas, c'est l'index d'un ensemble de documents multiples, plusieurs standards apparentés et des articles distincts sur un site Internet ou Wiki, dont une ontologie unique devrait être dérivée. L'appel d'offre dressait déjà une liste de 29 éléments. Nous en avons identifié 18 supplémentaires qui devaient être inclus dans notre étude, vu le cadre de l'étude tel que défini par la Commission Européenne. De ce total de 47 éléments, nous avons pu sélectionner

23 éléments fournissant une bonne base pour le développement futur d'une ontologie de référence. Les éléments présents sur la liste ont été sélectionnés essentiellement par rapport à leur couverture du cadre du projet, et si l'élément en question présentait un niveau suffisamment concret de spécifications sémantiques, préférablement sous la forme de fichiers XML ou OWL.

Ces éléments ont été répertoriés utilisant leurs:

a) Acronyme modèle et nom complet
b) URL les plus pertinents et autres références précises
c) Description générale
d) Description de la couverture sémantique
e) Description générale du processus de recherche de consensus menant au modèle

Les autres éléments inclus ont été décrits de manière plus succincte. Ceci a résulté dans la courte liste suivante:

- ECHONET
- FIEMSER
- UPnP
- SmartCoDE
- OMA Lightweight M2M
- SEP2
- EnOcean
- OMS
- Hydra
- KNX
- W3C SSN
- OSGi DAL
- eDIANA
- FAN
- DECT ULE
- Z-Wave
- SEEMPubs
- PowerOnt (previously SEIPF)
- FIPA
- Mirabel

Pour renforcer l'activité d'inventaire par une représentation générale afin d'aider le lecteur à visualiser les termes clé utilisés par les différents éléments, nous avons créé un représentation visuelle, qui décrit de manière schématique entre 10 à 15 termes clé pour chaque élément. Cette visualisation était destinée à constituer un premier pas vers la définition d'une sémantique commune pour le domaine des appareils intelligents. Bien que les éléments considérés fussent hétérogènes si l'on considère leur couverture, le fait d'utiliser cette visualisation des termes clé nous a permis d'aboutir à l'identification de trois tendances principales autour de trois axes, 1) *appareils, senseurs* et leur spécification en termes de *services, fonctions et états*, 2) information relative à la *consommation d'énergie et profils permettant l'optimalisation de l'efficience énergétique,* 3) modèles sémantiques relatifs aux *bâtiments*.

Les résultats de la première activité "Faire le point sur les éléments de sémantiques existants et les éléments de cas utilisateurs" ont été présentés dans le premier d'une série de rapports, le rapport intermédiaire D-S1. D-S1 a été révisé par le groupe Expert du projet, et plus tard discuté lors du premier workshop rassemblant les parties intéressées, workshop qui s'est tenu les 27 et 28 mai 2014 à Bruxelles.

Dans le cadre de la deuxième activité, nous avons traduit les éléments sélectionnés en ontologies OWL correspondantes et nous avons créé des cartographies initiales entre ces ontologies. Le but de ces cartographies était de relier les 20 éléments en utilisant les concepts les plus récurrents (concepts centraux), comme initialement identifié dans le cadre de la première activité.

La Table 1 présente, par ordre alphabétique, les ontologies OWL créées dans le cadre de l'activité 2. Les résultats de l'activité 2 ont été décrits dans un deuxième rapport d'étude intermédiaire (D-S2), rapport dans lequel nous fournissons pour chaque ontologie, un *titre*, le *document* source utilisé comme principale référence pour créer l'ontologie, une d*escription* des principales classes et propriétés, et, éventuellement, les *observations* nécessaires à une meilleure compréhension des choix sous-jacents au design de l'ontologie, ainsi que des suggestions pour une future extension. De la même manière que pour D-S1, D-S2 a été tout d'abord révisé par le groupe expert du projet, puis discuté plus tard durant le deuxième workshop des parties intéressées, qui eut lieu dans les murs d'ETSI, à Sophia Antipolis, le 15 octobre 2014.

Table 2. Ontologies décrites dans cette étude

Acronyme	Source	URL
DECT ULE	'HF-Protocol', 'HF-Service', 'HF-Interface', 'HF-Profile', version1.0, 23 Janvier 2014	https://sites.google.com/site/smartappliancesproject/ontologies/dect_ule-ontology
ECHONET	ECHONET Specifications Appendix 'Detailed Requirements for ECHONET Device Objects' Release C, 31 Mai 2013	https://sites.google.com/site/smartappliancesproject/ontologies/echonet-ontology
eDIANA	'D2.2-A Ontology for Device Awareness', 30 Novembre 2009	https://sites.google.com/site/smartappliancesproject/ontologies/ediana-ontology
EnOcean	'EnOcean Equipment Profiles (EEP)', Version 2.6, 17 Décembre 2013	https://sites.google.com/site/smartappliancesproject/ontologies/enocean-ontology
FAN	'Interface description: Interface report' , Version 1.0 (final), 7th Janvier, 2014	https://sites.google.com/site/smartappliancesproject/ontologies/fan-ontology
FIEMSER	'D5 FIEMSER Data Model', Février 2011	https://sites.google.com/site/smartappliancesproject/ontologies/fiemser-ontology
FIPA	'FIPA Device Ontology Specification', document number SC00091E , 3 Décembre 2002	https://sites.google.com/site/smartappliancesproject/ontologies/fipa-ontology
HYDRA	'Deliverable D6.6 Updated MDA Design Document', version 1.0, 20 Août 2009	https://sites.google.com/site/smartappliancesproject/ontologies/hydra-ontology
KNX	• 'KNX System Specifications Interworking Datapoint Types', Version 01.09.01, 18 Septembre 2014 • 'KNX Advanced Course-Interworking_E1209b'	https://sites.google.com/site/smartappliancesproject/ontologies/knx-ontology
MIRABEL	'D7.5 MIRABEL-ONE: Initial draft of the MIRABEL Standard', version1.0, 22 Décembre 2011	https://sites.google.com/site/smartappliancesproject/ontologies/mirabel-ontology
OMA LW M2M	'OMA Lightweight Machine-to-Machine Technical Specification', candidate version 1.0, 10 Décembre 2013	https://sites.google.com/site/smartappliancesproject/ontologies/oma-lightweight_m2m-ontology

OMS	'Open Metering System Specification Vol.2 – Primary Communication Issue 4.0.2', 27 Janvier 2014	https://sites.google.com/site/smartappliancesproject/ontologies/oms-ontology
OSGi DAL	'RFC-196 OSGi Alliance Device Abstraction Layer, Draft', 30 Janvier 2014	https://sites.google.com/site/smartappliancesproject/ontologies/osgi_dal-ontology
PowerOnt	• Politecnico di Torino, e-Lite research group webpage (http://elite.polito.it/dogont) • D. Bonino, F. Corno, 'DogOnt - Ontology Modeling for Intelligent Domotic Environments'.	https://sites.google.com/site/smartappliancesproject/ontologies/dogpower-ontology
SEEMPubs	'Deliverable D5.1, Data Format Definition', version 1.0, 30 Septembre 2012	https://sites.google.com/site/smartappliancesproject/ontologies/seempubs-ontology
SEP2	'Zigbee Alliance/HomePlug Alliance Smart Energy Profile 2 Application Protocol Standard, ZigBee Public Document 13-0200-00, Avril 2013'	https://sites.google.com/site/smartappliancesproject/ontologies/sep2-ontology
SmartCoDE	'Deliverable D1.1.2 -Model of local energy resource cluster', 31 Décembre 2012	https://sites.google.com/site/smartappliancesproject/ontologies/smartcode-ontology
UPnP	'UPnP Device Architecture 1.1', 15 Octobre 2008	https://sites.google.com/site/smartappliancesproject/ontologies/upnp-ontology
W3C SSN	W3C Semantic Sensor Network Incubator Group webpage (http://www.w3.org/2005/Incubator/ssn/ssnx/ssn)	https://sites.google.com/site/smartappliancesproject/ontologies/w3c_ssn-ontology
Z-Wave	'Z-Wave Technical Basics - Chapter 4: Application Layer', 1 Juin 2011	https://sites.google.com/site/smartappliancesproject/ontologies/z-wave-ontology

Ces ontologies ont été aussi mises à disposition en ligne sur le site des appareils intelligents[2]. Ce site Internet présente une page pour chacun des 20 éléments sémantiques sélectionnés, avec les URL permettant de télécharger l'ontologie correspondante et une explication lisible pour l'humain décrivant les principales classes et propriétés. Chaque page donne aussi la possibilité de laisser des commentaires (disponibles lorsque connecté au site par un compte Google), ce qui permet aux 'propriétaires' des éléments correspondants de vérifier si le sens qu'ils voulaient à l'origine donner à leur élément est effectivement reflété par nos ontologies. Ces ontologies étaient pour nous un moyen de créer l'ontologie de référence, mais ne constituaient pas le résultat final de ce projet. Par conséquent, des changements ou des extensions de ces ontologies, avec des cartographies nouvelles de l'ontologie référente qui pourrait apparaître dans le futur, devraient être réalisés par les parties prenantes ou parties intéressées une fois le projet fini.

Grâce à l'interaction avec les participants du deuxième workshop, nous avons ajouté les éléments suivants à la liste de sélection, après avoir réalisé la deuxième activité:

- ZigbeeHA
- CENELEC EN50491
- Adapt4EE

[2] https://sites.google.com/site/smartappliancesproject/ontologies

Durant l'Activité 3, nous avons créé une première version de l'ontologie de référence des appareils intelligents (Smart Appliances REFerence – SAREF -) qui, de manière explicite, spécifie les concepts centraux récurrents dans le domaine des appareils intelligents, les relations principales entre ces concepts, ainsi que les axiomes qui limitent l'usage de ces concepts et relations. SAREF est basée sur les principes fondamentaux de *réutilisation* et *d'alignement* des concepts et relations définis dans les éléments existants, de *modularité* afin de permettre la séparation et recombinaison des différentes parties de l'ontologie en fonction des besoins spécifiques, *d'extensibilité* permettant le développement futur de l'ontologie, et de *maintainabilité* facilitant le processus d'identification et de corrections des défauts, de prise en compte de nouveaux besoins et possibles changements (de parties) de l'ontologie SAREF. Nous avons ensuite mis côte à côte SAREF et l'Architecture ETSI M2M, et avons découvert qu'il existe une bonne corrélation entre l'Architecture ETSI M2M et les catégories relatives aux fonctions des appareils. La corrélation avec les catégories des appareils relatives à l'énergie et au bâtiment reste minime. Les résultats de l'Activité 3 furent présentés dans le *troisième rapport intermédiaire D-S3*. D-S3 a de même été révisé par le groupe Expert du projet, et plus tard discuté durant le troisième workshop des protagonistes et parties prenantes, qui eut lieu à Bruxelles, le 10 février 2015.

Le résultat final de l'étude des appareils intelligents réalisée par TNO est concrétisé par le *rapport d'étude final D-S4*, qui constitue l'un des trois rapports intermédiaires produits durant l'étude, rapport revu suite aux réactions et retour d'information communiqué par les protagonistes et parties prenantes, ainsi que du groupe Expert. Le rapport D-S4 fut officiellement présenté durant le quatrième workshop des parties intéressées le 1er avril à Bruxelles. D-S4 sera officiellement remis à l'ETSI Smart M2M pour être développé plus avant, en tant que Spécification Technique, tel qu'il est prévu en ce moment. Lorsque le projet sera terminé, c'est à l'industrie de maintenir et d'enrichir l'ontologie référente si nécessaire. Notre recommandons qu'ETSI soutienne ce processus. Le call ICT 30 d'Horizon 2020 est un excellent moyen de soutenir plus avant les activités de recherche et développement.

L'ontologie des appareils intelligents (Smart Appliances REFerence – SAREF-) est disponible en ligne : http://ontology.tno.nl/saref. La version Turtle de l'ontologie SAREF peut être téléchargée à l'adresse suivante: http://ontology.tno.nl/saref.ttl, et peut être ouverte grâce à n'importe quel éditeur d'ontologie, tel que TopBraid Composer, Protégé et NeOn.

Contents

Abbreviations

3G	Third Generation
AC	Alternating Current
AIM	A novel architecture for modelling, virtualising and managing the energy consumption of household appliances
AMM	Automated Meter Management
API	Application Programming Interface
BACnet	Building Automation and Control Networks
BACS	Building Automation and Control Systems
BEMO-COFRA	Brazil-Europe - Monitoring and Control Frameworks
BEMS	Building Energy Management Systems
BIM	Building Information Model
CECED	European Committee for Domestic Equipment Manufacturers
CEM	Customer Energy Managers
CEN	European Committee for Standardization
CENELEC	European Committee for Electrotechnical Standardization
CLC	CENELEC
CoAP	Constrained Application Protocol
COSEM	Companion Specification for Energy Metering
CSEP	Consortium for SEP2 Interoperability
DAL	Device Abstraction Layer
DCP	Device Control Protocol
DECT	Digital Enhanced Cordless Telecommunications
DEHEMS	Digital Environment Home Energy Management System
DHCP	Dynamic Host Configuration Protocol
DLMS	Device Language Message Specification
DnS	Descriptions and Situations
DogOnt	Ontology Modeling for Intelligent Domotic Environments
DOLCE	Descriptive Ontology for Linguistic and Cognitive Engineering
DomoML-env	An ontology for Human Home Interaction
DPWS	Devices Profiles for Web Services
DUL	DOLCE+DnS Ultralite
E2BA	Energy Efficient Buildings Association
Ebbits	Enabling business-based Internet of Things and Services
EC	European Commission
ECHONET	Energy Conservation and HOmecare NETwork
eDiana	Embedded Systems for Energy Efficient Buildings
EE	Energy Efficiency
EEP	EnOcean Equipment Profiles
EDI	Electronic Data Interchange
EHS	European Home System
ELC	European Lamp Companies Federation
EMU	Energy Management Unit
ENERsip	ENERgy Saving Information Platform
EN	European Norm

ENV	Environmental and Contextual data
EP	Energy Profile
EPI	Energy Performance Indicators
ERP	EnOcean Radio Protocol
ESCO	Enery Service Company
ESO	European Standardization Organisation
ETSI	European Telecommunications Standards Institute
EU	European Union
eu.bac	European building automation controls association
EupP	Energy using and producing Product
FAN	FlexiblePower Alliance Network
FIEMSER	Friendly Intelligent Energy Management Systems in Residential Buildings
FIPA	Foundation for Intelligent Physical Agents
FP7	European 7th Framework Program
FPAI	Flexible Power Application Infrastructure
FttH	Fiber to the Home
GENA	General Event Notification Architecture
GHz	Gigaherz
HAN	Home Area Network
HAN FUN	Home Area Network FUNctionality
HFC	High Frequency Communication
HGI	Home Gateway Initiative
HTTP	Hypertext Transfer Protocol
HVAC	Heating, ventilation, and air conditioning
Hydra	Heterogeneous physical devices in a distributed architecture
ICT	Information and Communication Technologies
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IES	Illuminating Engineering Society
IETF	Internet Engineering Task Force
IFC	International Foundation Classes
IoP	Internet of People
IOPTS	Internet of People, Things and Services
IoS	Internet of Services
IoT	Internet of Things
IP	Internet Protocol
IPR	Intellectual Property Rights
IRIS	International Repair Information System
ISO	International Organization for Standardization
kbps	kilobit per second
KNX	KNX Association
kWh	kilowatt-hour
kW	kilowatt
LDN	Logical Device Name

LEP	Local Energy Providers
LonWorks	Local Operating Network
LWM2M	Lightweight M2M
M-BUS	Meter Bus
M2M	machine-to-machine
ME3GAS	Middleware for Energy Efficient Embedded Services & Smart Gas Meters
MDA	Model Driven Architecture
MIRABEL	Micro-Request-Based Aggregation, Forecasting and Scheduling of Energy Demand, Supply and Distribution
MUC	Multi Utility Communication
OASIS	Organization for the Advancement of Structured Information Standards
oBIX	Open Building Information Exchange
OBIS	Object Identification System
OMA	Open Mobile Alliance
OMS	Open Metering System
OMS-DPL	OMS-Data Point List
OSGi	OSGi technology (defined by the OSGi Alliance)
OWL	Web Ontology Language
OWL-DL	Web Ontology Language – Description Logic
OpenIoT	Open Source cloud solution for the Internet of Things
PC	Personal Computer
PHEV	Plug in Hybrid Electric Vehicle
PI	Product Information
PLC	Power Line Carrier
PowerOnt	Power Profiling for Intelligent Domotic Environments
R&D	Research & Development
RDF	Resource Description Framework
REST	REpresentational State Transfer
RF	Radio Frequency
RFC	Request for Comments
SAREF	Smart Appliances REFerence ontology
SCL	Service Capability Layer
SD	Study Document
SDK	Software Development Kit
SDO	Standard Development Organization
SEEMPubS	Smart Energy Efficient Middleware for Public Space
SEIPF	Semantic Energy Information Publishing Framework
SensorML	Sensor Model Language
SEP2	Smart Energy Profile 2.0
SESAME	SEmantic SmArt MEtering – Services for Energy Efficient Houses
SG-CG	Smart Grid Coordination Group
SIG	Special Interest Group
SKOS	Simple Knowledge Organization System

SmartCoDe	Smart Control of Demand for Consumption and Supply to enable balanced, energy-positive buildings and neighbourhoods
SML	Smart Message Language
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SSDP	Simple Service Discovery Protocol
SSN	Semantic Sensor Network Ontology
SUMO	Suggested Upper Merged Ontology
SWE	Sensor Web Enablement
TC	Technical Committee
TIBUCON	Towards Integral BUilding CONnectivity
TM	Technical Memorandum
TNO	Netherlands Organisation for Applied Scientific Research TNO
TR	Technical Report
TRV	Thermostat Radiator Valves
TV	Television
ULE	Ultra-Low Energy
UML	Universal Markup Language
UPnP	Universal Plug and Play
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
USR	User Preferences
VIB-type	Vibration-type
VoCamp	Vocabulary Camp
W3C	World Wide Web Consortium
WG	Working Group
WGS84	World Geodetic System 1984
Wi-Fi	Wireless Fidelity
WP	WorkPackage
WSN	Wireless Sensor Network
xDSL	x Digital Subscriber Line
XML	eXtensible Markup Language
XSD	XML Schema Definition Language

1. Introduction

1.1. Context

Achieving higher energy efficiency is an important goal for the European society. The residential and tertiary sector, the major part of which are buildings, accounts for more than 40% of the final energy consumption in the European Community and is expanding, a trend which is bound to increase its energy consumption and hence its carbon dioxide emissions [1]. It is not so much the buildings as such that consume energy and produce greenhouse gasses, but the so-called Energy using and producing Products (EupP), also called “appliances”, inherently present in the buildings’ ecosystems, and the people using them.

An appliance is an instrument or device designed for a particular use or function. About two thirds of the energy consumed by buildings originates from the residential sectors and thus household appliances. Household appliances or home appliances are electrical/mechanical machines which accomplish some household functions, such as cooking or cleaning. The broad definition allows for nearly any device intended for domestic use to be a home appliance, including stoves, refrigerators, toasters, air conditioners as well as TVs, PCs, and light bulbs. Home appliances can be classified into major appliances (or White goods), small appliances (or Brown goods), and consumer electronics (or Shiny goods).

Nowadays, appliances are not stand-alone systems anymore. They are often highly intelligent (“smart”) and networked devices, that form complete energy consuming, producing, and managing systems. Therefore, reducing the use of energy and production of greenhouse gasses is not only a matter of increasing the efficiency of the individual devices, but managing and optimizing the energy utilization at a system level. One of the requirements for making such systems adopted by the mass market, is the flexible and dynamic extension with new smart devices and applications, based on the user’s needs and available budget. The systems will therefore inevitably consist of devices and sensors from different vendors, and open interfaces enabling further extensions. An open interface is a public standard for connecting hardware to hardware and software to software. Said otherwise, networked devices can be managed for energy saving measures if there is a system that can be flexibly enhanced. They also need to be able to communicate with service platforms from different service providers.

In such a system, the interfaces need to be properly standardized and offer external access on a semantic level both to any manageable and controllable function of the system as a whole, and to any device that is part of the system. However, the problem is not the lack of available standards. Actually, there already exist (too) many standards, all dealing with a smaller or larger part of the problem, sometimes overlapping and competing [2]. What is needed is a reference ontology, a shared data model.

Various workshops and FP7 projects already have explored this field and concluded that defining a useful and applicable reference data model should be possible in principle. Several of those exploratory discussions were held at the Energy Efficiency research community at the 2nd (2011) and 3rd (2012) Workshop on eeBuildings Data Models (Energy Efficiency Vocabularies and Ontologies).

These workshops presented results of FP7 and Artemis funded projects[3] related to energy efficiency with different approaches and solutions to bridge over the connectivity standards "jungle" for the smart appliances, but more importantly, explored expanded semantic ontologies to cover broader areas of interactions (more intelligent machine-to-machine "conversations") as the ones covered by the traditional control networks. The conclusion from these workshops were the following: Indeed, one single, reference ontology can be created to cover the needs of all appliances relevant for energy efficiency; indeed, this ontology can be designed in a way that it can be expanded to cover future intelligence requirements; and indeed, this ontology is a rather simple ontology as compared to the state of the art ontology engineering level of complexity. The workshops also concluded that these models show high mapping correlations, and that all what is needed is a formal agreement, a recognised standard and combined efforts of standardization organizations.

However, before launching a formal exercise, the industry was consulted to discover their support and their perception of this need. On 24 September 2012 the European Commission (EC) hosted a workshop on a roadmap for the standardization of smart appliances, inviting all relevant stakeholders:

<u>Stakeholders associations</u>

- Energy Efficient Buildings Association (E2BA)
- CECED, European Committee for Domestic Equipment Manufacturers
- eu.bac, European building automation controls association
- ELC, European Lamp Companies Federation (now succeeded by LightingEurope)
- Smart Grid Task Force
- Agora du Réseau Domiciliaire

<u>Standardisation Bodies and Organisations</u>

- ETSI M2M (now called ETSI Smart M2M)
- CENELEC TC59x WG7, Smart Grid/Smart Home Activities
- HGI Home Gateway Initiative
- buildingSmart International
- OASIS Open Building Information Exchange (oBIX)
- OSGi Alliance

The main recommendation of this meeting consisted of two objectives:

1. Propose a high-level semantic modelling of information to be exchanged (API-like) – the first step is a common vocabulary for appliances product information, commands, signals (like price or sensor information) and feedback.
    a. Take stock of the existing semantic assets, across different stakeholders and standardisation efforts, and perform a translation exercise. Agree on a nuclear vocabulary.
    b. Discuss a complete range of use cases, covering all devices (white goods, HVAC, plumbing, security and electrical systems, lightings, sensors and actuators (windows,

[3] E.g. SmarCoDe (www.fp7-smartcode.eu), eDiana (www.artemis-ediana.eu), ENERsip (www.enersip-project.eu), and FIEMSER (www.fiemser.eu)

doors, stores), micro renewable home solutions (solar panels, solar heaters, wind, etc.), multimedia and home computer equipment and all Building Energy Management Systems (BEMS), Building Automation and Control Systems (BACS), Customer Energy Managers (CEM), and Energy Boxes as defined by the Consumer Electronics industry, finding the messages and signals they may need to share. Extend the nuclear vocabulary.

2. With regard to connectivity, agree on an abstract architecture with a clear horizon and considering the world's machine-to-machine (M2M) standards, approaches and architectures to bridging the manifold communication layers already available.
    a. Propose available architectures that go in that direction
    b. Create open repositories of reusable pieces

With regard to objective 1, the European Commission has the intention to launch a standardisation exercise at ETSI to propose this high-level model, an ontology for smart appliances, as an ETSI standard. With regard to objective 2, the results should be integrated in the abstraction layer of the ETSI M2M architecture for the Home and Building environment.

1.2. Goal and objectives of this study

To provide this ETSI working group with the relevant background, the European Commission issued a tender for a Study on "Available Semantics Assets for the Interoperability of Smart Appliances. Mapping into a Common Ontology as a M2M Application Layer Semantics" [3], defining 3 tasks:

- **Task 1:** Take stock of existing semantic assets and use case assets
- **Task 2:** Perform a translation exercise of each model (or use case) to a common ontology language and a mapping or matching exercise between all the models
- **Task 3:** Propose a reference ontology and document the ontology into the ETSI M2M architecture

The study will thus contribute with recommendations for a reference ontology, based on semantic assets defined and examined within this study.

TNO was invited to perform this study. The study aims to provide the material needed to define these tools and data models, for the collection of devices that helps the EU to reach its 2020 goals regarding the reduction of greenhouse gas emission and buildings' energy consumption, being the said appliances. The work packages and tasks defined in the study will fulfil the following objectives:

- An overview of existing explicit or implicit semantic assets and use case assets.
- Detailed analysis of the existing semantic assets or requirements in an exhaustive way.
- Proposal for a reference ontology to be contributed to ETSI for consideration as a future standard.
- Documentation of the proposed ontology into the ETSI M2M architecture.

The first document, *D-S1 Interim Study Report*, presented the results of task 1 "take stock of existing semantic assets and use case assets". D-S1 was first reviewed by the project's Expert Group, and later on discussed in the 1st stakeholders' workshop that took place in Brussels on May 27/28, 2014. *D-S2 Second Interim study report,* resulted from task 2 "perform a translation exercise of each model (or use case) to a common ontology language and a mapping or matching exercise between all the models" and covers a translation of the most relevant assets identified in task 1 into OWL ontologies and an initial mapping between these ontologies. D-S2 was discussed at the 2nd stakeholders' workshop at the ETSI premises in Sophia Antipolis on October 15, 2014. The third interim deliverable, *D-S3 Third Interim study report*, covers the definition of the Smart Appliances REFerence (SAREF) ontology and a description of this ontology within the ETSI M2M architecture. It was discussed in the 3rd stakeholders' workshop on the 10th of February 10, 2015. This deliverable, D-S4 Final Study report, includes all results from the previous interim deliverables and contains updates to ensure that the report contains the newest insights at the time of publishing.

1.3. Study process

The study started in January 2014 with the stock taking of existing semantics assets and use case assets. This means that the long list is largely based on a snapshot of the state of the art in spring 2014. Models may have been developed and evolved since, but not taken into account in this document. Based on the long list a first version of the short-list was defined:

- ECHONET
- FIEMSER
- UPnP
- SmartCoDE
- OMA Lightweight M2M
- SEP2
- EnOcean
- OMS
- Hydra
- KNX
- W3C SSN
- OSGi DAL
- eDIANA
- FAN
- DECT ULE
- Z-Wave
- SEEMPubs
- PowerOnt (previously SEIPF)
- FIPA
- Mirabel

For each of the assets on the first version of the short list we performed the translation of the model into a common ontology language and a mapping between the model and the SAREF ontology. The long list and the mappings were discussed during the second workshop. During this second workshop the participants pointed out two additional assets that were relevant for creating the SAREF ontology, these were ZigbeeHA and CENELEC EN50491. We added these two assets to the long list and short-list and took them into account when creating the SAREF ontology. Due to the fact that we got these assets late in the study process, we did not create a visualisation for both of the assets, we did not create an ontology for CENELEC EN50491 and we did not create a description for the ZigbeeHA ontology. Next to these two assets we added the Adapt4EE project as they provided us the ontologies they created. This asset was in the long list for the study but had not delivered models that we could use in spring 2014. We therefore did not add it to the first version of short list. We added Adapt4EE to the short list, but did not create description of the ontology. The second and final version of the short list that is presented in this deliverable therefore contains the following assets:

- ECHONET
- FIEMSER
- UPnP
- SmartCoDE
- OMA Lightweight M2M
- SEP2
- EnOcean
- OMS
- Hydra
- KNX
- ZigbeeHA
- CENELEC EN50491
- W3C SSN
- OSGi DAL
- eDIANA
- FAN
- DECT ULE
- Z-Wave
- SEEMPubs
- PowerOnt (previously SEIPF)
- FIPA
- Mirabel
- Adapt4EE

1.4. Structure of the document

Chapter 2 describes the scope of the study and in particular of this document. It provides a brief introduction on ontologies and the ontology language of choice (OWL-DL). Chapter 3 presents the long list of assets that have been investigated during the study and presents the short list that was used to build the reference ontology. Chapter 4 presents the visualisation of the key terms of the assets in the short list. Chapter 5 presents the ontologies that were created during the study for the assets in the short list, in chapter 6 the mapping of the different ontologies to the Smart Appliances Reference ontology (SAREF) are presented. Chapter 7 present the SAREF ontology and in chapter 8 we defined how the SAREF ontology can be applied in relation to the ETSI M2M architecture. Chapter 9 presents our conclusions.

2. Scope

2.1 Sectors, use cases and appliances

Our study mainly addresses the consumer (mass) market of the home, private dwellings, but also common public buildings and offices, and the standard appliances used in that environment. Elevators and other special equipment are out of scope.

The following appliances are covered:

- Home and buildings sensors (temperature, humidity, energy-plugs, energy clams, energy meters, water-flow, water quality, presence, occupancy, air monitors, environmental sensors, CO_2 sensors, weather stations, etc.) and actuators (windows, doors, stores). Sensors belonging to appliances are treated individually.
- White goods, as classified by CECED[4]
  - Rinsing and Cleaning
  - Cooking and Baking
  - Refrigerating and Freezing
  - Vacuum Cleaning
  - Washing and Drying
- HVAC; heating, ventilation, and air conditioning, plumbing, security and electrical systems, as classified by Eu.bac[5]
- Lighting, with use cases as defined by LightingEurope[6] (formerly known as ELC)
- Micro renewable home solutions (solar panels, solar heaters, wind, etc.)

Multimedia and home computer equipment devices will be explored only with respect for semantic requirements for the energy relevant operations (switch on, standby), but not for the content management (i.e. channel choice).

The study further covers the following interoperability use cases:

- Interoperability with construction design tools (product information, product performance and product behaviour)
- Interoperability with Facility Management and Energy Management Systems
- Interoperability with Building Control systems
- ESCO (Energy Services) systems
- Interoperability with the Smart Grid

As primary stakeholders the manufacturers of the following home energy producing and consuming products are consulted:

- Manufacturers of white goods
- Manufactures of HVAC, plumbing, security and electrical systems
- Manufacturers of lightings
- Manufacturers of sensors and actuators (windows, doors, stores)
- Manufacturers of micro renewable home solutions (solar panels, solar heaters, wind, etc.)
- Manufacturers of multimedia and home computer equipment

[4] European Committee of Domestic Equipment Manufacturers, www.ceced.org
[5] European building automation controls association, www.eubac.org
[6] www.lightingeurope.org, the successor of the former ELC (European Lamp Companies federation)

Furthermore stakeholders from directly linked industries are consulted:

- Construction industry
- Facility Management and Building Control industry
- ESCO (Energy Services Providers)
- Utilities and operators of the power grid

2.2 ETSI Smart M2M

One of the SDOs (Standard Development Organization) playing a key role in this ecosystem is ETSI Smart M2M. ETSI[7] is one of the world's leading standards development organizations for Information and Communication Technologies (ICT). Founded initially to serve European needs, ETSI is now a highly-respected producer of technical standards for worldwide use. ETSI membership is composed of manufacturers and network operators plus national administrations, ministries, regulators, universities, research groups, consultancies and user organizations from more than 60 countries on 5 continents.

Machine to Machine (M2M) is a term being used to describe the technologies that enable computers, embedded processors, smart sensors, actuators and mobile devices to communicate with one another, take measurements and make decisions - often without human intervention [4]. ETSI has created a dedicated Technical Committee, ETSI Smart M2M (previously known as ETSI M2M) with the mission to develop standards for M2M communications. The group will provide an end-to-end view of M2M standardization.

ETSI M2M recently released its Functional Architecture [5], which describes the overall end-to-end M2M functional architecture, including the identification of the functional entities and the related reference points.

2.3 OneM2M

ETSI Smart M2M has finished their work on the M2M functional architecture with [5]. In 2012, the OneM2M Partnership Project[8] was formed with the goal to develop the technical specifications which address the need for a common M2M Service Layer like described in [5], that can be readily embedded within various hardware and software, and relied upon to connect the myriad of devices in the field with M2M application servers worldwide. OneM2M is a worldwide industrial organization and is not tied to the telecommunications sector or the ETSI scope, regulations, and infrastructure. OneM2M's scope includes:

- Use cases and requirements for a common set of Service Layer capabilities;
- Service Layer aspects with high level and detailed service architecture, in light of an access independent view of end-to-end services;
- Protocols/APIs/standard objects based on this architecture (open interfaces & protocols);
- Security and privacy aspects (authentication, encryption, integrity verification);
- Reachability and discovery of applications;
- Interoperability, including test and conformance specifications;
- Collection of data for charging records (to be used for billing and statistical purposes);
- Identification and naming of devices and applications;

[7] www.etsi.org
[8] www.onem2m.org

- Information models and data management (including store and subscribe/notify functionality);
- Management aspects (including remote management of entities); and
- Common use cases, terminal/module aspects, including Service Layer interfaces/APIs between:
    - Application and Service Layers;
    - Service Layer and communication functions

OneM2M's Working Group 5 "Management, Abstraction and Semantics" will focus on the technical aspects related to management of M2M entities and/or functions. This WG will also focus on providing support by the M2M system for application specific abstraction and semantics with regard to execution of M2M services. Working Group5 is working on a draft TR-007 "Abstraction & Semantics Capability Enablement" which is expected to absorb the results of ETSI Smart M2M in this field in due time.

2.4 About ontologies

An ontology is here defined as a formal specification of a conceptualization, used to explicit capture the semantics of a certain reality [5,6,7]. As such, we regard an ontology as:

– a set of concepts used to describe the reality under consideration e.g., the concepts of 'household appliance', 'task' and 'function';
– precise definitions of these concepts in natural language e.g., 'an appliance is a tangible object designed to accomplish a particular task in households, such as cooking or cleaning. In order to accomplish this task, the appliance performs one or more functions.';
– relations among these concepts e.g., a household appliance of type 'washing machine' accomplishes the task 'cleaning' and to accomplish this task performs the function 'start and stop'; and
– axioms to constrain the intended meaning of these concepts, e.g., special conditions under which an appliance should function, such as a specific timeslot during the night when the energy costs are reduced.

Users of the ontology can define instances of ontology concepts that are relevant to them, e.g., the specific household appliance of type 'washing machine' from manufacturer 'A and with serial number '123xyz'.

In this study, ontologies are used to improve the communication among stakeholders, providing a shared understanding that reduces ambiguities and confusion in the terminology adopted in the smart appliances domain. Ontologies are also used here to provide an interpretation to data and, therefore, facilitate interoperability between systems and devices provided by different vendors, providing a reference model that allows translation and mapping among different assets (models/standards/software) from different parties [4].

Ontologies require a language that is suitable to represent the ontology concepts. In practice, people often refer to ontologies as what are in fact specifications of conceptualizations loosely expressed in an informal language, such as natural language. These are not ontologies according to the definition adopted here. In contrast, we consider ontologies as formal specifications expressed using formal semantics and axioms [8]. Informal specifications may lead to ambiguities, and systems that are based on such specifications are more error-prone than systems based on formal ontologies, which, in contrast, allow automated reasoning and consistency checking. Therefore, ontologies expressed using formal semantics are engineering artifacts that can be processed and checked by machines.

It is important to choose a suitable language depending on the purpose of an ontology. The language chosen in the smart appliances study to express the ontologies corresponding to the semantic assets is OWL-DL [39], since it provides formal semantics to explicitly represent the meaning intended for these assets, and allows a high degree of semantic reasoning, being supported by a large number of software reasoning tools. In an OWL-DL ontology, the concepts used to describe the reality under consideration are called *classes*, the natural language definitions of these classes can be annotated as *comments*, the instances are called *individuals*, the relations are called *properties*, and the axioms are called *restrictions*.

3. Assets

This chapter provides an overview of the assets studied, including the long list and the short list that resulted from the study of the assets.

3.1 Long list

The Invitation to Tender [3] lists the following assets to be studied (proper references can be found in sections 3.3 and 3.4):

- From E2BA and the eeSemantics community:
    - eeSemantics Wiki and eeSemantics library of ontologies
    - FIEMSER Data Model for Monitoring & control network
    - eDiana (ARTEMIS) ontology for device discovery and interoperability
    - ENERsip M2M Communications Infrastructure Modelling Ontology
    - SmartCoDe EUP classification with respect to energy management
    - TIBUCON
    - SEEMPubS Ontology
    - SEIPF ontology (now called PowerOnt)
    - DEHEMS Digital Environment Home Energy Management System project
    - AIM*
    - Ebbits
    - SESAME Demonstrator
    - LinkSmart ontologies
    - Adapt4EE
    - MIRABEL Flex
    - eeSemantics EupP VoCamP
- Hydra
- From CECED, the EDI-SERVICE, EDI-WHITE, IRIS, and PI standards
- From European Lamp Companies Federation (now called Lighting Europe), the CEN/TC 169 - CIE Newsletter - searchable online database International Lighting Vocabulary
- buidlingSmart’s Industry Foundation Classes (IFC)
- FIPA
- Semantic Sensor Network Ontology (SSN)
- CEN/CLC/ETSI Smart Grid CG M490
- BACnet
- KNX
- LonWorks
- Related to BACnet, KNX, and LonWorks: the assets produced by Eu.Bac and various CENELEC TCs
- oBIX
- SensorML

Our study identified various other assets that need to be included also, given the scope of the study as presented in section 2.1:

- ECHONET
- HGI
- UPnP
- Agora
- OMA Lightweight M2M
- SEP2 (Zigbee, HomePlug, Wi-Fi)
- EnOcean
- OMS
- Zigbee HA
- CENELEC EN 50491-12
- Broadband Forum
- OSGi DAL
- Energy@Home
- FAN
- DECT ULE
- Z-Wave
- DLMS/COSEM
- CoAP

3.2 Short list

In order to perform the stocktaking task, we have followed a systematic approach that allowed us to deal with the quantity of assets to be considered and their complexity.

The first step was to identify the most representative resources that could characterize a specific asset. This activity resulted in the selection of a few documents for each asset, usually the specification of a standard, a project deliverable, sometimes a presentation, and in a few cases a scientific paper. We also collected XML schema specifications when available. On average, we have selected two or three documents per asset. The length ranged from 20 pages to more than 1000 pages, the latter being the case of the BACnet specification, however we encountered an average length of 100 pages per document. Note that whenever a project deliverable or a paper described an ontology, our analysis was based only on the documentation and excerpts of the ontology provided by these documents, but not on an actual OWL file, since in some cases no URL was provided nor we could find it searching on the Web. This was the case for eDiana, FIPA, SEIPF, DEHEMS, ebbits and SESAME.

The second step was to analyze the selected documents for each assets. The content and level of details of the selected documents was different but mainly spanning 1) architectural overviews useful to have a general idea of the considered asset, 2) explicit data models or ontologies from which we could (more or less) straightforward attempt to define a semantic coverage, 3) protocol descriptions with implicit semantics, 4) low-level data-container specifications from which it was cumbersome to capture a semantic coverage.

The third step was to define a semantic coverage based on key terms found in the selected documents, and give an initial indication whether the asset was aligned with the scope of our study, and/or whether the semantics was sufficiently explicit to be translated into an ontology, regardless of the fact that an ontology was actually provided. For example, some documents did not directly provide an ontology, but provided an explicit, clear and well-structured specification in natural language with the support of tables.

The fourth step was to make a short list to narrow down the 47 assets based on the following criteria:

1. Given the scope of this study "Available Semantics Assets for the Interoperability of Smart Appliances. Mapping into a Common Ontology as a M2M Application", does the considered asset target M2M Devices that run M2M Application(s) using M2M Service Capabilities, as described in Section 2.2?
2. Does the considered asset address (one of) the smart appliance categories as laid out in section 2.1?
3. Does the asset provide a clear and well-structured specification that can be used to explicitly capture the semantics in an ontology?
4. Could the asset be translated into an ontology with maximum 20-25 classes and some corresponding relations, or would this simplification harm the actual purpose and meaning of the asset?
5. If the entire asset seems to be too extensive and complex to be reduced to 20-25 classes, could we focus on one single part/module that is especially relevant according to the criteria 1 and 2 mentioned above?
6. Does the asset directly provide an ontology and OWL representation?
7. Does the asset provide XML schemas that could be used to automatically support the translation to an OWL ontology?

The reader should be aware that this short list is NOT an endorsement by TNO or the European Commission, NOR does it signify anything about the relevance of the underlying technology for the industry or the market, nor any other commercially valuable qualification. On the contrary, most long listed bodies and projects have provided useful assets, if not in the form of ready-to-use XML or OWL files, such as use cases, low-level data-container specifications, and interoperability specifications.

The following assets were identified as providing a good basis for further ontology development in this study. They are described in detail in section 3.3, one asset per subsection, in alphabetical order. The other assets are summarily described together in section 3.4. The short list consists of:

- ECHONET
- FIEMSER
- UPnP
- SmartCoDE
- OMA Lightweight M2M
- SEP2
- EnOcean
- OMS
- Hydra
- KNX
- Zigbee HA
- CENELEC EN 50491-12
- W3C SSN
- OSGi DAL
- eDIANA
- FAN
- DECT ULE
- Z-Wave
- SEEMPubs
- PowerOnt (previously SEIPF)
- FIPA
- Mirabel
- Adapt4EE

3.3 Asset descriptions

3.3.1 Adapt4EE

Most relevant URLs, and other precise references

D3.2 Adapt4EE Middleware Specification, Ontology and Semantic Components, May 2013, http://www.adapt4ee.eu/adapt4ee/files/document/deliverables/Adapt4EE-Deliverable-D3.2.pdf. Edited by Fraunhofer, CERTH, and Technical University Kosice

Overall description

Adapt4EE[9] is a European 7th Framework R&D project running from 2011-2015. It aims to develop and validate a holistic energy performance evaluation framework that incorporates architectural metadata (Building Information Model, BIM), critical business processes (BPM) and consequent occupant behaviour patterns, enterprise assets and respective operations as well as overall environmental conditions. The Adapt4EE framework, having as a central point of reference the occupancy behaviour (presence and movement) will align energy consumption points to all interrelated enterprise aspects (business processes, enterprise assets and utility state and operations). As part of the work, Adapt4EE will develop an enhanced semantic enterprise model that treats, learns and manages the enterprise environment as an intelligent agent, perceives environmental state using multi-type sensors and information modalities. The Adapt4EE Model will incorporate business processes and occupancy data. It will also constitute a formal model for enterprise energy performance measuring, monitoring and optimization. The model will be calibrated during the training phase based on sensor data captured during operation and then applied and evaluated in real-life every day enterprise operations. More specifically the Adapt4EE Enterprise Models will allow for the proactive identification of optimum local adaptations of enterprise utility operations, based on predictions of possible occupancy patterns and respective business operations and energy profiles. D3.2 "Adapt4EE Middleware Specification, Ontology and Semantic Components" contains semantic assets. It is edited by by Fraunhofer, CERTH, and Technical University Kosice.

Description of the semantic coverage

D3.2 describes the Adapt4EE middleware architecture, its components, its interactions with other modules in the system, and in particular the ontology representing the information used by the modules. The LinkSmart Middleware is used to integrate existing sensor and building management technology systems with the technologies developed in Adapt4EE. Core concepts such as LinkSmart-enabled Device, LinkSmart Device, Gateway, LinkSmart Manager, and Device Discovery are defined. Subsequently the LinkSmart Device Ontology is presented that contains knowledge about device classes, their properties and services offered. The deliverable also includes an example of the semantic model in OWL for a use case, including a table of the classes and properties in the example.

Of the Adapt4EE Ontology, the events part and the BIM model are the most relevant to our project. The semantic coverage is depicted in Figure 1, and includes Events (sensor events and context events), LinkSmart-enabled Device, LinkSmart Device, Gateway, LinkSmart Manager, Device Discovery. The semantic coverage is obviously partially overlapping with the result from the HYDRA project. From the LinkSmart Device Ontology we derive:

[9] www.adapt4ee.eu

- Device Classifications of LinkSmart Devices. A LinkSmart Device is the software representation of a LinkSmart-enabled physical device. Type of devices are D0 Device, D1 Device, D2 Device, D3 Device, and D4 Device. This classification scheme helps developers to decide how a specific device is to be integrated into a network.
- Runtime Architecture. LinkSmart facilitates communication among devices via a P2P overlay network. The basic LinkSmart component enabling network communication is the NetworkManager. Inside a LinkSmart network, unique Internet Of Things Identifiers (IoTID-s) identify all devices and services.
- Event Management. The Event Manager implements a publish/subscribe mechanism on the level of Web Services. The LinkSmart Event Manager provides decoupling in space and synchronization through a content-based publish/subscribe mechanism. In this type of publish/subscribe, subscribers subscribe on topics and receive events that are published by publishers on that topic through a notification mechanism.

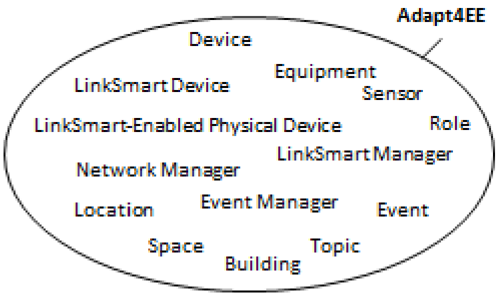


Figure 1: Visual representation of the semantic coverage of Adapt4EE

3.3.2 DECT ULE

Model Acronym and Full Name

DECT (Digital Enhanced Cordless Telecommunications) ULE (Ultra-Low Energy) HAN FUN (Home Area Network FUNctionality)

Most relevant URLs, and other precise references

DECT ULE HAN FUN is described in 5 complementary documents:

- HF-Overview V1.00 (2014-23-1)
- HF-Protocol V1.00 (2014-23-1)
- HF-Service V1.00 (2014-23-1)
- HF-Interface V1.00 (2014-23-1)
- HF-Profile V1.00 (2014-23-1)

They can be downloaded for free after registration at http://www.ulealliance.org/registration.aspx?f=11

Overall description

DECT is a wireless protocol used for in-home cordless phones. The air interface has been standardized by ETSI as EN 300 175 [10]. ULE is its Ultra-Low Energy variety of DECT, which is designed for use of DECT with home automation, security, and climate control devices and applications. The ULE specification was created as an initial cooperation between DECT Forum[10] and ETSI.

The physical layer of ULE makes use of the existing ETSI DECT specification EN 300 175-2. The technical specification work for the upper part of the ULE transport layer has been carried out in ETSI TC DECT, with the following updates:

- Medium Access Control Layer (EN 300 175-3)
- Data Link Control (EN 300 175-4)
- Network Layer (EN 300 175-5)
- Security (EN 300 175-7)
- Interworking Unit and Application Layer Protocol Negotiation (TS 102 939-1) [11]

ULE has its own application layer protocol called HAN FUN (Home Area Network FUNctionality) or simply HF. It has been released by the ULE Alliance[11] in November 2013. HAN FUN ensures interoperability for the specified applications and devices. In addition it supports proprietary extensions where required.

Description of the semantic coverage

The HF-Protocol document describes the network topology, the network entities (concentrator and device), the HF message formats and types. DECT is based on a star network topology, where a single Concentrator is the network's master device and supports up to thousands of HF Devices connected to it. A Unit is a conceptual entity inside a HF device that instantiates the functionality of a specific type. Unit types, for example smoke detector, simple switch and more are described in the HF-Profile document. An Interface is a conceptual entity inside a unit that defines a collection of commands and attributes, allowing for units to understand one another. Interfaces can be mandatory or optional to implement by a unit, and they have a role – client or server – associated with them. Interfaces are described in the HF-Interface document. The Service document lists the services of the HF standard. The semantic coverage is shown in Figure 2.

List of services:

- Device Management, Service responsible for device registration and discovery.
- Bind Management, Service that allows for the creation of logical communication links between devices.
- Group Management, Service that allows for the creation of logical groups for message broadcasts.
- Identify, Service that provides a simple method of identifying devices without the hassle of looking and matching serial numbers.
- Device Information, Service that defines information that any HF device can/must provide.

List of interfaces:

- Alert, Use when device wants to indicate an alert
- On-Off, Use to turn some device feature On or Off (you may also toggle it)
- Level Control, Use to set some device feature to a defined level
- Simple Power Metering, Use when device requires doing or providing measurements over electric quantities.

[10] www.dect.org
[11] www.ulealliance.org

– Reserved, Use for proprietary features (e.g. technical , manufacture, etc)
– Attribute Reporting , Service that allows a unit to receive automatic notifications about other units or device's attributes whenever an event triggers.
– Tamper Alert, Service that allows a device to indicate it is being tampered with.
– Time, Service that allows a device to maintain time referenced to UTC.
– Power, Service that allows a device to provide information about the power supplies it has and their characteristics.
– Keep Alive, Service that allows a device to signal it is alive.

List of Unit types:

– Home control unit types: Simple On-Off Switchable, Simple On-Off Switch, Simple Level Controllable, Simple Level Control, Simple Level Controllable Switchable, Simple Level Control switch, AC Outlet, AC Outlet with Simple Power Metering, Simple Light, Dimmable Light, Dimmer Switch, Simple Door Lock, Simple Door Bell, Simple Power Meter (definitions of each type are available in Table 3 of the document).
– Security unit types: Simple Detector, Door Open Close Detector, Window Open Close Detector, Motion Detector, Smoke Detector, Gas Detector, Flood Detector, Glass Break Detector, Vibration Detector, Siren ((definitions of each type are available in Table 3 of the document).
– Homecare Unit Types: Simple Pendant
– Application Unit Types: User Interface, Generic Application Logic
– Proprietary Unit Types: Proprietary

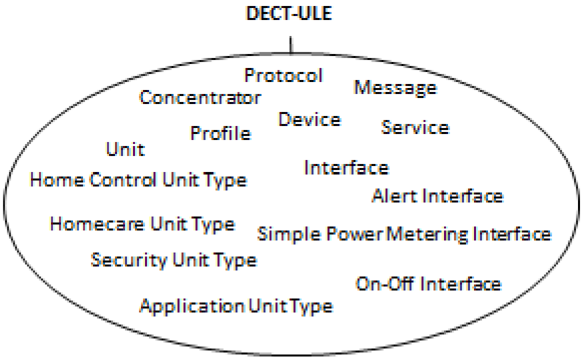


Figure 2: Visual representation of the semantic coverage of DECT-ULE

Overall description of the consensus driven process leading to the model

ULE Alliance has 5 Promotor Members, 9 Contributor Members, and 38 Adopter Members. Only the Promotor Members and Contributor Members pay a fee and can contribute. Promotor Members are the semiconductor manufacturers Dialog Semiconductor and DSP Group, and the device manufacturers Gigaset and Vtech. They drive the agenda of the alliance. Also DECT Forum is a Promotor Member. Contributors are Arcadyan, AVM, Cisco, The Crow Group, Deutsche Telekom, Lantiq, Panasonic, RTX, and Sercom.

3.3.3 CENELEC EN50491-12

Model Acronym and Full Name

CENELEC prEN50491-12:2014: General requirements for Home and Building Electronic Systems (HBES) and Building Automation and Control Systems (BACS) - Part 12: Smart grid - Application specification - Interface and framework for customer

Most relevant URLs, and other precise references

The prEN50491-12:2014 draft standard can be downloaded from http://www.cenelec.eu/dyn/www/f?p=104:110:143575753489701::::FSP_ORG_ID,FSP_PROJECT,FSP_LANG_ID:1258281,54098,25 with the right credentials. It can be obtained for a fee from the national standards development organizations, e.g. from NEN at http://www.nen.nl/NEN-Shop/Norm/NENEN-50491122014-Ontw.-en.htm .

Overall description

CENELEC prEN50491-12:2014 is a published draft of the upcoming EN50491-12 standard. The standard is expected to be ratified in 2016. It specifies the data model, to be used above the Application Layer by the interface between the Customer Energy Manager (CEM) and the mappings. The CEM is a local (in-home) automation function for optimizing the energy consumption / production according to the preferences of the customer and based on signals from the smart grid and various local flexibilities. A mapping is a function that connects entities (smart appliances, or local smart appliances networks such as Zigbee and KNX) to the CEM. The standard applies to public or private building / homes; industrial areas are excluded. It covers a technology independent data structure described in XSDs (XML schema Definitions) to be used to exchange information over the said interface. The standard is for a significant part based on previous work by Energy@Home (see section 3.4.11) and the EEBus organization (see below), which in turn based their work on Zigbee SEP2 (see section 3.3.18) and CIM [19].

Description of the semantic coverage

EN50491 defines an Entity as a logical representation of a smart device or a HBES/BACS (Home and Building Electronic System / Building Automation and Control Systems). The Entity Profile is a collection of Functional Profiles within an entity. A Function Profile is a combination of a Functional Capability and Functional Data. Example: Function Profile “Switching” with the Functional Capability “Switching” and Functional Data “ON” or “OFF”. The following Function Profiles are subsequently defined in XSD: General Functions, Switch Actuator Profile, Level Actuator Profile, Incentive Profile, Metering Profile, Measurement Profile, Time Information Profile, Device Classification Profile, Commodity Resource Profile, Sensing Profile, Messaging Profile, Load Management Threshold Profile, Power Sequences Profile, and Supply Condition Profile. Also Basic Data Types and Application Specific Data Types are specified. The standard provides many pages of non-normative data model examples.

Overall description of the consensus driven process leading to the model

The standard is produced by CENELEC TC205 (Home and Building Electronic Systems) with input from CENELEC TC59x (Performance of household and similar electrical appliances). The standard is for a significant part based on previous work by Energy@Home (a collaborative project between Electrolux, Enel, Indesit Company and Telecom Italia) and the EEBus organization. EEBus is a Germany-based consortium founded in 2011 and developing a standardised and consensus-oriented smart grid and smart home networking concept, initially developed by the Kellendonk company. In due time the organization also aims to go beyond smart home and smart building in order to develop a comprehensive concept for almost all smart devices. Founding members included ABB, Miele, Kellendonk, Vaillant, Schneider Electric, MVV, and Landis+Gyr. Today it counts 50 members including Bosch, BSH, Eaton, Devolo, Liebherr, Deutsche Telekom, Intel, Hager, and E.on.

3.3.4 ECHONET

Model Acronym and Full Name

ECHONET (Energy Conservation and HOmecare NETwork) Detailed Requirements for ECHONET Device Objects.

Most relevant URLs, and other precise references

ECHONET Specifications Appendix "Detailed Requirements for ECHONET Device Objects" Release C, 31 May 2013,
http://www.echonet.gr.jp/english/spec/pdf_spec_app_c_e/SpecAppendixC_e.pdf

Overall description

ECHONET[12] is a largely Japan-based consortium that promotes the development of basic software and hardware for home networks that can be used for remote control or monitoring of home appliances. For this purpose the consortium developed the ECHONET specifications and established basic technology for it. The aim in doing so has been to reduce CO_2 emissions while responding to the increasing sophistication of home security and home healthcare. ECHONET further developed home network technologies on home appliances and home facility equipment, and published "the ECHONET Lite Specification" in 2011, which is easier to use than the original ECHONET specifications, and enables interworking with other standard protocols. The ECHONET Specifications Appendix "Detailed Requirements for ECHONET Device Objects" is part of the Communication Middleware Specifications of ECHONET as well as ECHONET Lite.

Description of the semantic coverage

The semantic coverage is very detailed and schematically depicted in Figure 3. The main class (superclass) is the <u>Device Object</u> and it is specified with all its <u>properties</u>, such as <u>Operation status</u> (ON/OFF), Installation location (location at which the device has been installed, e.g., Living room, dining room, kitchen, bathroom, etc. or free definition), Standard version information (release order of the semantic model), Fault status (indicates whether a fault has occurred in the actual device), Fault description (code values for recoverable faults, faults that require repair, or other type of faults), Identification number (unique identifier in the domain), Measured instantaneous power consumption (in Watts), Measured cumulative power consumption (in increments of 0.0001kW), Manufacturer info, Product code, Production number and date, Power saving operation setting (if the device is operating in power-saving mode), Power limit setting (maximum consumable power), Current time, Current date.

The Device Object class is then specialized in sub-classes. One example is <u>Sensor-related device</u>, which includes Gas leak sensor, Crime prevention sensor, Emergency button, Humidity sensor, etc.

[12] www.echonet.gr.jp

Each of these sensor-related device classes (e.g., Gas leak sensor) is further detailed specifying its properties.

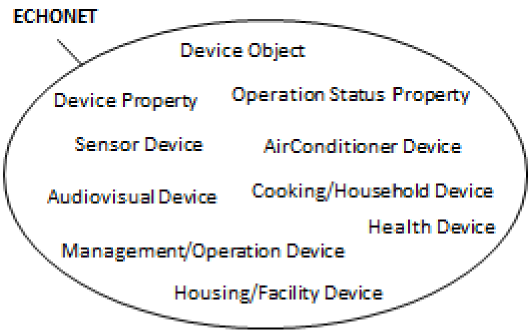


Figure 3: Visual representation of the semantic coverage of ECHONET

Other examples include:

- Air Conditioner-related device, for example, home air conditioner, electric fan, dehumidifier, electric blanket, etc. Each of these air conditioner-related device classes (e.g., home air conditioner) is further detailed specifying its properties.
- Housing/Facilities-related device, for example, electrically operated shade, electric water heater, floor heater, etc. Each of these housing/facilities-related device classes (e.g., electrically operated shade) is further detailed specifying its properties.
- Cooking/Household-related device, for example, refrigerator, washing machine, clothes dryer, etc. Some of these cooking/household-related device classes (e.g., refrigerator) are further detailed specifying their properties.
- Health-related device, such as weighing machine, clinical thermometer, blood pressure meter, blood sugar meter and body fat meter. Only the weighing Machine class is further detailed specifying its properties.
- Management/Operation-related device, such as secure communication shared key setup node, switch, portable (mobile) terminal and controller. Only the switch class is further detailed specifying its properties.
- Audiovisual-related device, such as display and television. Both the display and television classes are further detailed specifying their properties.

Overall description of the consensus driven process leading to the model

The consortium was founded in 1997 and now has 160 members. Member have a number of benefits, including the right to view and offer opinions concerning drafts during development of the ECHONET Specification. Only the so-called “managing members” have voting rights. They are Hitachi, Ltd., Mitsubishi Electric Corp., NIPPON TELEGRAPH AND TELEPHONE CORPORATION (NTT), Panasonic Corporation, Sharp Corp., SOFTBANK TELECOM Corp., Tokyo Electric Power Company, Inc., and Toshiba Corp.

3.3.5 eDIANA

Model Acronym and Full Name

eDIANA (Embedded Systems for Energy Efficient Buildings) Ontology for Device Awareness

Most relevant URLs, and other precise references

D2.2-A "Ontology for Device Awareness", 30 November 2009, http://s15723044.onlinehome-server.info/artemise/documents/D22A_Ontology_for_Device_Awareness_m10_IMSML.pdf.

Overall description

eDIANA[13] was an ARTEMIS[14] project running between 2009 and 2012. It aimed to address the need of achieving energy efficiency in buildings through innovative solutions based on embedded systems. The technical approach included the development of a reference model-based architecture, implemented through an open middleware including specifications, design methods, tools, standards, and procedures for platform validation and verification. The platform is designed to achieve the interoperability of heterogeneous devices at the Cell and MacroCell levels, and to provide the hook to connect the building as a node in the producer/consumer electrical grid. The architecture describes a network of composable, interoperable and layered embedded systems that will be instantiated to several physical architectures. The eDIANA Platform realisations will then cope with a variable set of location and building specific constraints, related with parameters such as climate, Cell/MacroCell configuration (one to many, one to one, etc.), energy regulations etc.

Description of the semantic coverage

D2.2 describes the eDIANA ontology, which aims at defining the universe of concepts or classes and their relations in the domain of eDIANA Platform Architecture, related to device awareness. The document first analyses middleware that considers device awareness in its specification (OSGi, KNX, ZigBee- Home Automation Public Application Profile and CORBA), then analyses different ontology languages and also describes the methodology that has been used to create the ontology. Section 5 presents the semantic structure of the ontology. The document defines three layers and for each layer a taxonomy in OWL is shown in a picture. The information and service layers ontology is actually only a taxonomy of classes. No object properties, data properties or restrictions are specified in this document. Some properties and restrictions are defined for the device layer ontology making it a proper ontology. The semantic coverage is depicted in Figure 4.

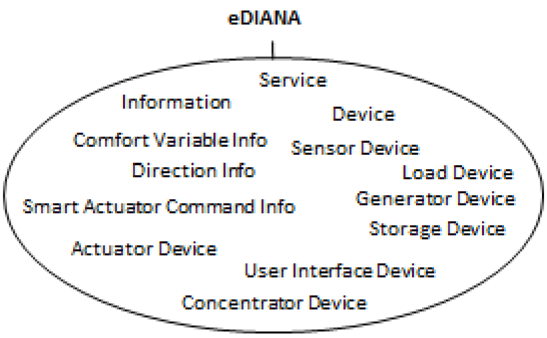


Figure 4: Visual representation of the semantic coverage of eDIANA

[13] http://s15723044.onlinehome-server.info/artemise/index.php
[14] http://www.artemis-project.eu

The three layers in the ontology are defined as information, services and devices layers. The information layer contains the different categories of information that will be referenced by the elements defined in the services layer and devices layer. They include Direction_Information, Comfort_Variable_Information (such as Humidity_Information, Luminosity_Information, Noise_Information, and Temperature_Information), and Smart_Actuator_Command_Information (such as Change_Configuration_Command_Information, Delayed_Turn_Off_Command_Information, Delayed_Turn_On_Command_Information, and Turn_Off_Command_Information, Turn_On_Command_Information).

The services layer specifies the different interfaces at a very high level (the concrete definition of the interfaces is recommended as future work in the document), including External_Services and Internal_Services. The devices layer contains different categories of devices that compose the eDIANA platform to enable device awareness services and plug-and-play services by characterizing the devices, their properties and their interfaces. They include Concentrator, Actuator, Appliance (including Generator, Load, Storage), Sensor (including Video Camera, Airflow Sensor, Gas Sensor, Humidity Sensor, Light Sensor, Power Sensor, Sound Sensor, Sun Radiation Sensor, Temperature Sensor, Fire Sensor, Movement Sensor, and Smoke Sensor), and User Interface.

Overall description of the consensus driven process leading to the model
The majority of the eDiana consortium members are based in Spain, including Acciona Infrastructures, Atos Origin, Tecnalia, Fagor, Ikerlan, I&IMS, Gaia, and Mondragon University. Other partners include STM, Philips, Elsag, Fideliz, Quintor, Infineon, VTT, Bologna University, and Sapienza University Rome. The authors of the ontology are from ESI Tecnalia and I&IMS.

3.3.6 EnOcean

Model Acronym and Full Name
EnOcean Alliance Equipment Profiles

Most relevant URLs, and other precise references
EnOcean Equipment Profiles (EEP), Version 2.6, 17 December 2013, http://www.enocean-alliance.org/eep/.

Overall description
EnOcean[15] is a company that develops energy harvesting wireless sensors which are claimed to be maintenance free and flexible allowing cost reduction in buildings and industrial facilities. They founded the EnOcean Alliance[16], which develops and promotes self-powered wireless monitoring and control systems for sustainable buildings by formalizing an interoperable wireless communication technology. In 2012 this technology has subsequently been standardized as ISO/IEC 14543-3-10 [12]. The standard covers the OSI (Open Systems Interconnection) layers 1-3 which are the physical, data link and networking layers, and is geared to wireless sensors and wireless sensor networks with ultra-low power consumption. It also includes sensor networks that utilize energy harvesting technology to draw energy from their surroundings – for example from motion, light or temperature differences. This principle enables electronic control systems to be used that work independently of an external

[15] www.enocean.com
[16] www.enocean-alliance.org

power supply. Full interoperability is guaranteed together with the EnOcean Equipment Profiles (EEPs) drawn up by the EnOcean Alliance.

Description of the semantic coverage

The EnOcean Equipment Profile (EEP) contains information about devices "enabled by EnOcean", including RORG (identifies the EnOcean Radio Protocol (ERP) radio telegram type), FUNC (identifies the basic functionality of the data content), and TYPE (identifies the type of device in its individual characteristics). The semantic coverage is depicted in Figure 5.

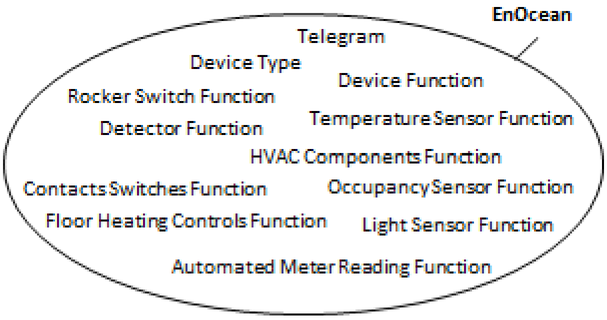


Figure 5: Visual representation of the semantic coverage of EnOcean

There are 4 types of Telegrams (RPS, 1BS, 4BS, VLD) and for each of them there are several corresponding devices functions and types.

The RPS telegram contains the following device functions: Rocker Switch, which has several channels and states, and can be further classified in 2 Rocker or 4 Rocker, Position Switch Home and Office Application, Detectors, and Mechanical Handle. Each of these functions is further divided in device types, for example, the Rocker Switch – 2 rockers function has types "01 Light and Blind Control – Application Style 1", "02 Light and Blind Control – Application Style 2", "03 Light and Blind Control – Application Style 3" and "04 Light and Blind Control ERP 2".

The 1BS telegram contains only one function and type, namely the Contacts and Switches device function with type "01 Single Input contact".

The 4BS telegram contains the following device functions: Temperature Sensors, which is further classified in types depending on the range of temperature handled, Temperature and Humidity Sensor, Light Sensor, Occupancy Sensor, Light-Temperature-Occupancy Sensor, Gas Sensor, Room Operating Panel, Controller Status with types Light controller, Temperature Controller Output, Blind Status and Extended light status, Automated meter reading (AMR) with types Counter, Electricity, Gas and Water, Environmental Applications with types Weather station, Sun Intensity, Date exchange, Time and Day exchange, Geographic position exchange, sun position and radiation, Multi-Func Sensor, HVAC components, Digital Input, energy management, Central command, Universal.

The VLD telegram contains the following device functions: Electronic switches and dimmers with energy measurement and local control, Sensors for temperature-illumination-occupancy and smoke, Light Switching + Blind Control, CO2-Humidity-Temperature-Day/Night and Autonomy, Fan Control, Floor heating controls and automated meter reading, Automated reading meter gateway, Standard valve.

Overall description of the consensus driven process leading to the model
The EnOcean Alliance is founded in 2008 and includes over 250 members and aims to create interoperability between the OEM partners of the EnOcean technology. The Alliance has 9 so-called promotor members which besides EnOcean include BSC Computer GmbH, Honeywell, OPUS greenNet, Pressac Communications, ROHM, Texas Instruments, Thermokon Sensortechnik, and Verve Living Systems.

3.3.7 FAN

Model Acronym and Full Name
FAN (FlexiblePower Alliance Network) FPAI (Flexible Power Application Infrastructure)

Most relevant URLs, and other precise references
HEGRID AD1305 Interface description: Interface report, Version 1.0 (final), 7th January, 2014, http://www.flexiblepower.org/downloads/ (after free registration).

Overall description
FAN[17] (The Flexiblepower Alliance Network) is a network of companies and institutions that jointly develop and manage the international FAN standard. The alliance assures the quality of the standard and monitors compatibility of devices and services that are FAN-labeled. FAN is an independent foundation. FAN develops and maintains a standard for the communication layer between devices and energy services. On the one hand, appliances indicate the minimum amount of energy they need to operate properly. On the other hand, energy services can work out when the circumstances are ideal for energy use (depending on e.g. the weather forecast and energy price). The FPAI (Flexible Power Application Infrastructure) framework can be used to flexibly support different Supply and Demand Management approaches towards end-customers. With the FPAI framework the household is managed via an FP Home Box. This box will be responsible for the negotiation between the energy service providers and the household and the coordination and management of energy resources located within the household. The FPAI application is implemented on top of the OSGi platform.

Description of the semantic coverage
The semantic coverage of FAN is depicted in Figure 6 and consists of the following main concepts:

- A Device represents a functional (hardware) component that consumes, produces, releases, or converts electricity or physical substances and that has some flexibility in its energy usage and can therefore be energy-managed;
- Resources represent devices within a household or a building that can provide flexibility with regard to consumption, storage and production of energy;
- Device manager, or resource manager, describes the energetic flexibility of a device in a generic and standardized way. An energy app is only interested in exploiting energetic flexibility and not in the specifics of a washing machine, for instance. The energetic flexibility is expressed in so called Control Spaces. There are four different types of Control Spaces that cover most appliances:
  - Time shifters are resources that can shift the generation or usage of energy over a specific period of time. Examples are washing machines with a possibility to postpone the start time. Parameters in the Control Space of a Time shifter are: an energy profile

[17] www.flexiblepower.org

and a period over which the start moment can be shifted. Another example is an electric car which needs to be charged before a certain moment (before it will be used) but the actual loading can be performed and shifted within a certain time period;
  - Buffers are resources that can temporarily consume (or produce) more energy so they will use (or generate) less energy at a later moment in time. In most cases these are thermic buffers such as heating devices or refrigerators. Examples of parameters for the Buffer Control Space are: total buffer capacity, filling, loading curve and discharge curve;
  - Storage are resources that resemble a Buffer Resource, but a Storage resource can both store and return energy. Apart from parameters describing the Buffer resource, a Storage Control Space also has parameters for storage loss;
  - Uncontrolled load/generation are type of Resources whose energy behaviour cannot be controlled (e.g. solar panels, TV, computers, etc.). For this resources only a prediction can be made for the expected consumption or production of energy. These predictions can be used in the rest of the framework to make decisions on energy control;
- An energy app receives the control spaces and decides how to exploit the energetic flexibility. As a response to a control space an energy app will send an allocation. The allocation simply contains the energy profile that a resource will have to follow. An allocation should always respect the constraints that were expressed in the control space.

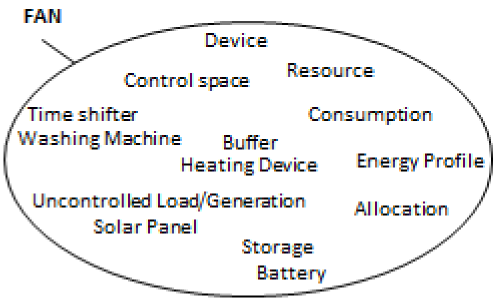


Figure 6: Visual representation of the semantic coverage of FAN

Overall description of the consensus driven process leading to the model

The FlexiblePower Alliance Network is founded in 2013 and includes TNO and Alliander. FAN aims to develop a worldwide standard to help households, businesses, manufacturers, energy companies and software suppliers to accelerate the future of sustainable energy together. FAN has participants and platinum participants. Decisions are made by the general assembly by majority of votes. Every participant has one vote and every platinum participant has five votes in the general assembly.

3.3.8 FIEMSER

Model Acronym and Full Name

FIEMSER (Friendly Intelligent Energy Management Systems in Residential Buildings) Data Model

Most relevant URLs, and other precise references

D5 FIEMSER Data Model, February 2011, http://www.fiemser.eu/wp-content/uploads/2011/12/D5_FIEMSER-data-model_m9_CSTmb_REVIEW.pdf.

Overall description

FIEMSER ran between 2010 and 2013. The main objective of this FP7 European R&D project[18] was the development of an innovative energy management system for existing and new residential buildings, which pursues the increase of the efficiency of the energy used and the reduction of the global energy demand of the building, but without penalizing the comfort levels of the users. To the achievement of this goal, it followed two main strategies:

- Minimizing the energy demand from external resources, through the reduction of the energy consumption in the building and the correct management of local generation (heat and electricity) and energy storage equipment to satisfy the energy demand of the building, and even provide the capability to export energy to the utilities when needed.
- Interaction with the building user, in such a way as to increase the consciousness of the consumer of his energy consumption and CO_2 emissions, providing hints to make punctual changes in his behaviour without major disruptions of his comfort conditions.

To specify this energy management system, the project defined and published a system architecture (D4) [13], a data model (D5), and interface modules (D9) [14]. D5 describes the methodology used to develop a data model for Friendly Intelligent Energy Management System for Existing Residential Buildings (FIEMSER) and specifies the data model itself. The methodology followed a bottom-up approach that started from 8 specific sub-models that were afterwards integrated in a single data model. The several sub-models are described using natural language and UML diagrams.

Description of the semantic coverage

The semantic coverage of FIEMSER is depicted in Figure 7. The specific sub-models used to create the FIEMSER data model belong to the following 8 corresponding categories of data: Environmental and Contextual data (ENV), Energy-focused Building Information Model (BIM), Data from sensors (WSN), User Preferences (USR), Resources scheduling data (SCH), Advices (ADV), Energy Performance Indicators (EPI), and User access right (RGH).

The ENV data model represents the environment of the building, such as the climate, location, orientation, and economical environment (prices). Main classes are: Weather Forecast class and Day Ahead Prices class to describe the hourly evolution of, respectively, weather and energy prices, during a certain period of time.

The Energy-focused BIM data model describes the building space organization and the building resources (loads that consume energy to offer a service to the user, generators that provide part of the energy required by the building, storage devices to provide convenient energy management strategy).

[18] www.fiemser.eu

The WSN data models represents the Control device class, i.e., the devices that can be directly controlled and monitored by FIEMSER, which can be interfaced with Hardware Components (sensors or actuators), and handles a number of software and network protocols. The energy consumption of each control device is maintained and estimated.

The USR data model represents the daily planning of the building usage by the end-user using home usage profiles at the level of building zones.

The SCH data model represents the overall building energy usage planning and the individual use of resources. The main classes are: building program schedule class, temperature schedule class, resource schedule class.

The ADV data model represents the advices given to the user to improve the energy efficiency of the building (Order class, Advice class and Control Action class).

The EPI data model represents the control devices, the link with the equipment they operate upon, and the details of operations of sensors and actuators (DataLog class, Sensor class, Actuator class, Home Daily MeasurementLog class).

The RGH data model represents the (groups of) users and their different permissions to access and operate upon the FIEMSER system.

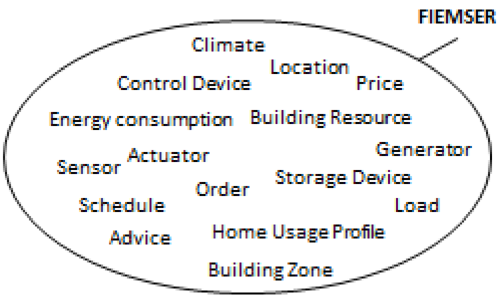


Figure 7: Visual representation of the semantic coverage of FIEMSER

Overall description of the consensus driven process leading to the model
The consortium is mainly based in France and Spain, and includes Tecnalia, Centre Scientifique et Technique du Bâtiment (CSTB), Fraunhofer Institute for Building Physics (IBP), University College Dublin, Acciona Infraestructuras SA, Tenesol SA, TP Vision, and Thales Communications SA. D5 was edited by CSTB, with input from Tecnalia, UC Dublin, Fraunhofer, and Tenesol.

3.3.9 FIPA

Model Acronym and Full Name

FIPA (Foundation for Intelligent Physical Agents) Device Ontology Specification

Most relevant URLs, and other precise references

FIPA Device Ontology Specification, SC00091E, 3 December 2002,
http://www.fipa.org/specs/fipa00091/SI00091E.pdf or
http://www.fipa.org/specs/fipa00091/SI00091E.html.

Overall description

FIPA[19] is an IEEE[20] Computer Society standards organization that was formed in 1996 and is dedicated to promoting the industry of intelligent agents by openly developing specifications supporting interoperability among agents and agent based applications. FIPA Technical Committees (TCs) are intended to carry out the technical work of FIPA. Currently, the following TCs are tasked with work:

- Ad-Hoc
- Methodology
- Modeling
- Security
- Semantics
- Services

In 2002, the then existing FIPA Gateways TC published an ontology for describing devices and their properties. The FIPA Device Ontology Specification describes a device ontology that aims at enabling interoperability between software agents. The FIPA device ontology can be used by agents when communicating about devices.

Description of the semantic coverage

Two devices may exchange device profiles (either directly or through a brokering agency) and acquire a list of services provided by the other device. The list of services may include both hardware and software services, for example, a software component that provides access to a hardware component of the device (such as microphone, headset or GPS service). The profile needs to support the identification of services for various input and output capabilities, such as audio input and output. Agents pass profiles of devices to each other and validate them against the fipa-device ontology. For example, an agent can ask another agent whether a certain device has enough capabilities to handle some task. The classes represented in the FIPA device ontology are agent, device, profile, info-description, hardware-description, connection-description, user interface-description, screen-description, resolution-description, memory-description, memory-type-description, software-description. The semantic coverage is shown in Figure 8.

[19] www.fipa.org
[20] www.ieee.org

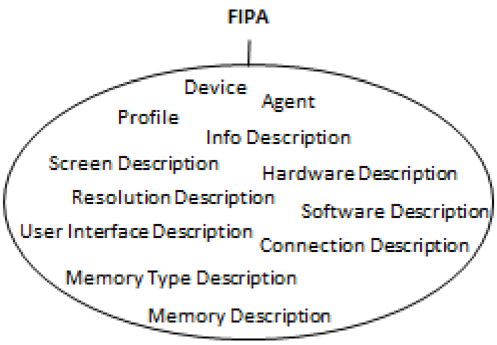


Figure 8: Visual representation of the semantic coverage of FIPA

Overall description of the consensus driven process leading to the model
FIPA is an open organization with free membership. When the standard was published, FIPA consisted of 56 member organizations and companies. The Gateways TC was founded in 2000 and supported by BT, EPFL, Nokia, Siemens AG, Sonera Ltd, and University of Helsinki.

3.3.10 HYDRA

Model Acronym and Full Name
HYDRA (Heterogeneous physical devices in a distributed architecture) MDA Design Document

Most relevant URLs, and other precise references
Deliverable D6.6 Updated MDA Design Document, version 1.0, 20 August 2009, http://www.hydramiddleware.eu/hydra_documents/D6.6_Updated_MDA_Design_Document.pdf.

Overall description
HYDRA[21] was a European 6th Framework R&D project running from 2006-2010. The first objective of the Hydra project was to develop middleware based on a Service-oriented Architecture, to which the underlying communication layer is transparent. The middleware will include support for distributed as well as centralised architectures, security and trust, reflective properties and model-driven development of applications. The HYDRA middleware is designed to be deployable on both new and existing networks of distributed wireless and wired devices, which operate with limited resources in terms of computing power, energy and memory usage. It says to allow for secure, trustworthy, and fault tolerant applications through the use of distributed security and social trust components. The embedded and mobile Service-oriented Architecture is expected to provide interoperable access to data, information and knowledge across heterogeneous platforms, including web services, and support true ambient intelligence for ubiquitous networked devices.

The second objective of the HYDRA project was to develop a Software Development Kit (SDK) to be used by developers to develop innovative Model-Driven applications. The middleware developed within HYDRA has been open-sourced as LinkSmart[22], allowing developers to incorporate heterogeneous physical devices into their applications through easy-to-use web services for controlling any device. LinkSmart is still actively developed and released on 20 March 2014 its version 2.1.

---
21 www.hydramiddleware.eu
22 http://sourceforge.net/projects/linksmart

Description of the semantic coverage

D6.6 explains the methodology, architecture and semantic models used in HYDRA. HYDRA aims to interconnect devices, people, terminals, buildings, etc., not only providing interoperability at a syntactic level, but also at a semantic level. This is done by combining the use of ontologies with semantic web services. Hydra relies on semantic descriptions/annotations to expose device capabilities (using ontologies) so that applications can understand these capabilities and use them. There are several ontologies developed in the project, namely:

- The Device ontology
    - Basic device information
    - Device services
    - Device Events
    - Device malfunctions
    - Device capabilities and state machine
- Semantic Discovery ontology
- Semantic Device ontology
- Application Specific ontology

HYDRA has done muchwork on service discovery and composition using ontologies. Most relevant for this study is HYDRA's device ontology, especially the Basic device information module, but also Device services and Device events. The semantic coverage is schematically depicted in Figure 9. It includes concepts such as Hydra application, Semantic Device, Hydra Device, Physical Device, Device Application Catalog, Device services, Device events, Device mulfunctions, etc.

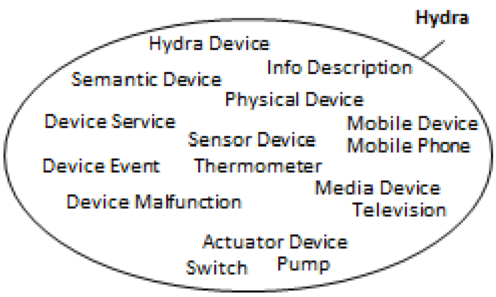


Figure 9: Visual representation of the semantic coverage of Hydra

The Basic device information module represents general device information. The <u>HydraDevice</u> concept presents the main ontology class. The HydraDevice is further subclassed to the model of the <u>PhysicalDevice</u> and the <u>SemanticDevice</u>, which share the common device properties (such as deviceId or location), but have different semantic interpretation and behaviour. The concept <u>InfoDescription</u> contains basic information about device friendly name, manufacturer data (such as manufacturer name and URL) and device model data, namely model name, model description and model number. The information is represented as OWL data type properties. The InfoDescription class is referred from the HydraDevice concept using the <u>info</u> OWL object property. An important part of the basic device information is the representation of <u>device type</u> modeled as the OWL is-a hierarchy by sub classing the PhycicalDevice concept. Further, the OWL object property hasEmbeddedDevice of <u>SemanticDevice</u> concept recursively refers to <u>HydraDevice</u> concept. This property enables the creation of models of composite devices, such as in case of HeatingSystem device used in first system prototype application. HeatingSystem can be, for example, composed of Thermometer and Pump devices.

Overall description of the consensus driven process leading to the model

The partner of the HYDRA consortium include CNet Svenska AB, The Fraunhofer Institute for Applied Information Technology, The Fraunhofer Institute for Secure Information Technology, In-JeT ApS, Priway, T-Connect S.r.l., Telefonica I+D SA, University of Aarhus, Innova S.p.A., University of Reading, Siemens IT Solutions and Services, Technical University of Kosice, and University of Paderborn. The editors of D6.6. are from CNet and Technical University of Kosice.

3.3.11 KNX

Model Acronym and Full Name

KNX Datapoint Types

Most relevant URLs, and other precise references

KNX System Specifications Interworking Datapoint Types, Version 01.09.01, 18 September 2014, http://www.knx.org/media/docs/downloads/03%20-%20KNX%20Standard/KNX%20Standard%20Public%20Documents/03_07_02%20Datapoint%20Types%20v1.07.00%20AS.zip.

Overall description

KNX Association[23] is a non-profit-oriented organisation which members are manufacturers developing devices for several applications for home and building control based on KNX, like lighting control, shutter control, heating, ventilation, air conditioning, energy management, metering, monitoring, alarm/intrusion systems, household appliances, and audio/video. The association is the owner of the KNX standard. It standardizes an OSI-based network communications protocol for intelligent buildings. It defines several physical communication mediums and is designed to be independent of any particular hardware platform. The most common form of installation however is over twisted pair. A KNX Device Network can be controlled by anything from an 8-bit microcontroller to a PC, according to the needs of a particular implementation. Any product labeled with the KNX trademark are tested by KNX accredited third party test labs. During these tests, it is not only checked that the device supports the KNX protocol but that its useful data is coded according to the KNX standardized Datapoint Types. The KNX specifications, known as KNX Handbook and standardized as ISO/IEC 14543-3 [15] and CENELEC EN 50090 [16], are not freely available, but the section that specifies the Datapoint Types is.

Description of the semantic coverage

KNX does not define devices, but it is a data driven standard that defines commands. The parameters in those commands are standardized and called Datapoints. There are several types of Datapoints, namely Datapoint Types for common use, Datapoint Types for HVAC, Datapoint Types for Load Management, Datapoint Types for Lighting, Datapoint Types for Systems. Combinations of data point types into a device are called Functional blocks. Many functional blocks have been standardized, but we take into account here only two functional blocks as example: Dimmer Actuator Basic and

[23] www.knx.org

Sunblind Actuator Basic. A functional block is generally not more than just a selection of datapoint types. Dimmer and Sunblind also contain state, which goes beyond just datapoint types. The semantic coverage of KNX is schematically depicted in Figure 10.

Overall description of the consensus driven process leading to the model

KNX Association, as of 1 March 2014, had 339 members/manufacturers from 37 countries. Companies in the Executive Board are mostly from Germany and include ABB Stotz-Kontakt GmbH, Albrecht Jung GmbH & Co. KG, Busch-Jaeger Elektro GmbH, Feller AG, GIRA Giersiepen GmbH & Co. KG, HAGER Group, Insta Elektro GmbH, Kellendonk Elektronik GmbH, Merten GmbH, Schneider Electric, Siemens AG, Siemens Switzerland Ltd., and Theben AG. Siemens and HAGER deliver the president and the vice-president of the Association. The KNX standard is largely based on the communication stack of EIB (European Installation Bus), which was originally developed by Berker, Gira, Jung, Merten and Siemens AG.

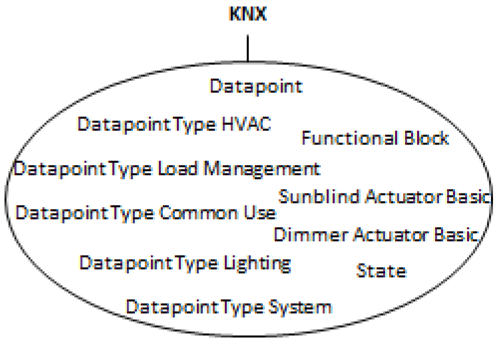


Figure 10: Visual representation of the semantic coverage of KNX

3.3.12 MIRABEL

Model Acronym and Full Name

MIRABEL (Micro-Request-Based Aggregation, Forecasting and Scheduling of Energy Demand, Supply and Distribution)

Most relevant URLs, and other precise references

D7.5 “MIRABEL-ONE: Initial draft of the MIRABEL Standard, version1.0”, 22 December 2011, http://wwwdb.inf.tu-dresden.de/miracle/publications/D7.5.pdf.

Overall description

MIRABEL[24] was a European 7th Framework R&D project running from 2010-2013. Its main goal was to develop an approach on a conceptual and an infrastructural level that allows energy distribution companies to balance the available supply of renewable energy sources and the current demand in ad-hoc fashion. MIRABEL worked on a concept of micro-requests with time shifts to handle the demand and supply of energy on a household level. Further, they defined methods to predict the energy supply and demand in the small (i.e., for households) and in the large and to update predictions over time. The idea is then to aggregate (and disaggregate) the micro-requests on a regional level, and to develop a scheduling approach for energy production and consumption based on aggregated requests. Energy distribution companies may use the aggregated request information to re-schedule energy demands/supplies and thus have additional means to react to shortages or an

[24] www.mirabel-project.eu

abundance of energy. They may also trade their demand requests with other energy distribution companies.

Description of the semantic coverage

The goal of D7.5 is to define a specification for modeling flexibility and the exchange of flexibility information between stakeholders in the energy domain, especially between consumers and electricity suppliers. The specification is described in terms of a generic data model for energy flexibility and messages for information exchange on flexibility offerings. The intention of this specification is to be used as input for formal European standardization and acceptance in the electricity market. The document presents detailed data models and messages. Its coverage is depicted in Figure 11.

The ontology consists of several concepts, some specific to the energy/smart grid domain, others more general, like time and price. The ontology describes how actors can express their energy flexibility for a specific device with respect to amount, time and price in user preferences. Each device has an energy profile that describes how much (amount) energy the device consumers and or produces over a time span. A FlexOffer will be issued by an actor. It combines the user preferences and the corresponding device energy profile.

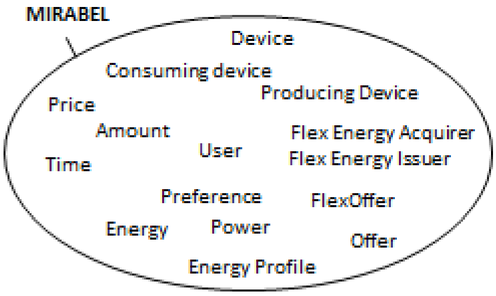


Figure 11: Visual representation of the semantic coverage of MIRABEL

Overall description of the consensus driven process leading to the model

The MIRABEL Consortium consists of six research and technology partners and two use-case partners: SAP AG, Aalborg Universitet, CRES, Energie Baden-Württemberg, INEA, the Josef Stefan Institute, the Technische Universität Dresden, and TNO. D7.5 is edited by TNO.

3.3.13 OMA Lightweight M2M

Model Acronym and Full Name

OMA (Open Mobile Alliance) Lightweight M2M (Machine-to-Machine) Technical Specification

Most relevant URLs, and other precise references

OMA Lightweight Machine-to-Machine Technical Specification Candidate version 1.0, 10 December 2013, http://technical.openmobilealliance.org/Technical/release_program/docs/LightweightM2M/V1_0-20131210-C/OMA-TS-LightweightM2M-V1_0-20131210-C.pdf.

Overall description

OMA[25] was formed in June 2002 by world-wide mobile operators, device and network suppliers, information technology companies and content and service providers. OMA delivers open specifications for creating interoperable services that work across all geographical boundaries, on any bearer network. OMA's specifications support the billions of new and existing fixed and mobile terminals across a variety of mobile networks, including traditional cellular operator networks and emerging networks supporting machine-to-machine device communication. OMA has developed and is developing a number of standards for managing light weight and low capability devices on a variety of networks

Description of the semantic coverage

The OMA Lightweight M2M Technical Specification specifies the Lightweight M2M protocol between the server and the client that resides in a device. The target devices are resource constrained devices. The document also specifies the core set of LightweightM2M Objects. Each piece of information made available by the client is a resource, a client may have any number of resources, and these resources are organized into objects. Each object and resource supports one or more operations. There are reusable resources that are common to several objects. The objects described are: LWM2M Security, LWM2M Server, Access Control, Device, Connectivity Monitoring, Firmware, Location, and Connectivity Statistics. Each object is presented with all its resources. The XML schemas of the objects with their resources are available.

The semantic coverage is depicted in Figure 12. Besides client and server we can derive from the technical specification document: objects, resources, operations, instances, LWM2M Server object, Access Control object, Device object, Connectivity Monitoring object, Firmware object, Location object, Connectivity Statistics object. Examples of resources for the Device object are: manufacturer, modal number, serial number, firmware version, reboot, factory reset, available power sources, power source voltage, power source current, and battery level, among others.

[25] http://openmobilealliance.org

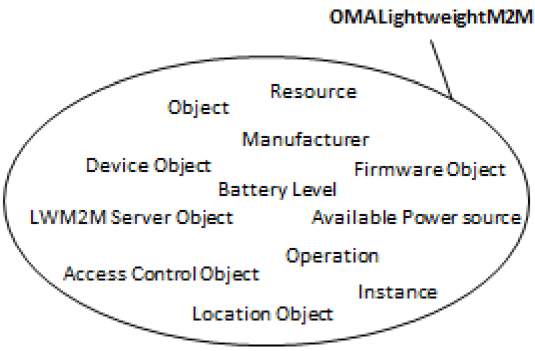


Figure 12: Visual representation of the semantic coverage of OMALightweightM2M

Overall description of the consensus driven process leading to the model

OMA has 104 members, of which 8 so-called promoter members: AT&T, Microsoft, Orange SA, BlackBerry Limited, Motorola Solutions Inc., Qualcomm Inc, Intel Corporation, and NTT DOCOMO INC. Members particularly supporting the Lightweight M2M specifications are China Mobile, China Unicom, Huawei, Intel, and ZTE.

3.3.14 OMS

Model Acronym and Full Name

OMS (Open Metering System)

Most relevant URLs, and other precise references

Open Metering System Specification Vol.2 – Primary Communication Issue 4.0.2, and OMS-Data Point List –RELEASE A, Annex B to Volume 2: Primary Communication Issue 4.0.2, 27 January 2014, http://oms-group.org/fileadmin/pdf/OMS-Spec_Vol2_Primary_v402.pdf and http://oms-group.org/fileadmin/pdf/OMS-Spec_Vol2_AnnexB_A031.pdf.

Overall description

The Open Metering System (OMS) specification is defined by the OMS Group [26], and focuses on an automatic meter readout system. Part of this system is a hardware system (a Multi Utility Communication, or MUC) which is used to readout different metering devices and to transfer subsets of this data to automated meter management (AMM) back office systems for billing, servicing or other purposes. Metering devices are sensors and actuators. Metering devices and AMM Systems have to follow certain protocols which are described within the specification. Communication between the meter and the MUC is called Primary Communication and can be based on DLMS/COSEM (see section 3.21.9), SML (Smart Message Language [17], a German specification), or M-Bus. M-Bus is a significant communication technology for remote reading of meters in Europe, and standardized as EN13757-x, „Communication System for Meters and Remote Reading of Meters" [18]. The standard defines wired and wireless remote reading of meters. The wireless variety (W-Mbus) is also part of the KNX standard (see section 3.9). OMS Specification Vol.2 – Primary Communication Issue 4.0.2 defines the OMS Application Protocols for Primary Communication. For M-Bus it just cites the M-Bus Application Protocol as described in EN 13757-3:2013, and restricts it by some additional rules to ensure interoperability. This includes a list of mandated and harmonized M-Bus Data Points as given by the separate Annex B document.

[26] www.oms-group.org

Description of the semantic coverage

The semantic coverage is shown in Figure 13. Metering devices are sensors and actuators. Sensors are metering devices which at least provide meter index data (current metering counter value). Basic meter are meters with minimal functionality. Current metering data are given by request or sent in regular intervals. Sophisticated meters are basic metering devices with additional features such as data logging. The metering data given by these devices could include timestamps and metering profiles of the recorded consumption data. Actuators are appliances which can limit consumption or cut-off the supply. Terms which are included in the term 'actuator' are breaker, limiter, shut-off-valve, gas valve or switch. Multi Utility Communication Controller (MUC) is a hardware system which is used to readout different metering devices and to transfer subsets of this data to AMM back office systems for billing, servicing or other purposes. Metering devices and AMM Systems have to follow certain protocols. The AMM back office system maintains a connection to several MUCs. The MUCs themselves keep the connection to several meters.

The *Open Metering System Specification- Vol.2-Primary Communication_version4.0.2* provides a classification of smart meters and other devices addressed by OMS and can be used to make a taxonomy for smart meters.

– Device Types of OMS-Meter: Electricity meter, Gas meter, Heat meter, Warm water meter (30°C ... 90°C), Water meter, Heat Cost Allocator, Cooling meter (Volume measured at return temperature: outlet), Cooling meter (Volume measured at flow temperature: inlet), Heat meter (Volume measured at flow temperature: inlet), Combined Heat / Cooling meter, Hot water meter (≥ 90°C), Cold water meter, Waste water meter.
– Device Types of other OMS-devices: Breaker (electricity), Valve (gas or water), Customer unit (display device), Communication controller, Unidirectional repeater, Bidirectional repeater, Radio converter (system side), Radio converter (meter side).
– Device Types of not certifiable device: Oil meter, Steam meter, etc. (See table 4, Open Metering System Specification- Vol.2-Primary Communication_version4.0.2)

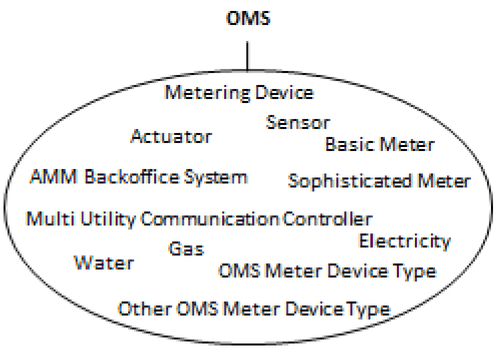


Figure 13: Visual representation of the semantic coverage of OMS

Overall description of the consensus driven process leading to the model

The vendor associations Figawa[27] and KNX are chairs in this specification creation process. Figawa is the German trade organization for water, gas, and pipeline companies. KNX Association is described in section 3.9. The goal is to guarantee a future-proof communication standard and interoperability between all the meter products: gas, water, electricity, heating.

[27] www.figawa.org

3.3.15 OSGi DAL

Model Acronym and Full Name

OSGi DAL (Device Abstraction Layer)

Most relevant URLs, and other precise references

RFC-196 OSGi Alliance Device Abstraction Layer, Draft, 30 January 2014, https://github.com/osgi/design/blob/a71f2871f4ed0b97c4da79cf756a15876a61a347/rfcs/rfc0196/rfc-0196-DeviceAbstractionLayer.pdf?raw=true. This is the 9th draft of the document and it is the one we analysed. In February 2014, a new draft was published under https://github.com/osgi/design/blob/master/rfcs/rfc0196/rfc-0196-DeviceAbstractionLayer.pdf. The basic changes are: Basic device functions are moved to another RFC document, DeviceFunction renamed to Function, DeviceFunctionEvent renamed to FunctionEvent, DeviceFunctionData renamed to FunctionData.

Overall description

The OSGi Alliance[28] is a worldwide consortium founded in 1999 to create open specifications that enable the modular assembly of software built with Java technology. The OSGi technology facilitates the componentization of software modules and applications and assures remote management and interoperability of applications and services over a broad variety of devices. The alliance provides specifications, reference implementations, test suites and certification to foster a valuable cross-industry ecosystem. OSGi specifications define a dynamic component system for Java. These specifications reduce software complexity by providing a modular architecture for large-scale distributed systems as well as small, embedded applications.

Description of the semantic coverage

The OSGi Device Abstraction Layer document specifies a reference architecture that introduces an abstraction layer to allow the decoupling of devices and services from specific protocols. This architecture is based on a service registry in which services and device functions are registered. Section 5 of the document ("Technical solutions") provides a detailed specification of the entities and properties involved in the architecture.

The semantic coverage is depicted inFigure 14. The concept of device is central in the OSGi architecture. Devices can play different roles in their networks as events reporters, controllers etc. The dynamic behavior of a device can be mapped to the dynamic OSGi service registry, where it is registered as a device service. Device services realize a basic set of operations and provide a set of properties. Applications running in the architecture are allowed to track the device status, to read descriptive information about the device and to follow the device relations. A set of functions can belong to the device and can be found in the OSGi service registry. These functions represent the device operations and related properties in a modular way. Applications are allowed to get directly the required functions if they do not need information about the device. For example, light device is registered as a device service and there is a device function to turn on and turn off the light.

[28] www.osgi.org

Overall description of the consensus driven process leading to the model

The OSGi Alliance consists of about 30 membership paying companies and over 100 supporter companies. RFC-196 is edited by ProSyst Software. France Telecom, Telekom Italia, Deutsche Telekom, NTT, ProSyst, Makewave, Oracle, IBM, EnOcean, Hitachi, IS2T, NEC, Paremus, and invited researchers are member of the relevant working group within the OSGi Alliance. As part of the development process non-OSGI-members can access the specification draft and provide comments under the OSGi IPR policy. The final specification needs to be approved by all paying members of the OSGi Alliance. OSGi Alliance expects it to become final in Q3, 2014.

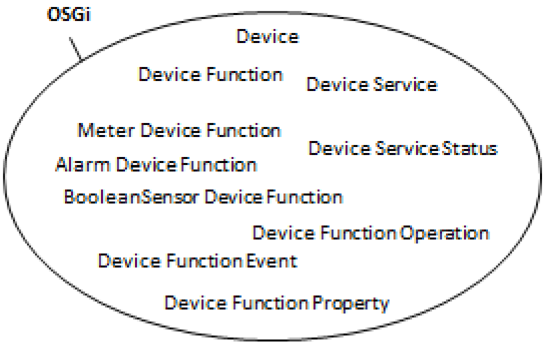


Figure 14: Visual representation of the semantic coverage of OSGi DAL

3.3.16 SEEMPubs

Model Acronym and Full Name

SEEMPubS (Smart Energy Efficient Middleware for Public Space) Data Format Definition

Most relevant URLs, and other precise references

Deliverable D5.1 “Data Format Definition, version 1.0”, 30 September 2012, http://seempubs.polito.it/images/stories/documents/WP5/D.5.1.pdf.

Overall description

SEEMPubs[29] was a European 7th Framework R&D project running from 2010-2013. Its goal was to provide control of appliances to effortlessly optimise energy efficiency usage without compromising comfort or convenience and offering decision makers strategies and tools needed to plan energy saving measures. SEEMPubS makes use of the service-oriented middleware for embedded systems being developed in the HYDRA project and uses its huge potential to create services and applications across heterogeneous devices to develop an energy-aware platform. The SEEMPubS platform is developed to provide necessary functionality and tools to add energy efficiency features to monitor dynamic sensor data in real time, taking advantage of natural resources (like daylight and solar energy) and controlling the operation of both passive and active environmental systems to ensure the best possible comfort conditions with the most efficient use of energy. Another European R&D project (ARTEMIS framework) that used a very similar approach is ME3Gas (Middleware for Energy Efficient Embedded Services & Smart Gas Meters)[30].

[29] http://seemspub.polito.it
[30] www.me3gas.eu

Description of the semantic coverage

D5.1 summarizes the architecture used in the SEEMPubS project to control building services and monitor indoor comfort conditions, electric and thermal energy consumption in a room. Two architectures are designed in order to take into account possible different types of buildings to be monitored. The first architecture is a wireless control architecture, mainly used in the Valentino Castle due to its architectural constraints (paintings, stuccos and historical structures) that did not allow destructive interventions on the buildings component. The second architecture is a wired control architecture that was mainly adopted in the modern parts of the Politecnico di Torino Campus, together with some wireless devices to monitor the electric energy consumptions of lights or other equipment (PC, printers, etc.) or to collect more detailed data of indoor air temperature distribution. Section 3 of D5.1 describes the data that are recorded from the hardware devices and processed through the SEEMPubS platform. The description categorizes the data in data related to indoor comfort conditions, energy consumptions, and spaces and building services usage. These data are represented in the deliverable using tables.

The semantic coverage is shown in Figure 15 and includes the concepts of Sensor/Device, such as Light sensor , Indoor air temperature sensor, Wireless sensors for indoor air temperature and relative humidity, Sensor of supplied air temperature, Controller, Power meter for lighting systems, Power meter for appliances, Occupancy sensor, Switch, Outdoor air temperature sensor, Indoor air temperature sensor (thermostat); Sensor number and position, e.g., Ceiling, Wall, Work plane; Measured quantity, e.g., Illuminance (lux), Indoor Air Temperature (°C), angle of rotation for manual control (°), Indoor Relative Humidity (%); Timing of data communication; Corresponding communication protocol, e.g., EnOcean, BACNet, IEEE802.15.4; Basic data processing, e.g., Average values over 15 minutes, Average hourly value; Basic data representation, e.g., Daily values, Weekly values, Monthly values, Seasonal trend, Annual values (carpet plot), Frequency distribution, and Cumulative frequency.

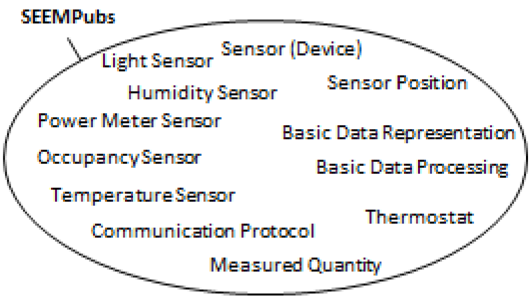


Figure 15: Visual representation of the semantic coverage of SEEMPubs

Overall description of the consensus driven process leading to the model

Project partners are Politecnico di Torino, STMicroelectronics, Centro Ricerche Fiat, Fraunhofer-FIT, CNet Svenska AB, Katholieke Universiteit Leuven, Universite Claude Bernard Lyon 1, Sinovia SA, Istituto Superiore Mario Boella, ENI Servizi. D5.1 was edited by Politecnico di Torino together with Unversite Claude Bernard Lyon 1.

3.3.17 SEIPF (now called PowerOnt)

Model Acronym and Full Name

SEIPF (Semantic Energy Information Publishing Framework)

Most relevant URLs, and other precise references

Dario Bonino, Fulvio Corno, Faisal Razzak "Enabling Machine Understandable Exchange of Energy Consumption Information in Smart Environments", Energy and Buildings 43 (2011) 1392–1402, http://dx.doi.org/10.1016/j.enbuild.2011.01.013.

Overall description

The goal of the Bonino *et al* paper is enabling residential gateways to provide energy consumption information and other properties for different appliances in the house in a machine understandable format by using a Semantic Energy Information Publishing Framework (SEIPF). This information is published according to Semantic Web standards and best practices. Appliance properties are exposed according to the existing semantic modeling supported by home gateways, power consumption is modeled by introducing a new modular Energy Profile (EP) ontology. The SEIPF framework is able to expose data both as simple RDF triples (according to Linked Data requirements) and as full ontology instances, for the benefits of applications needing intelligent processing.

Description of the semantic coverage

The goal of the Semantic Energy Information Publishing Framework (SEIPF) is enabling residential gateways to provide energy consumption information and other properties for different appliances in the house. This information is published in a machine understandable format according to Semantic Web standards and best practices. Appliance properties are represented using existing semantic models supported by home gateways, while the power consumption is modeled by introducing the Energy Profile (EP) ontology. The EP ontology models the energy consumption information about the appliance using the underlying DogOnt ontology [31], which models the domotic system of a house supporting intelligent operations. The paper claims that the EP ontology is published according to Semantic Web practices and an OWL version is available, but it does not provide a URL.

The semantic coverage of the EP Ontology is depicted in Figure 16. Basic concepts of this ontology are Device Profile, which describes the energy profiles of all the major device categories in the house (e.g., TV and dishwasher), and Consumption, which encodes the power consumed by an appliance in a given state. A device profile has some properties, such as the estimated and measured power consumption of a device in a state, and the unit of power for the power consumed by the appliance, expressed as using the Metric Unit class from the Measurement Units Ontology. The Consumption class also has some properties, such as the hasConsumption property that relates the device profiles and consumption instances, and the hasDevice property that relates an instance of device energy profile to a particular device. The EP Ontology is linked to the DogOnt, which has been designed along 7 main classes, corresponding to the Building Thing, Building Environment, Functionality, Command, Domotic Network Component, State, StateValue. Building Environment and Building Things are used in the EP ontology to describe the environment of the house.

[31] http://elite.polito.it/dogont

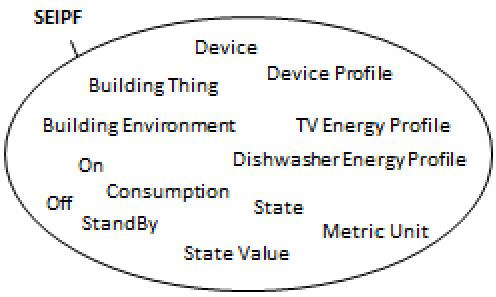


Figure 16: Visual representation of the semantic coverage of SEIPF

Overall description of the consensus driven process leading to the model

This paper reflects the work of a single research group at Politecnico di Torino.

3.3.18 SEP2

Model Acronym and Full Name

SEP2 (Smart Energy Profile 2.0)

Most relevant URLs, and other precise references

Zigbee Alliance / HomePlug Alliance Smart Energy Profile 2 Application Protocol Standard, ZigBee Public Document 13-0200-00, April 2013, http://www.zigbee.org/Standards/ZigBeeSmartEnergy/ZigBeeSmartEnergy20Standard.aspx.

Overall description

Consumers should be able to manage their usage and generation of energy. CSEP[32] (Consortium for SEP2 Interoperability) sees this as a critical feature of the Smart Grid and a basis of innovation for new products and services in energy management. To enable this capability, information flow between devices such as meters, smart appliances, plug-in electric vehicles, energy management systems, and distributed energy resources (including renewable energy and storage elements) must occur in an open, standardized, secure, and interoperable fashion. The SEP2 specification is intended to fulfil those needs. The standard offers IP- based Home Area Network (HAN) energy management functionality and was ratified in April 2013. It addresses the following needs of the market:

- Deployments in multi-dwelling units such as apartment buildings
- Supports multiple Energy Service Interfaces into a single premises
- Control of plug in hybrid electric vehicle (PHEV) charging
- Supports any transport based on IETF IP compliant standards, including but not limited to ZigBee IP, other RF-based and Power Line Carrier (PLC)-based transports
- Supports internationally recognized standards to ensure long-term interoperability with multiple technologies

The document is drafted by the Zigbee Alliance[33] and the HomePlug Alliance[34], which are both member of CSEP, and assures interoperability between ZigBee IP, HomePlug and other IP network technologies that could adopt SEP2. They include the networks supported by the other members of CSEP, the Wi-Fi Alliance and the Bluetooth SIG. The application function sets implemented for SEP 2

[32] www.csep.org
[33] www.zigbee.org
[34] www.homeplug.org

have been mapped to the IEC Common Information Model [19]. Additional data beyond the IEC Common Information Model will be proposed back into IEC. SEP2 has been standardized in 2013 as IEEE 2030.5-2013 [20].

Description of the semantic coverage

This document defines the application protocol used by the Smart Energy Profile release 2.0 (SEP2) and specifies the mechanisms for exchanging application messages, the exact messages exchanged including error messages, and the security features used to protect the application messages. Clients issue requests to all devices on the network requesting resource(s) of interest. Servers hosting the requested resource(s) respond with the information necessary to access the server and its resource(s). The semantic coverage is depicted in Figure 17 and was inferred from the model presented in Section 15- Appendix B of the SEP2 document. XML schemas and UML representations are also available.

The following classes characterize the semantic coverage of SEP 2:

- Commodity Type Object , which has as example values: Electricity secondary metered value, Electricity primary metered value, Air, Natural Gas, Propane, Potable Water, Steam, Waste Water, Heating Fluid, Cooling Fluid;
- Device Category Type Object, which has as example values: Programmable Communicating Thermostat, Strip Heaters, Baseboard Heaters, Water Heater, Pool Pump, Sauna, Hot tub, Smart Appliance, Irrigation Pump, Managed Commercial and Industrial (C&I) Loads, Simple misc. (Residential On/Off) loads, Exterior Lighting, Interior Lighting, Electric Vehicle;
- Service Kind Object , which has as example values: Electricity, Gas, Water, Time, Pressure, Heat, Cooling;
- Unit Type Object , which has as example values: kWh, kW, Watts, Cubic Meters, Cubic Feet, US Gallons, Imperial Gallons, Liters; and Uom Type Object, which has as example values A (Current in Amperes (RMS)), Kelvin (Temperature), Degrees Celsius (Relative temperature), Voltage, J (Energy joule), Hz (Frequency);
- Device Status Object, which has as example values: Not operating, Operating, Starting up, Shutting down, At disconnect level;

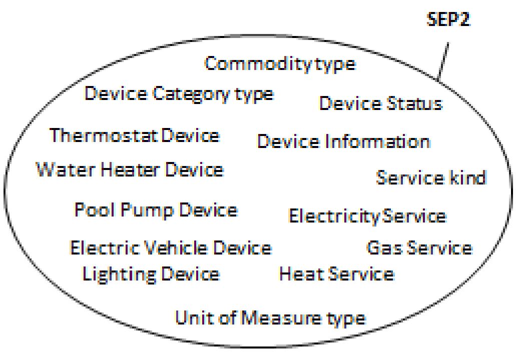


Figure 17: Visual representation of the semantic coverage of SEP2

Device Information Object, which has as example attributes : mfDate attribute (Date/time of manufacture), fHwVer attribute (Manufacturer hardware version), mfID attribute (Manufacturer's IANA Enterprise Number), mfInfo attribute (Manufacturer dependent information related to the manufacture of this device), mfModel attribute (Manufacturer's model number), mfSerNum attribute (Manufacturer assigned serial number), primary Power attribute (Primary source of powe)r, secondary Power attribute (Secondary source of power), swActTime attribute (Activation date/time of currently running software), swVer attribute (Currently running software version).

Overall description of the consensus driven process leading to the model

SEP2 is a joint production of Zigbee Alliance and HomePlug Alliance, although it seems that Zigbee Alliance had a leading role in its production, as they officially ratified it as a Zigbee standard and published it on their website. The Zigbee Alliance has about 150 participant members and a few hundred adopter members. It is governed by 10 promotor members: Comcast, Freescale, Itron, Kroger, Landis+Gyr, Legrand, Philips, Schneider Electric, Silicon Labs and Texas Instruments. The SEP2 standard has many editors and contributors from many different companies.

Not only Zigbee Alliance and HomePlug Alliance are member of CSEP. Also Wi-fi Alliance and Bluetooth SIG are. CSEP is not responsible for drafting and maintaining the standard though. The members of the Consortium for SEP 2 Interoperability are working together to develop common testing documents and processes for certifying SEP 2 interoperability to ensure interoperability of SEP 2 products.

3.3.19 SmartCoDE

Model Acronym and Full Name

SmartCoDE (Smart Control of Demand for Consumption and Supply to enable balanced, energy-positive buildings and neighbourhoods) Model of Local Energy Resource Cluster

Most relevant URLs, and other precise references

Deliverable D1.1.2 “Model of local energy resource cluster”, 31 December 2012, https://www.fp7-smartcode.eu/system/files/page/d-1.1.2.pdf.

Overall description

SmartCoDE[35] is a European 7th Framework R&D project running from 2009-2013. The objective of SmartCoDe is to enable the application of demand side management and smart metering in private and small commercial buildings and neighbourhoods by

- Developing new methods for automated energy management that specifically consider the requirements of Energy using Products in homes / offices and local renewable energy providers such as information security and dependability.
- Developing an inexpensive (<3€) hardware/software implementation that can be integrated into arbitrary Energy using Products, providing them with the ability to communicate and to remotely control its use of power.
- Demonstration of technical and economic feasibility and benefit of intelligent energy management in buildings and neighbourhoods with an initial focus on electric lighting.

[35] www.fp7-smartcode.eu

This project aims at providing a wireless communication infrastructure for the demand side management of energy in the domestic sector. Wireless sensor/actuator nodes are integrated into appliances to enable remote control by an Energy Management Unit (EMU). SmartCoDe builds on the ZigBee wireless standard. D1.1.2 provides an extensive explanation of the SmartCoDE classifications. In addition to the classification of Energy Using Products (EuPs), also a classification of Local Energy Providers (LEPs) into 4 classes is presented.

Description of the semantic coverage

The semantic coverage is depicted in Figure 18 and covers the classification of Energy Using Products (EuPs) that characterize the SmartCoDe project. The Energy Using Products are divided in the following classes:

- SKDSVC class, namely schedulable service, which provides a service that runs for a certain time and can be scheduled within a certain time span. Examples of SKDSVC are washing machine, dryer, dishwasher, baking machine;
- VSTSVC class, namely virtually storable service, which provides an inert service that can serve as a virtual storage. Examples of VSTSVC are Fridge, Freezer, HVAC, Water-boiler;
- VARSVC class, namely variable service, which provides a service that might vary due to user interaction and/or daytime. Examples of VARSVC are lighting controlled by luminance level (e.g. in garden, at entrance), dimmable lighting, blinds;
- ETOSVC class, namely event-timeout controlled service, which provides a service such that the device is switched on and kept on by sensor events, and switched off in absence of sensor event. Examples are lighting controlled by presence detector (e.g. on corridor);
- COMCON class, namely complete control, charging and using up power decoupled; latter only restricted w.r.t. time slots & minimal service. Examples are robot vacuum, robot lawn-mower;
- CHACON class, namely charge control, charging and using up power decoupled; latter is mostly (or solely) user-dependent. Examples are battery & cellphone chargers, hand-held vacuum, emergency backup storages;
- CUSCON class, namely custom control, when the device does not fit into other classes, therefore custom control by user and/or EMU. Examples are HiFi, PC, Oven.

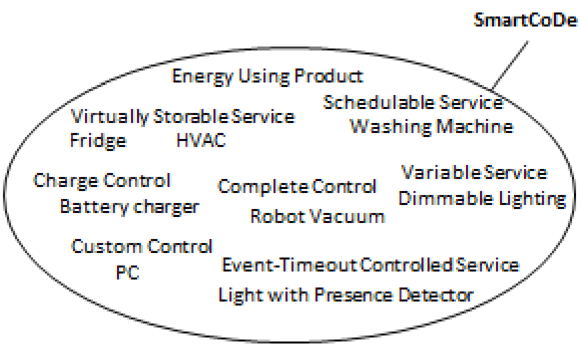


Figure 18: Visual representation of the semantic coverage of SmartCoDe

SmartCoDe also provides a classification of Local Energy Providers (LEPs), which are divided into the ENGRID class, namely energy grid, which is a conventional energy provider, the VOLAEP class, namely volatile energy provider, which is an energy source that depends on weather, day time, etc., the ENSTOR class, namely energy storage, which is an energy source that needs to be charged, and the LENGEN class, namely local energy generator, which is an energy source that transforms fuel to energy.

Overall description of the consensus driven process leading to the model
The partners of SmartCoDE include Edacentrum GmbH, Infineon Technologies Austria AG, Vienna University of Technology, Ennovatis GmbH, Tridonic GmbH & Co KG, Ardaco, a.s., Quiet Revolution Ltd, and University of Novi Sad. The project also has many associate partners, including BSH Bosch und Siemens Hausgeräte GmbH, BuildDesk Austria GmbH, Next Energy e.V., SMA Solar Technology AG, Q.met GmbH and TELEFUNKEN Semiconductors. The editors of D1.1.2 are from TU Vienna, Edacentrum, and Ennovatis.

3.3.20 UPnP

Model Acronym and Full Name
UPnP (Universal Plug and Play) Device Architecture and Home Automation Device Control Protocols (DCPs)

Most relevant URLs, and other precise references
UPnP Device Architecture 1.1, http://upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf
The UPnP Home Automation DCPs consist of 4 Device descriptions with corresponding Service description. 3 Devices are relevant for this project:

- SolarProtectionBlind:1, http://upnp.org/specs/ha/solarprotectionblind1/
- HVAC:1, http://upnp.org/specs/ha/hvac/
- Lighting Controls:1, http://upnp.org/specs/ha/lighting/

Overall description
UPnP[36] is a client/server based interoperability framework for devices and services in a relatively small-scale best-effort IP subnetwork. It distinguishes three logical entities in the network: UPnP Services, which represent the service functionality of a device, UPnP Devices, which act as services servers, and UPnP Control Points (CPs), which act as clients for controlling the services. For clarity, UPnP Devices are not the physical devices but the UPnP server software running on them, providing UPnP Services to UPnP CPs. UPnP defines Simple Service Discovery Protocol (SSDP), SOAP, and General Event Notification Architecture (GENA) for discovery, control, and eventing, respectively. Device and service descriptions are expressed and partially standardized in XML templates, the so-called Device Control Protocols (DCPs). The Device Architecture and many DCPs have been standardized in ISO/IEC as ISO/IEC 29341-x-y [21].

There are DCPs for the following device categories: Audio/Video, Basic, Device Management, Home Automation, Networking, Printer, Remote Access, Remoting, Scanner, Sensor Management, and Telephony. There are also a number of add-on service standardized, such as DataStore:1, DeviceProtection:1, EnergyManagement:1, Low Power:1, ContentSync:1, Device Security:1, Security

[36] www.upnp.org

Console:1, and Quality of Service:3. The Home Automation DCPs are most relevant to this project. The Sensor Management DCPs provide very limited semantic assets for specific sensors and actuators. The EnergyManagement and LowPower DCPs only concern management of low-power states of devices.

Description of the semantic coverage

The UPnP Device Architecture describes the protocols for discovery, description, control, eventing and presentation. Two general types of devices are defined, namely controlled devices, or simply devices, and control points. A controlled device has the role of a server, responding to requests from control points. The pre-requisite for a device or control point is the *IP addressing* using the DHCP protocol to obtain an IP address. Once the IP address is given, the 1st step is the *discovery*, in which a device advertises its services to the control points in the network and, vice versa, a control point searches for devices of interest. The discovery message contains data about the device type, device ID and a pointer to more detailed info. The 2nd step is the *description* that allows the control points in the network to learn about a device and its capabilities, and how to interact with this device. The description is specified in XML and contains vendor specific manufacturer info (model name and number, serial number, manufacturer name, website, etc.), and a list of any embedded devices or services. For each service, the description contains commands (or actions) to which the service responds, parameters (or arguments) and variables that model the device state (data type, range, events). The 3rd step is the *control* used by a control point to send actions to a device's services. To do this, a control point sends a suitable control message in XML to the service URL provided in the device description. The device returns action-specific values that may enforce changes in the variables that describe the run-time state of the service. The 4th step is the *eventing* that allows to subscribe and/or listen to changes in the state of variables for a specific service of a device. There is an option that allows subscription to events and a multicasting option. Event messages (also in XML) contain the names of one or more state variables and the current value of those variables. The 5th step is the presentation that is used by a control point to retrieve the URL from which it is possible to control the device and/or view the device status.

The semantic coverage of UPnP is shown in Figure 19. UPnP covers three type of devices (solar protection blind, HVAC system and HVAC_Zone Thermostat) and several services (Two Way Motion Motor service, HVAC_User Operating Mode service, HVAC_Setpoint Schedule service, HVAC_Fan Operating Mode service, Fan Speed service, Temperature Sensor Service, Control Valve service, House Status service).

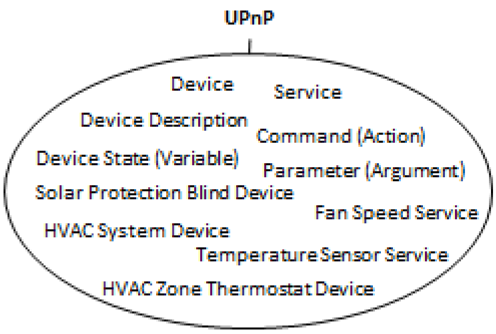


Figure 19: Visual representation of the semantic coverage of UPnP

Overall description of the consensus driven process leading to the model

The UPnP Forum, founded in 1999, consists of many hundreds of members, most of them Implementer Members and Basic Members. The Steering Committee members are the most influential and currently consist of CableLabs, Cisco Systems Inc., Intel, LG Electronics, PV, Samsung, TPVision, and ZTE. The Home Automation DCPs were drafted by Somfy, Siemens, and Honeywell.

3.3.21 W3C SSN

Model Acronym and Full Name

W3C (World Wide Web Consortium) SSN (Semantic Sensor Network) Ontology

Most relevant URLs, and other precise references

Semantic Sensor Network Ontology, http://www.w3.org/2005/Incubator/ssn/ssnx/ssn (2011).

Overall description

The World Wide Web Consortium (W3C)[37] has initiated the Semantic Sensor Networks (SSN) Incubator Group (which later became Community Group) to develop the Semantic Sensor Network ontology which can model sensor devices, systems, processes, and observations. The SSN ontology enables expressive representation of sensors, sensor observations, and knowledge of the environment. It is encoded in OWL and has begun to achieve broad adoption and application within the sensors community. It is currently being used by various organizations, from academia, government, and industry, for improved management of sensor data on the Web, involving annotation, integration, publishing, and search. The latest version was published in 2011.

Description of the semantic coverage

The SSN ontology is an OWL ontology that provides a framework to describe sensors, observations and related concepts. It does not describe domain concepts, such as time and locations, since these concepts are intended to be included from other ontologies via OWL imports. A sensor is a specific device whose purpose is to report measurements and observation real world phenomena. A sensor is different in nature from other types of devices such as actuators, because of its event based behaviour and the temporal relationships that need to be considered. The SSN ontology is a basis for reasoning about the measurements that can ease the development of advanced applications. For instance, when reasoning about sensors, constraints such as power restriction, limited memory, variable data quality need to be taken into account. It is possible to reason either about individual sensors as well as about the connection of a number of sensors. The semantic coverage of the SSN ontology is shown in Figure 20.

[37] www.w3.org

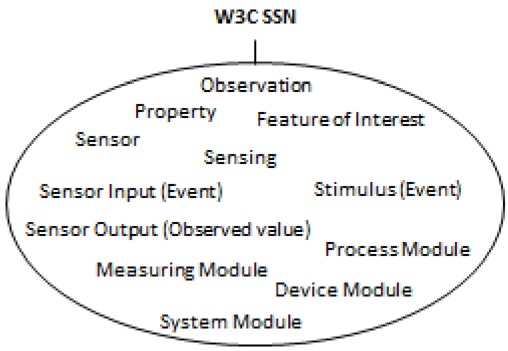


Figure 20: Visual representation of the semantic coverage of W3C SSN

The SSN ontology is composed of several modules, including a module imported from the DUL ontology to define some foundational concepts. These modules are the following:

- DUL module, which represents Designed Artifact, Event, Information Object, Method, Object, Physical Object, Process, Quality, Region, Situation;
- Skeleton module, which represents
    - Feature Of Interest, i.e., an abstraction of real world phenomena, such as thing, person, event;
    - Observation, i.e., a situation in which a sensing method has been used to estimate or calculate a value of a Property of a Feature Of Interest;
    - Property, i.e., an aspect of an entity that is intrinsic to and cannot exist without the entity and is observable by a sensor;
    - Sensing, i.e., a process that results in the estimation, or calculation, of the value of a phenomenon);
    - Sensor, i.e., any entity that can follow a sensing method and thus observe some Property of a Feature Of Interest. Sensors may be physical devices, computational methods, a laboratory setup with a person following a method, or any other thing that can follow a Sensing Method to observe a Property;
    - Sensor Input, i.e., an Event in the real world that triggers the sensor;
    - Sensor Output, i.e., a sensor outputs a piece of information (an observed value), the value itself being represented by an Observation Value),
    - Stimulus (an Event in the real world that 'triggers' the sensor. The properties associated to the stimulus may be different to eventual observed property. It is the event, not the object that triggers the sensor)
- System module, which represents systems;
- Process module, which represents Input, Output and Process;
- Measuring module, which represents Sensing Device, Sensor Data Sheet;
- Measuring Capability module, which represents Accuracy, Detection Limit, Drift, Frequency, Latency, Measurement Capability, Measurement Property, Measurement Range, Precision, Resolution, Response Time, Selectivity, Sensitivity;
- Deployment module, which represents Deployment, Deployment Related Process;
- Platform Site module, which represents Platform;
- Operating Restriction module, which represents Maintenance Schedule, Operating Property, Operating Range, Survival Property, Survival Range, System Lifetime;
- Data module, which represents Observation Value;

- Time module, which represents end Time and start Time;
- Constraint Block module, which represents conditions
- Device module, which represents devices;
- Energy Restriction module, which represents Battery Lifetime, Operating Power Range.

Overall description of the consensus driven process leading to the model

The editors of the SSN ontology are affiliated to CSIRO, Wright State University, University of Surrey, Universidad Politécnica de Madrid, Monterey Bay Aquarium Research Institute, Fraunhofer Gesellschaft, Pennsylvania State University, The Open University, University of Southampton, Open Geospatial Consortium, DERI at the National University of Ireland, Ericsson, Boeing, Fundacion CTIC, and others.

3.3.22 ZigbeeHA

Model Acronym and Full Name

Zigbee Home Automation Public Application Profile, Revision 29, Version 1.2

Most relevant URLs, and other precise references

The standard can be downloaded from http://zigbee.org/zigbee-for-developers/applicationstandards/zigbeehomeautomation/ for free after registration.

Overall description

ZigBee Home Automation is a global standard helping to create smarter homes that enhance the comfort, convenience, security and energy management for the consumer. It defines device descriptions and standard practices for applications commonly found in a residential or light commercial environment. Installation scenarios range from a single room to an entire home. The key applications included are lighting, HVAC, window shades, security, door locks, electricity measurement and smart appliances. Zigbee HA is an application profile on top of the Zigbee PRO networking technology.

Description of the semantic coverage

The standard specifies various devices in terms of their features and functions, and of the clusters they are expected to support. A cluster is a container for one or more attributes and/or messages in a command structure. An attribute is a data entity which represents a physical quantity or state. This data is communicated to other devices using commands. The standard then specifies many different devices including On/Off Switch, Level Control Switch, Remote Control, Mains Power Outlet, Door Lock, Simple Sensor, Consumption Awareness Device, Home Gateway, Smart plug, White Goods, Meter Interface, On/Off Light, Dimmable Light, Color Dimmable Light, Light Sensor, Occupancy Sensor, Shade, Window Covering Device, Heating/Cooling Unit, Thermostat, Temperature Sensor, Pump, Pressure Sensor, Flow Sensor, and various Intruder Alarm Systems.

Overall description of the consensus driven process leading to the model

Zigbee HA is a product of the Zigbee Alliance, which was founded in 2002 to promote and extend the IEEE 802.15.4 communication standard in order to make Zigbee products truly interoperable. Promoter members are Comcast, Freestyle, Itron, Kroger, Landis+Gyr, Legrand, NXP, Philips, Schneider Electric, Silicon Labs, and Texas Instruments. The alliance as a further ~150 contributing members and a few hundred adopter members. In the Zigbee HA standard, the ZigBee Alliance specifically and uniquely thanks Energy@home (see section 3.4.11) for its contribution to these technical specifications through technical documents, organization of test events, and active participation of its members.

The ZigBee Alliance plans to put all forms of its low-power wireless technology under one standard, ZigBee 3.0, with the aim to make it easier to connect many wireless devices in homes. Besides Zigbee HA these low-power wireless standards include Zigbee Light Link, Zigbee Building Automation, Zigbee), Zigbee Health Care, ZigBee Retail Services, and Zigbee Telecommunication Services.

3.3.23 Z-Wave

Model Acronym and Full Name
Z-Wave Application Layer

Most relevant URLs, and other precise references
Z-Wave Technical Basics Chapter 4 "Application Layer", 1 June 2011,
http://www.domotiga.nl/attachments/download/1075/Z-Wave%20Technical%20Basics-small.pdf.

Overall description
The Z-Wave protocol is a wireless RF-based communications technology designed specifically for control, monitoring and status reading applications in residential and light commercial environments. The protocol is specified by Z-wave Alliance[38] and the specifications are not publicly available. Various papers and text books describe in the technology in some detail though. It is a low- powered RF communications technology that supports full mesh networks without the need for a coordinator node. It operates in the sub-1GHz band, is designed specifically for control and status apps, and supports data rates of up to 100kbps. The application layer specification defines what and why two Z-Wave nodes communicate with each other, and contains the relevant semantics.

Description of the semantic coverage
Each device in a home or office can either control other devices or being controlled by other devices. Controlling devices are called Controllers, reporting devices are called Sensors, and controlled devices are called Actuators. It is also possible to combine a logical sensor controller or actor function within one physical device. Actors switch either digital (on / off for an electrical switch) or analogue signals (0 %. 100 % for a dimmer or venetian blind control). Sensors deliver either a digital signal (door, glass breaking, motion detector, window button on the wall) or an analogue signal (temperature, humidity, power). Z-Wave devices on the market can be categorized into one of the following function groups:

- Electrical switches are designed either as plug in modules for wall outlets or as replacement for traditional wall switches (digital actors). It's also possible to have these actors already built into certain electrical appliances such as electrical stoves or heaters.
- Electrical dimmers, either as plug in modules for wall outlets or as replacement for traditional wall switches (analogue actors)
- Motor control, usually to open or close a door, a window, a window sun blind or a venetian blind (analogue or digital actors)
- Electrical Display or other kind of signal emission such as siren, Led panel, etc (digital actors)

[38] www.z-wavealliance.org

- Sensors of different kind to measure parameters like temperature, humidity, gas concentration (e.g. carbon dioxide or carbon monoxide) (analogue or digital sensors)
- Thermostat controls: either as a one knob control or using a temperature display (analogue sensors)
- Thermostats controls such as TRVs (Thermostat Radiator Valves) or floor heating controls (analogue or digital actors)
- Remote Controls either as universal remote control with IR support or as dedicated Z-Wave Remote Control with special keys for network functions, group and/or scene control
- USB sticks and IP gateways to allow PC software to access Z-Wave networks. Using IP communication these interfaces also allow remote access over the internet

All communication within the Z-Wave network is organized in Command Classes, which are a group or commands and responses related to a certain function of a device. The Basic command class is the smallest common denominator of all Z-Wave devices. Every Z-Wave device must support the Basic command class. Device classes are organized as a hierarchy with three layers: 1) Every device must belong to a Basic device class; 2) Devices can be further specified by assigning them to a Generic device class; 3) Further functionality can be defined as assigning the device to a Specific device class. In case the Z-Wave device is assigned to a specific device class, it is required to support a set of command classes as functions of this specific device class. These required command classes are called Mandatory command classes and they are individual of certain generic and specific device classes. Besides the mandatory device classes, Z-Wave devices can support further Optional command classes. They may be very useful but the standard does not enforce the implementation of these classes. With Z-Wave it is not only possible to operate individual actions with appliances such as lights, heating and window blinds, but also create Scenes like “Leave for Work”, and select what you want to happen in your home, when you leave for the day. Also it is possible to create Events, for example, when a motion detector is tripped, a light can come on for 5 minutes. There is a Timer setting to set the lights or the thermostat to go on or off at a certain time. The semantic coverage of Z-wave is schematically depicted in Figure 21.

Overall description of the consensus driven process leading to the model

The Z-Wave Alliance was founded in 2005. It is a consortium of over 250 independent manufacturers as of 2013, who have agreed to build wireless home control products based on the Z-Wave standard. Principal members include ADT, GE/Jasco, Evolve, Ingersoll-Rand, Linear, FAKRO and Sigma Designs. Z-Wave was developed by a Danish startup called Zen-Sys that was acquired by Sigma Designs in 2008.

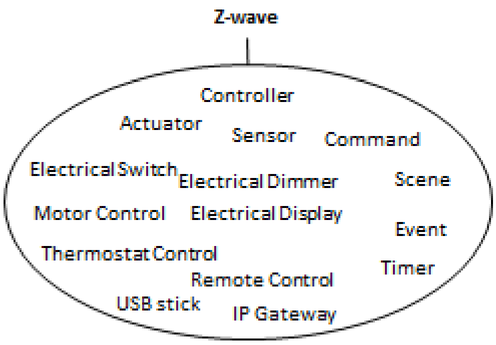


Figure 21: Visual representation of the semantic coverage of Z-wave

3.4 Other relevant bodies and projects

3.4.1 Agora

Most relevant URLs, and other precise references

http://www.reseau-domiciliaire.fr/home

Overall description

Agora was born when several French companies joined forces to design and distribute components, products and terminals that would communicate with services to provide better “smart home” living. The idea was to jointly review all ways to enable domestic technologies to communicate, interact and cooperate. The partners’ shared goal was to provide residents of “smart homes” with more fluid, more economical, more efficient services by building a bridge (which the consortium calls an “Agora bus”) linking everything together. So far, the consortium has been able to build prototypes of the Agora bus based on existing technologies, including many discussed in this document, such as UPnP, HGI, and Broadband Forum. No additional semantic assets were defined.

3.4.2 AIM*

Most relevant URLs, and other precise references

http://www.ict-aim.eu/

Overall description

AIM[39] was a European 7th Framework R&D project running from 2008-2010. AIM's main objective was to foster a harmonised technology for profiling and managing the energy consumption of appliances at home. The goal was to introduce energy monitoring and management mechanisms in the home network and provide a proper service creation environment to serve virtualisation of energy consumption, with the final aim of offering users a number of standalone and operator services. The main idea was to forge a generalised method for managing the power consumption of devices that are either powered on or in stand-by state. The AIM technology was aimed at white goods (refrigerators, kitchens, washing machines, driers), communication devices (cordless phones and wireless communication devices for domestic use) and audiovisual equipment (TV Sets and Set-top-boxes). The project did not produce semantic assets up and above the ones defined elsewhere in this document. The project partners in AIM were EURESCOM, France Telecom, KELETRON, CEFRIEL, Politecnico di Milano, INDESIT, Döbelt Datenkommunikation, Lantiq, Power Plus Communications, Philips Electronics Nederland, and BlueChip Technologies.

[39] www.ict-aim.eu

3.4.3 BACnet

Most relevant URLs, and other precise references

BACnet - A Data Communication Protocol for Building Automation and Control Networks - Overall description, ANSI/ASHRAE Standard 135-2012, Chapter 12 "Modeling Control Devices as a Collection of Objects", http://www.techstreet.com/ashrae/products/1852610.

Overall description

BACnet is a standard data communication protocol that enables interoperability between different building systems and devices in building automation and control applications. It was designed to allow communication of building automation and control systems for applications such as heating, ventilating, and air-conditioning control, lighting control, access control, and fire detection systems and their associated equipment. Its development started in 1987 and it was first standardized by the ASHRAE BACnet Committee (SSPC 135)[40] in 1995, and later became part of the ISO-EN-16484 suite [22]. It is promoted by the BACnet International[41] organization. Key promoter companies include Siemens, Honeywell, Delta Controls, Reliable Controls, Johnson Controls, Trane, Automated Logic, Lutron, and others. Chapter 12 of the BACnet standard defines 54 "objects", which are basic devices or device components. Every object has a required set of properties.

Description of the semantic coverage

The BACnet Standard addresses Fire, Security, Lighting, HVAC, Vertical Transport (elevators) products, among others. A BACnet device is often comprised of a microprocessor-based controller and software combination that is designed to understand and use the BACnet protocol. A BACnet device is typically a controller, gateway, or user interface. Every BACnet device contains a device object that defines certain device information, including the device object identifier or instance number. All information within an interoperable BACnet device is modeled in terms of one or more information objects. Each object represents some important component of the device, or some collection of information that may be of interest to other BACnet devices. A BACnet property conveys information about a BACnet object. Objects have a collection of properties, based on the function and purpose of the object. BACnet services are formal requests that one BACnet device sends to another BACnet device to ask it to do something. Services are grouped into five categories of functionality, namely object access (read, write, create, delete); device management (discover, time synchronization, initialize, backup and restore database); alarm and event (alarms and changes of state); file transfer (trend data, program transfer); and virtual terminal (human machine interface via prompts and menus). The semantic coverage is schematically depicted in Figure 22.

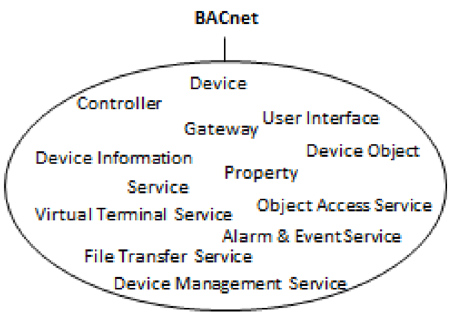


Figure 22: Visual representation of the semantic coverage of BACnet

---

[40] www.bacnet.org

[41] www.bacnetinternational.org

3.4.4 Broadband Forum

Most relevant URLs, and other precise references

Broadband Forum SD-282 "Control Signaling Device Abstraction Layer", http://www.broadband-forum.org/technical/technicalwip.php.

Overall description

Broadband Forum[42] develops multi-service broadband packet networking specifications addressing interoperability, architecture and management. Its work is directed at enabling home, business and converged broadband services, encompassing customer, access and backbone networks. The Broadband Forum issues Technical Reports (TR), which are prepared internally in Study Document (SD) before becoming Working Texts and finally TRs. One of the SDs the Broadband Forum currently is working on is SD-282 Control Signaling Device Abstraction layer. The document is still under development and its contents has not yet been published.

Description of the semantic coverage

SD-282 defines a Control Signaling and Device Abstraction layer that provides applications access to any M2M network of devices without the burden of understanding the communication technology of each device. Application of the Control Signaling and Device Abstraction layer is possible in several domains with different types of devices (camera device, sensor device, light device, monitor device, medication device) and associated protocols, and the general concepts of devices, objects, services, discovery, methods, parameters, state. This semantic coverage is schematically depicted in Figure 23.

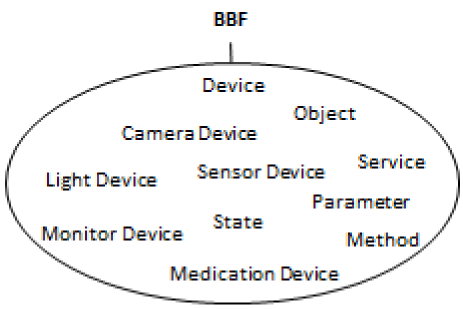


Figure 23: Visual representation of the semantic coverage of BBF

3.4.5 CECED

Most relevant URLs, and other precise references

- PI Standard 15.0, 21 March 2014, http://www.picertified.com/download/xml_download/Lang_PI15_0_EN_all.xml
- EDI-WHITE Final Messages, http://www.ceced.eudata.be/ICECED/easnet.dll/ExecReq/Search?eas:parent_id=201013 , 5 April 2006.

Overall description

CECED[43] is the European Committee of Domestic Equipment Manufactures, or the trade organization of white good manufacturers. They have executed a number of project to develop a protocol to

[42] www.broadband-forum.org

[43] www.ceced.org

make intelligent energy management a reality. The most relevant for this project are the projects EDI-WHITE and PI. PI is about standardizing a template for product information for cataloguing purposes. This template is continuously updated. The EDI-WHITE project finished in 2006 and standardized messages for electronic data interchange with suppliers, forwarders, banks and customers. EDI-WHITE is basically a subset of the UN/EDIFACT standard on United Nations/Electronic Data Interchange For Administration, Commerce and Transport[44]. Other projects that CECED is doing in the field op protocol standardization are EDI-Service and IRIS. EDI-Service is an electronic commerce standard for the after-sales service market, again based on UN/EDIFACT. IRIS is a standardized common language for exchanging repairing information among countries. CECED members are Arcelik, Ariston Thermo Group, BSH Bosch und Siemens Hausgeraete GmbH, Candy Hoover Group, DAIKIN Europe NV, de Longhi SpA, AB Electrolux, FAGOR Group, GORENJE d.d., INDESIT Company SpA, LG Electronics, LIEBHERR Hausgeraete, Miele & Cie. KG, Philips Consumer Lifestyle NV, Samsung Electronics, Groupe SEB, and Vorwerk Elektrowerke GmbH & Co. KG.

Description of the semantic coverage

A visual representation of the semantic coverage of CECED is provided in Figure 24. The CECED's EDI-WHITE project standardizes messages such as order, order response, order change, invoice, dispatch advice, price catalogue, remittance advice, receiving advice, transport instruction, sales and stock report, partner identification, delivery forecast and schedule and just in time delivery.

The CECED's PI Standard for Product Information aims at standardizing product information for cataloguing purposes. The white goods product groups in the PI catalogue include: Accessories, Built-in coffee machines, Cookers and Double Cookers , Dishwashers, Freezers, Fridge/Freezers combinations, Hobs, Hoods, Microwave Ovens, Modules (fryers, grills, hobs scales and sinks), Outdoor Grills, Ovens and Double Ovens , Plate warmers, Refrigerators, Side-by-Side, Steam Ovens, Steamers, Tumble Dryers, Vacuum Cleaners, Washer-Dryers, Washing Machines, Water Machines, Wine-storage.

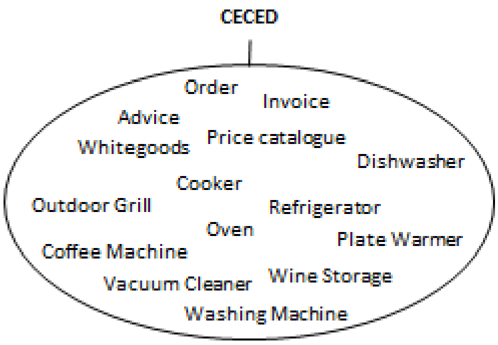


Figure 24: Visual representation of the semantic coverage of CECED

44 www.unece.org/trade/untdid/welcome.html

3.4.6 CEN/CLC/ETSI Smart Grid CG M490

Most relevant URLs, and other precise references

- CEN-CENELEC-ETSI Smart Grid Coordination Group “First Set of Standards”, November 2012, ftp://ftp.cen.eu/EN/EuropeanStandardization/HotTopics/SmartGrids/First%20Set%20of%20Standards.pdf
- CEN-CENELEC-ETSI Smart Grid Coordination Group “Smart Grid Reference Architecture”, November 2012, ftp://ftp.cencenelec.eu/EN/EuropeanStandardization/HotTopics/SmartGrids/Reference_Architecture_final.pdf

Overall description

In March 2011, the European Commission issued the Smart Grid Mandate M/490 [23] which was accepted by the three European Standards Organizations (ESOs), CEN, CENELEC and ETSI in June 2011. M/490 requests the ESOs to develop a framework that enables the ESOs to perform continuous standard enhancement and development in the smart grid field. In order to perform the requested work, the ESOs combined their strategic approach and established the CEN-CENELEC-ETSI Smart Grid Coordination Group (SG-CG) in July 2011. SG-CG is responsible for coordinating the ESOs reply to M/490. In 2012, the SG-CG produced a number of reports, which obtained approval from the ESOs. Among them are the Smart Grid Reference Architecture, and a First Set of Standards. The latter shows a first list of standards, enabling or supporting the deployment of Smart Grid systems in Europe. The list includes standards such as EN 13757 [18] (see section 3.3.14), EN 50090 [16] (see section 3.3.11), EN 14908 [24] (see section 3.4.17), IEC 61968 [19] (see section 3.3.18), and IEC 62056-53 [25] (see section 3.4.8). The document does not produce new semantic assets up and above the standards listed. The Smart Grid Reference Architecture document describes a technical reference architecture for European smart grids. In section 8.3.3 it discusses data models for the related information architecture. It concludes that besides the in our section 3.3.18 already referenced [19] also IEC 61850 [26] is relevant. However, IEC 61850 applies to electrical substations and is therefore out of the scope of this project. We therefore conclude that we have covered all the relevant semantics from the smart grid field in the various sections throughout this chapter.

3.4.7 CoAP

Most relevant URLs, and other precise references

Constrained Application Protocol (CoAP), draft-ietf-core-coap-18, 28 June 2013, https://datatracker.ietf.org/doc/draft-ietf-core-coap/.

Overall description

CoAP is a specialized web transfer protocol for use with constrained nodes and constrained (e.g., low-power, lossy) networks. The nodes often have 8-bit microcontrollers with small amounts of ROM and RAM, while constrained networks such as 6LoWPAN [27], often have high packet error rates and a typical throughput of 10s of kbit/s. The protocol is designed for M2M applications such as smart energy and building automation. It is defined as a subset of REST (REpresentational State Transfer) common with HTTP. It does not specify or standardize the contents of the messages. An example of a CoAP message exchange is given: The client sends a Confirmable GET request for the resource coap://server/temperature to the server. A response is returned in the Acknowledgement message that acknowledges the Confirmable request, including a Payload of "22.3 C". The URI scheme semantics still need to be standardized, but can be easily constructed from existing XML data models as given for other M2M networks. This draft standard is final and is in the process of becoming an IETF Proposed Standard RFC. The editors are from Sensinode and University of Bremen.

3.4.8 DLMS/COSEM

Most relevant URLs, and other precise references

Companion Specification for Energy Metering: COSEM interface classes and OBIS identification system, DLMS User Association, 27 August 2013, http://dlms.com/PASSWORD/Books/Blue_Book_11th_edition.pdf , also called "Blue Book". Available for members only. A free excerpt is available at http://dlms.com/documents/Excerpt_BB11.pdf , which is what we used for our analysis. Part of the COSEM standard is an Object Identification System (OBIS). Its latest version (2.3, October 2005, with a corrigendum published in April 2006) can be found here: http://dlms.com/documents/members/OBIS_list_v2.3_GK051026.zip

Overall description

DLMS/COSEM (Device Language Message Specification / COmpanion Specification for Energy Metering) is a world-wide standard that specifies smart meter functionality. It is developed and maintained by the DLMS User Association[45]. DLMS is a generalized concept for abstract modelling of communication entities. It is a middleware protocol that can be applied on various physical layer technologies, such as Zigbee, M-bus, but also Internet. It is designed to support messaging to and from (energy) distribution devices in a computer-integrated environment. It is an international standard published as IEC 61334-4-41 [28]. Applications like remote meter reading, remote control and value added services for metering any kind of energy, like electricity, water, gas or heat are supported.

COSEM (Companion Specification for Energy Metering) sets the rules, based on existing standards, for data exchange with energy meters. It is designed for use with DLMS but can also be applied to other protocols. COSEM achieves this by using object modelling techniques to model all functions of the meter, without making any assumptions about which functions need to be supported, how those functions are implemented and how the data are transported. The formal specification of COSEM interface classes forms a major part of COSEM. To process and manage the information it is necessary to uniquely identify all data items in a manufacturer-independent way. Therefore, the definition of OBIS (Object Identification System) is an essential part of COSEM. OBIS is standardized as IEC 62056-61 [29]. DLMS/COSEM is standardized as IEC 62056-53 [25] and IEC 62056-62 [30], of which the latter matches the Blue Book specifications.

The DLMS User Association has 281 members (29 April 2014), which are mostly Full Members having one vote. In October 2011, the Management Committee of DLMS User Association consisted of representatives of Électricité de France R&D, ERDF, ITRON ITALIA SpA, GNARUS ENGINEERING Services Ltd., IBERDROLA, Elster GmbH, Görlitz AG, Landys+Gyr (Europa) AG, SAGEM Communication, and ISKRAEMECO d.d..

[45] www.dlms.org

Description of the semantic coverage
COSEM is an extensive and complex specification whose semantics cannot be straightforward captured. Chapter 1 gives an introduction on the DLMS/COSEM system, Chapter 4 (especially section 4.2) presents the COSEM Interface Object Classes and specifies the logical names of the objects. The semantic coverage is shown in Figure 25.

A COSEM Physical device contains COSEM Logical Device(s) and must contain a Management Logical Device. A Logical device must have a Logical Device Name (LDN), and contains some objects (Associations objects and Application objects), but also holding parameters and measurement values. The naming system is based on OBIS, the Object Identification System: each logical name is an OBIS code. There are OBIS values groups (A,B,C,D,E,F) and for example, the group A has values:

- 0: Abstract Object
- 1: Electrical Related Object
- 4: Heat Cost Allocator Related Object
- 5: Cooling Related Object
- 6: Heat Related Object
- 7: Gas Related Object
- 8: Cold Water Related Object
- 9: Hot Water Related Object

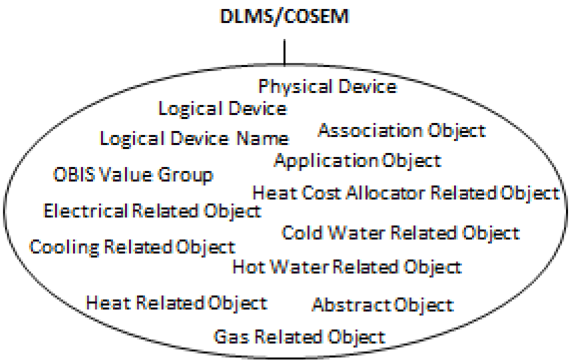


Figure 25: Visual representation of the semantic coverage of DLMS/COSEM

3.4.9 DEHEMS

Most relevant URLs, and other precise references
Nazaraf Shah, Kuo-Ming Chao, Tomasz Zlamaniec, Adriana Matei, "Ontology for Home Energy Management Domain", Digital Information and Communication Technology and Its Applications, Communications in Computer and Information Science, Volume 167, 2011, pp 337-347. http://dx.doi.org/10.1007/978-3-642-22027-2_28.

Overall description
The Digital Environment Home Energy Management System (DEHEMS) [46]project is a European 7th Framework R&D project running from 2008-2011, looking at how technology can improve domestic energy efficiency. The intention was to develop and test a home energy management system for the home market using Living Labs http://www.dehems.eu/living-labs in 5 cities across Europe.

[46] www.dehems.eu

Description of the semantic coverage
In Reference [31] the authors of *Ontology for Home Energy Management Domain* describe a system for intelligent energy management for home appliances that uses house hold profiles and energy consumption profiles of electrical appliances to provide households with effective advice on their energy consumption thereby enabling them to take focused and effective actions towards efficient energy use. Energy profiles are used to detect and diagnose abnormalities in energy consumption and recommend remedial actions to household in order to remove or minimize the effect of abnormalities. The encoding of the knowledge is distributed among rules (using Jess as rule base system) and a domain ontology. The domain concepts are used as runtime facts of the rule base system on which rules operate. The pieces of advice like what action to perform are encoded in an ontology. All pieces of advice within ontology are linked to hierarchy of energy consumption activities. The *Ontology for Home Energy Management Domain* paper subsequently describes the DEHEMS ontology as an extension of the SUMO ontology and provides some excerpts of the ontology. From this we can extrapolate the semantic coverage shown in Figure 26, which includes Device, Electrical appliances, Household appliances, Cleaning appliances, Laundry appliances, Washing machine, Energy saving tips, Energy star, Energy Consumed, Spin performance, Spin speed, Wash performance, Brand name, Standby wattage, Wattage, Number of washing programs, EU Energy label class, and Energy Star rating.

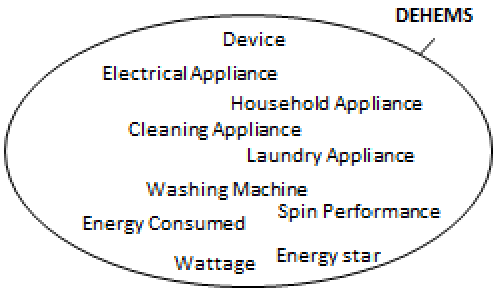


Figure 26: Visual representation of the semantic coverage of DEHEMS

3.4.10 ebbits

Most relevant URLs, and other precise references
Ebbits ontology, D3.2 “Vertical and horizontal business vocabularies”, http://www.ebbits-project.eu/downloads.php?cat_id=1&download_id=27 , D4.3 “Coverage and scope of a semantic knowledge model”, http://www.ebbits-project.eu/downloads.php?cat_id=1&download_id=28 , D4.5 “Analysis and design of semantic interoperability mechanisms”, http://www.ebbits-project.eu/downloads.php?cat_id=1&download_id=47, and D7.2 “Event and data structures, taxonomies and ontologies”, http://www.ebbits-project.eu/downloads.php?cat_id=1&download_id=31.

Overall description
The *ebbits* project[47] is a 7th Framework European R&D project, running from 2010-2014, which does research in architecture, technologies and processes, which allow businesses to semantically integrate the Internet of Things into mainstream enterprise systems and support interoperable end-to-end business applications. It will provide semantic resolution to the Internet of Things and hence present a new bridge between backend enterprise applications, people, services and the physical

[47] www.ebbits-project.eu

world. The *ebbits* platform features a Service oriented Architecture (SoA) based on open protocols and middleware, effectively transforming every subsystem or device into a web service with semantic resolution. nhe *ebbits* platform thus is expected to enable the convergence of the Internet of People (IoP), the Internet of Things (IoT) and the Internet of Services (IoS) into the "Internet of People, Things and Services (IoPTS)" for business purposes. Another 7th Framework European R&D project taking a similar approach is BEMO-COFRA (Brazil-Europe - Monitoring and Control Frameworks)[48]

Description of the semantic coverage

The ebbits deliverables D3.2, D4.3, D4.5, D7.2 discuss vocabularies, semantic models, ontologies, etc., but are not beneficial to define a precise semantic coverage. The ebbit ontology is based on the HYDRA ontology, which is extensively described in the ebbits deliverables. Some deliverables (D4.5: pages 21-22, and D7.2) contains some excerpts which cover the concepts of <u>Service</u>, <u>Device</u>, <u>Sensor</u> (<u>Thermometer</u>, <u>RFID tag</u>), <u>Event</u>, <u>State</u>, <u>Sensing</u>, <u>Alert</u>, and <u>Measurement</u>, which we have used to define the semantic coverage depicted in Figure 27.

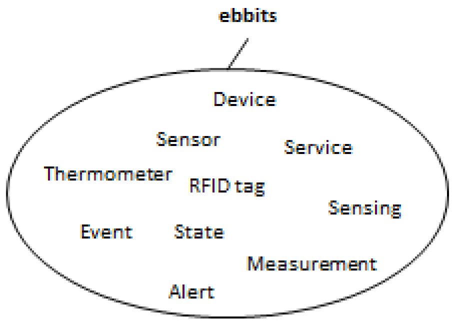


Figure 27: Visual representation of the semantic coverage of ebbits

3.4.11 Energy@Home

Most relevant URLs, and other precise references

Energy@Home Data Model, version 0.9, revision 0.5, 3 February 2014, http://www.energy-home.it/Documents/2014-02-dm/E@h_data_model_v0.9.pdf.

Overall description

Energy@home is a collaborative project between Electrolux, Enel, Indesit Company and Telecom Italia. The aim of the project is to develop a communication infrastructure that enables provision of Value Added Services based upon information exchange related to energy usage, energy consumption and energy tariffs in the Home Area Network (HAN). The communication infrastructure enables cooperation between the main devices involved in residential energy management, namely electronic meter, smart appliances, smart plugs, home residential gateways and customer interfaces.

Description of the semantic coverage

The definition of the Energy@Home data model is based on the ZigBee SEP2 specification. Energy@Home identified missed functionalities and attributes in the ZigBee SEP2 specification and submitted them to the ZigBee Alliance. The parts added to the SEP2 specification by Energy@Home consist of an Appliance Identification Package, an Appliance Events and Alerts Package, an Appliance

[48] www.bemo-cofra.eu

Statistics Package, an Appliance Control Package, an On/Off Package, and a Power Profile Package. Also describes the Metering, Pricing and Time function sets were expanded compared to the original ZigBee SEP2 specification.

The semantic coverage of Energy@Home is shown in Figure 28. The Home Area Network (HAN) is used for communication between devices within the home such as sensors, smart plugs, smart thermostats and household appliances. A Smart appliance is an appliance connected in the HAN with some intelligence to cooperate with the other home actors in order to provide new services to the consumer. Smart plug is a device that typically has a power meter to calculate the power/energy consumption of the connected load and can be used to remotely power on/off the load. Customer Interfaces are physical devices (logical components) that can be visualized by a PDA, a pc or a Smart Phone. The Load profile is the variation in the electrical load versus time, and is specialized by the Power profile concept, which represents the variation of power consumption of an electrical load versus time. The Appliance Power Profile specializes the power profile with information about the energy consumption of an appliance (load profile related to its cycles) and some information for load shedding or load shifting its usage. Load Shedding is as method of reducing demand on the energy generation system by temporarily rationing distribution of energy to different geographical areas. Load Shifting is an electric load management technique to shift the pattern of energy use of a device (load profile), moving demand from the peak hours to off-peak hours of the day. Peak demand or peak load describes a period in which electrical power is expected to be provided for a sustained period at a significantly higher than average supply level. Peak demand fluctuations may occur on daily, monthly, seasonal and yearly cycles. An Energy Cost Algorithm is used to obtain the price of energy at a given time (e.g. € per kWh from 08:00 to 19:00) replicating the conditions applied by the Energy Retailer. An Energy Regulation Algorithm defines the strategy for coordinating Smart Appliances behavior, in order to reach energy consumption or cost optimization and to guarantee the overall performance of the system, using as inputs the global energy consumption, its cost, Appliances Power Profile and their status. Main control techniques involved in the Energy Regulation algorithm are load shifting and shedding.

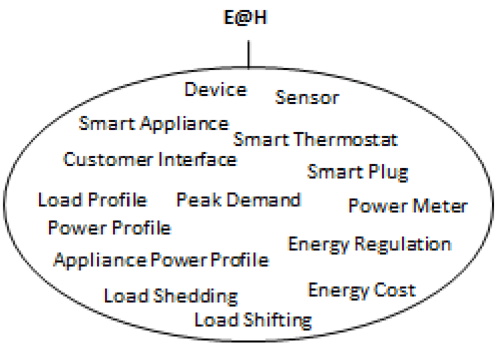


Figure 28: Visual representation of the semantic coverage of Energy@Home

3.4.12 ENERsip

Most relevant URLs, and other precise references

ENERsip deliverables for WP3 "M2M for ADR Infrastructure", https://sites.google.com/a/enersip-project.eu/enersip-project/results/deliverables/wp3.

Overall description

ENERsip (ENERgy Saving Information Platform for generation and consumption networks)[49] was a European 7th Framework R&D project running from 2010-2012. The main objective of the ENERsip project was to create an adaptive, intelligent and open service-oriented platform that allows end users to optimise, in near real-time, and to save energy by remotely monitoring, controlling and coordinating power generation and consumption in neighbourhoods with residential and commercial buildings. The objective is both short-term, with respect to development and testing of prototypes, and also long-term, with respect to adoption of service-oriented compliant and energy efficiency solutions. Methods for measuring effectiveness and quality of proposed solutions were to be selected and/or developed as part of the pilot and validation phase. D3.2 of the project defined M2M communication middleware interfaces to integrate with M2M Concentrators, to monitor third party Sensors and actuators, different elements of the Power distribution infrastructure and applications. The middleware platform is said to provide open interfaces, based on the SOA paradigm, to allow integration between the different components of the ENERsip architecture to enable data exchange and monitoring of the different elements of the Power distribution infrastructure and applications. D3.2 has not been made publicly available.

3.4.13 eu.bac

Overall description

Eu.bac[50] is the European Building Automation Controls Association. It is an industry association which mission includes:

- Influence the development and effective implementation of EU directives and regulation, in order to achieve the optimal balance of controls and automation systems & services in new and existing homes and buildings
- Advocate the benefits of balancing controls and automation systems & services; validate outstanding technology; and establish best practices
- Drive European and worldwide industrial standards for the benefit of the industry
- Initiate quality standards through audits, certificates or labels to endorse quality and energy efficiency products and systems & services

The eu.bac Homes sector group represents European manufacturers of control and balancing equipment for residential properties. The long term aim of the group is to ensure that all homes in Europe benefit from appropriately controlled and balanced heating, cooling, ventilation and hot-water systems. An important part of their work is certification. Home Controls and Building automation Controllers with the eu.bac Certification Mark and eu.bac Energy Efficiency Label demonstrate proven quality and energy efficiency according to European standards and directives. The certification tests against the standards EN 15500 [32], EN 15232 [33], EN 16484-2/3 [22] (see section 3.4.3), EN 50090 [16] (see section 3.3.11) and EN 14908 [24] (see section 3.4.17). No

[49] https://sites.google.com/a/enersip-project.eu/enersip-project/
[50] www.eubac.org

additional semantic assets are defined by eu.bac. eu.bac has 26 member organizations, of which Schneider Electric, Honeywell, Belimo, Siemens, Danfoss, and Sauter are in the board.

3.4.14 HGI

Overall description

The HGI[51] publishes requirements for digital home building blocks. Those building blocks are the hardware and software in the digital home that connect consumers and services. They include home gateways, home networks, and home network devices. Currently the HGI is constructing a *Smart Home Appliance (Device) Model Template,* to be published as GD-042[52]. It is supposed to accompany RD-036, which will describe HGI's Smart Home Architecture and System Requirements.

Description of the semantic coverage

GD-042 outlines a list of the representations in XSD/XML or OWL/RDF currently provided by other parties, rather than providing an HGI specification. A common Device Model Template is proposed to specify device capabilities. Part of this template are Services (interfaces), Actions (operations), Device classes (types), Device instances with some attributes (i.e., name, manufacturer, model name, model number, universal product code) , States (state variables), and Events (asynchronous info). The semantic coverage of HGI is shown in Figure 29.

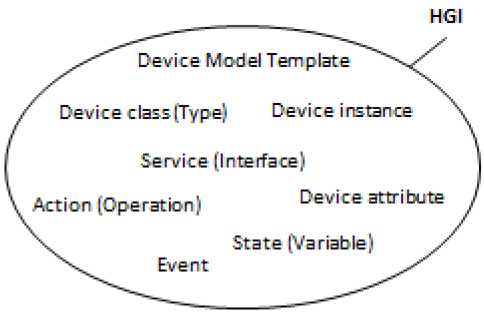


Figure 29: Visual representation of the semantic coverage of HGI

3.4.15 IFC

Most relevant URLs, and other precise references

IFC4 (Industry Foundation Classes 4) Specifications (March 2013): http://www.buildingsmart-tech.org/ifc/IFC4/final/html/index.htm . Also available for download (after free registration) at http://www.buildingsmart-tech.org/downloads/ifc/ifc4/ifc4-html-documentation-68mb.

Overall description

Industry Foundation Classes are standards for the use of object technology in construction and facilities management. They are produced by buildingSMART[53], an international organisation which aims to improve the exchange of information between software applications used in the construction industry. In 1995, the founding members were Autodesk, AT&T, ARCHIBUS, Carrier Corporation , Hellmuth, Obata & Kassabaum (HOK), Honeywell, Jaros Baum & Bolles, Lawrence Berkeley

---

[51] www.homegatewayinitiative.org

[52] http://www.telecomitalia.com/content/dam/telecomitalia/it/archivio/documenti/Innovazione/HotTopic/Casa%20connessa/Overall%20slide%20pack%20BBWF%202013_final.pdf

[53] www.buildingsmart.org

Laboratory, Primavera Systems, Softdesk, Timberline Software Corp, and Tishman Research Corp. Today the organization consists of 15 regional chapters with their own membership, and an International Council made up of 31 members from the regional chapters.

IFC represent an open specification for Building Information Modeling (BIM) data that is exchanged and shared among the various participants in a building construction or facility management project, between applications developed by different software vendors without the software having to support numerous native formats. The latest version of IFC, IFC4, incorporates several extensions of IFC in building, building service and structural areas; enhancements of geometry and other resource components; various quality improvements; fully integrated simple ifcXML specification; and a new documentation format. It is standardized as ISO 16739 [34].

Description of the semantic coverage
The semantic coverage is shown in Figure 30. The IFC specification is structured in Data item names for types, entities, rules and functions, Attribute names within an entity, Property set definitions, and Quantity set definitions. The IFC specification consists of the following four conceptual layers:

- Resource layer, which is the lowest layer and includes all individual schemas containing resource definitions, those definitions do not include an globally unique identifier and shall not be used independently of a definition declared at a higher layer. Examples of resources are DateTime, Material, Actor, Profile, Geometry, Measure, Property, Quantity, Topology, Utility, Cost, Presentation, Constraint, Approval, Structural Load.
- Core layer, which includes the kernel schema and the core extension schemas, containing the most general entity definitions, all entities defined at the core layer, or above carry a globally unique id and optionally owner and history information. Core layers are Control Extension, Product Extension and Process Extension.
- Interoperability layer, which includes schemas containing entity definitions that are specific to a general product, process or resource specialization used across several disciplines, those definitions are typically utilized for inter-domain exchange and sharing of construction information. The interoperability layer consists of Shared building services elements, Shared components elements, Shared building elements, Shared management elements, Shared facilities elements.
- Domain layer, which is the highest layer and includes schemas containing entity definitions that are specializations of products, processes or resources specific to a certain discipline, those definitions are typically utilized for intra-domain exchange and sharing of information. Domain layer contains Building Controls domain, Plumbing Fire Protection Domain, Structural Elements domain, Structural Analysis domain, HVAC domain, Electrical Domain, Architecture domain, Construction management domain.

The IFC is a an extensive and complex specification that covers many domains that are not of interest for our study, except for the HVAC domain, which is represented in the *IfcHvacDomain* schema and whose scope is defined as:

- The segments, fittings and connections that constitute duct and piping distribution systems typically used for building services, such as for air conditioning, ventilation and exhaust-air systems; chilled water, steam and heating hot water, potable water, waste, natural gas and LPG systems, etc.

- Equipment typically used in building services systems, such as boilers, chillers, fans, and pumps and the vibration isolation associated with these components.
- Terminal and flow control devices, such as air vents and grilles, variable air volume modulators, valves, and dampers.

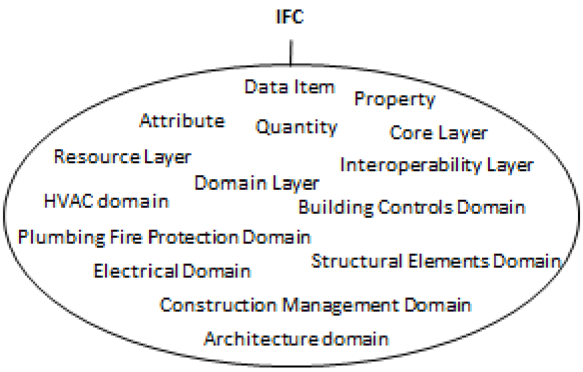


Figure 30: Visual representation of the semantic coverage of IFC

3.4.16 LightingEurope

Most relevant URLs, and other precise references

IES TM-23-11 "Lighting Control Protocols", Illuminating Engineering Society of North America (IES 2011), http://www.ies.org/PDF/Store/TM-23-11_FINAL.pdf.

Overall description

LightingEurope emphasizes the importance of control in the lighting domain, where control refers to the systems or commands that regulate the intensity of electric luminaires in response to some stimulus or action on the part of the building occupants. This stimulus can be direct, e.g., the moving of a switch from one position to the other that completes an electrical circuit and causes the luminaires to energize, or it can be less direct, e.g. in case of occupancy, time, motion, and the presence or absence of daylight. One of the documents LightingEurope has produced is IES TM-23-11 "Lighting Control Protocols". It identifies 17 different protocols being used for lighting control today, on 4 different physical layers. The protocols are widely varying where some have specified their semantics (e.g. Zigbee Light Link[54]) whereas others are still under development (e.g. DALI[55]). With IES TM-23-11, LightingEurope aims to "encourage greater coordination among disciplines and allow the continued integration of lighting control with other major building systems. Greater integration will ultimately lead to more efficient and healthier buildings enhancing the experience of the built environment for more people".

Description of the semantic coverage

Section 2 of IES TM-23-11 subsequently describes the core concepts and terms related to lighting in order to provide a common vocabulary. Section 3 presents a basic architecture that is common for controlling the light output of a luminaire or light source. This architecture consists of components (such as user initiated devices and power controllers), interfaces and signals. Examples of user initiated devices are Switch and Wallbox Dimmer. There are also initiating devices such as occupancy sensors. Examples of power controllers are Ballast, Driver and Transformer. There are different types

[54] http://zigbee.org/Standards/ZigBeeLightLink/Overview.aspx
[55] www.dali-ag.org

of light sources, such as incandescent lamp, fluorescent lamp, and LED lamp, which require different input forms from the user device and different type of power controllers. The semantic coverage of LightingEurope is shown in Figure 31.

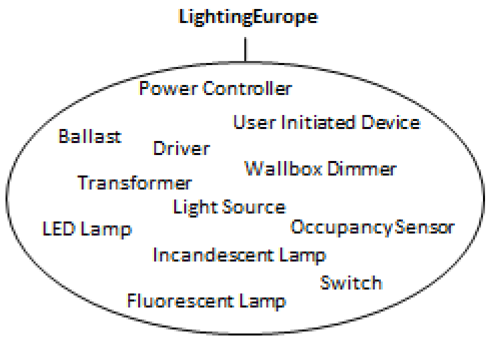


Figure 31: Visual representation of the semantic coverage of LightingEurope

3.4.17 LonWorks

Most relevant URLs, and other precise references

LonMark Device Classes and Functional Profiles,
http://www.lonmark.org/technical_resources/resource_files/spid_master_list#DeviceClasses.

Overall description

LonWorks is a networking platform created to address the needs of control applications used for the automation of various functions within buildings, such as lighting and HVAC. It is a peer to peer network in which all devices speak to each other. The platform is built on a protocol created by Echelon Corporation for networking devices over media such as twisted pair, powerlines, fiber optics, and RF. The communications protocol, twisted pair signaling technology, power line signaling technology, and Internet Protocol (IP) compatibility standard were standardized in 2008 as ISO/IEC 14908-1, -2, -3, and -4 [24].

LonMark International[56] is a global membership organization created to promote and advance the business of efficient and effective integration of open, multi-vendor control systems utilizing ISO/IEC 14908-1 and related standards. LonMark establishes interoperability guidelines by profiling the interfaces to a device's functions at the exchange level, and certifies manufacturers' products accordingly.

Description of the semantic coverage

Each device or node contains a microprocessor to communicate the protocol to each other device. On the LonMark website we could find a list of device classes for defining the sematic coverage of LonWorks, which is shown in Figure 32.

Access/Intrusion/Monitoring devices, Automated Food Service devices, Energy Management devices, Fire & Smoke Devices, Gateways devices (Telephone Gateway, Internet Gateway, etc.), Generic Actuators, Generic Controllers, Generic Human-Machine Interfaces (Remote Control, Panel Interface), HVAC devices (VAV Controller, Fan Coil Unit Controller, Roof Top Unit Controller, Chiller,

[56] www.lonmark.org

Thermostat, Pump Controller, Unit Heater, etc.), I/O devices, Industrial devices (e.g., Filtration Systems and Power Supply), Lighting (e.g., Dimmer, Lamp actuator, Occupancy controller, Switch, Lighting controller), Motor Controls (e.g., Variable-Speed Motor Drive, Sunblind Actuator, Sunblind Controller), Network Infrastructure devices, Programmables devices, Refrigeration devices (Defrost Controller, Evaporator Controller, Refrigeration Thermostat Controller, Railheat Controller), Semiconductor Fabrication, Sensors (e.g., Light Sensor, Global Solar Radiation Sensor, Time Sensor, Pressure Sensor, Temperature Sensor, Occupancy Sensor, etc.), Transportation (e.g., door controls and electric vehicles), Vertical/Conveyer Transportation (Elevator), Whitegoods (Clothes-e.g., Clothes Dryer, Cooking-e.g., oven, Storage-e.g., refrigerator, Miscellaneous-e.g., dishwasher), Wiring Devices (e.g., Hardwired Gas Detection Shutdown and Hardwired Fire Alarm Shutdown).

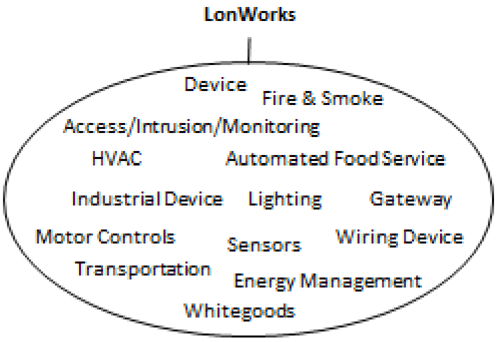


Figure 32: Visual representation of the semantic coverage of LonWorks

3.4.18 oBIX

Most relevant URLs, and other precise references

OASIS, obix-v1.1-csprd02, Committee Specification Draft 02 /Public Review Draft 02, 19 December 2013, http://docs.oasis-open.org/obix/obix/v1.1/csprd02/obix-v1.1-csprd02.pdf.

Overall description

oBIX (OASIS Open Building Information eXchange Technical Committee)[57] is an industry-wide initiative to define XML- and web-services-based mechanisms for building control systems. The scope of the OASIS (Organization for the Advancement of Structured Information Standards) Open Building Information Exchange (oBIX) TC is to develop a publicly available web services interface specification that can be used to obtain data in a simple and secure manner from HVAC, access control, utilities, and other building automation systems, and to provide data exchange between facility systems and enterprise applications. In addition, the TC will develop implementation guidelines, as needed, to facilitate the development of products that use the web service interface. oBIX defines a common information model to represent diverse M2M systems and an interaction model for their communications. The current version is 1.1. The current oBIX mailing list consists of representatives of Cisco Systems, Continental Automated Buildings Association (CABA), IBM, William Cox, Institute of Computer Aided Automation, NEXTDC Ltd., Schneider Electric Industries SAS, TIBCO Software Inc., Trane, Tridium, University of North Carolina at Chapel Hill, and US Department of Defense (DoD).

[57] www.obix.org , https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=obix

Description of the semantic coverage

The common information model represents Objects that can be extended through the so-called Contracts. Contracts are standard OBIX objects used as a template or patterns. The oBIX standard addresses building control systems, such as Heating and Cooling (HVAC), Lighting, Security, Energy Management, and Life/Safety Alarms. Different type of data, from simple to complex are considered, for example, the room temperature of the lobby (simple), the list of people currently in East Wing with time of entry (lengthy), the current state of all systems across an entire university campus (complex), and the variation of internal humidity of sports hall over the last 6 months (reports). Simple data follows the International Unit of Measurement System, i.e., Mass in kilograms, Length in meters, Time in second, etc., and more complex data structures are built from these. Data is accessed at a Datapoint and exposed by a Point Service, Datapoints are revealed by a Discovery Service, Data trends are reported by a History Service, Critical events are signaled by the Alarm Service. The semantic coverage is shown in Figure 33.

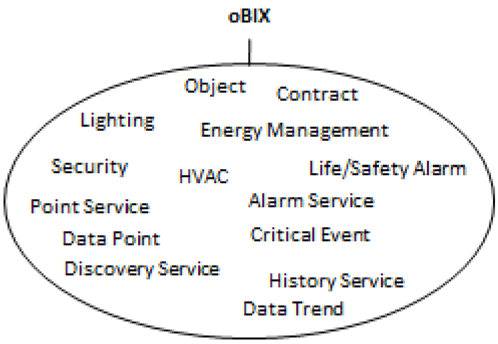


Figure 33: Visual representation of the semantic coverage of oBIX

Figure 34. Visual representation of the semantic coverage of oBIX

3.4.19 SensorML

Most relevant URLs, and other precise references

OGC SensorML: Model and XML Encoding Standard, v2.0.0, 4 February 2014, https://portal.opengeospatial.org/files/?artifact_id=55939. Here we analysed an older version: OpenGIS Sensor Model Language (SensorML) Implementation Specification, v1.0.0, 17 July 2007.

Overall description

Sensor Model Language (SensorML)[58] is an initiative part of the Sensor Web Enablement (SWE) [59]activity of OGC (Open Geospatial Consortium)[60] for establishing a “sensor web” through which applications and services should be able to access sensors of all types over the Web. The aim of SensorML is to define processes and processing components associated with the measurement and post-measurement transformation of observations. SensorML provides standard models and an XML encoding for describing any process, including the process of measurement by sensors and instructions for deriving higher-level information from observations. Processes described in SensorML are discoverable and executable. All processes define their inputs, outputs, parameters, and method, as well as provide relevant metadata. SensorML models detectors and sensors as processes that convert real phenomena to data.

58 www.ogcnetwork.net/SensorML
59 www.ogcnetwork.net/SWE
60 www.opengeospatial.org

The document we analyzed specifies the models and XML encoding for the core SensorML, as well as the definition of several SWE Common data components of the SWE framework (the specification of SWE Common data components is out of our scope). The document also gives an informal description of the SensorML components and the common data components of SWE in natural language. It also provides UML diagrams that can be used as basis to build ontologies and should be used as such, as recommended in the specification. Also XML schemas and example instances are provided.

Description of the semantic coverage

The essential elements of SensorML strongly overlap with the SSN ontology and are schematically depicted in Figure 34. These elements are:

- Phenomenon - A physical property that can be observed and measured, such as temperature, gravity, chemical concentration, orientation, number-of-individuals;
- Observable property - A parameter or a characteristic of a phenomenon subject to observation;
- Observation - An act of observing a property or phenomenon, with the goal of producing an estimate of the value of the property;
- Measurement - An observation whose result is a measure;
- Component - Physical atomic process that transforms information from one form to another. For example, a Detector typically transforms a physical observable property or phenomenon to a digital number. Example Components include detectors, actuators, and physical filters;
- System- Composite physically-based model of a group or array of components, which can include detectors, actuators, or sub-systems. A System relates a process to the real world and therefore provides additional definitions regarding relative positions of its components and communication interfaces;
- Process Model- Atomic non-physical processing block usually used within a more complex Process Chain. It is associated to a Process Method which defines the process interface as well as how to execute the model. It also precisely defines its own inputs, outputs and parameters;
- Process Chain- Composite non-physical processing block consisting of interconnected sub-processes, which can in turn be Process Models or Process Chains. A process chain also includes possible data sources as well as connections that explicitly link input and output signals of sub-processes together. It also precisely defines its own inputs, outputs and parameters;
- Process Method- Definition of the behavior and interface of a Process Model. It can be stored in a library so that it can be reused by different Process Model instances (by using 'xlink' mechanism). It essentially describes the process interface and algorithm, and can point the user to existing implementations;
- Detector- Process Model profile that represents an atomic component of a Measurement System defining sampling and response characteristic of a simple detection device. A detector has only one input and one output, both being scalar quantities. More complex Sensors such as a frame camera which are composed of multiple detectors can be described as a detector group or array using a System or Sensor. In SensorML a detector is a particular type of Process Model;

Sensor- Specific type of System representing a complete Sensor. This could be for example a complete airborne scanner which includes several Detectors (one for each band).

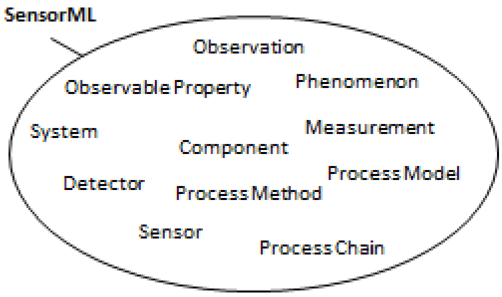


Figure 34: Visual representation of the semantic coverage of SensorML

3.4.20 SESAME

Most relevant URLs, and other precise references

Slobodanka Tomic, Anna Fensel, Tassilo Pellegrini, "SESAME Demonstrator: Ontologies, Services and Policies for Energy Efficiency", in Proceedings of the 6th International Conference on Semantic Systems I-SEMANTICS 2010, 1-3 September 1-3, Graz, Austria, http://dx.doi.org/10.1145/1839707.1839738.

Overall description

SESAME Demonstrator: Ontologies, Services and Policies for Energy Efficiency is a publication in the context of the SESAME[61] project, which uses semantic modelling and reasoning to support home owners and building managers in saving energy and in optimizing their energy costs, while maintaining their preferred quality of living. The SESAME project (running from September 2009 to November 2010) was an international research collaborative project of Forschungszentrum Telekommunikation Wien GmbH, E-Smart Systems d.o.o., eSYS Informationssysteme GmbH (Austria), EZAN – Experimental Factory of Scientific Engineering, Upper Austria University of Applied Sciences, and Semantic Web Company GmbH. It resulted in a technical solution that actively assists end-consumers to make well-informed decisions and control regarding their energy consumption. The SESAME solution is a full-fledged prototype covering a sensor and smart metering solution that can be installed in the house, equipped with the semantic software and user interfaces performing reasoning and control of the house on the basis on defined policies, sensor inputs and interactions. The cited paper gives an overview of the system, which encodes domain knowledge in an RDF/OWL ontology (not publicly available) that is then used to create SWRL (Semantic Web Rule Language) rules for more advanced reasoning.

Description of the semantic coverage

The semantic coverage is shown in Figure 35. The SESAME ontology includes a number of general concepts such as resident and location, and concepts specific to the automation and the energy domains, such as Device, Tariff, Energy Usage Profile, Account. The Device class is further specialized to model an Appliance, Sensor, or UI device. Properties in the device model are consumption per hour, peek power, the switch on/off status but also the required state "to be switched on/off". The property "canBeStarted" models the state of the devices whose activation can be scheduled, e.g., a filled washing machine. The central function-level concept in the SESAME ontology it the Configuration class, which has two the subclasses Activity (automation activity) and EnergyPolicy. A

[61] http://sesame.ftw.at

Configuration connects Appliance, Sensor and UI Device into a joint task. The Configuration can provide regulation of different types, e.g. regulation on time, occupancy of location, threshold value.

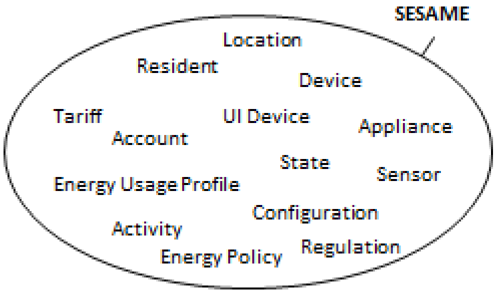


Figure 35: Visual representation of the semantic coverage of SESAME

3.4.21 TIBUCON

Most relevant URLs, and other precise references

TIBUCON D2.3 High Level Data Models and Message Structures, http://www.tibucon.eu/docs/D2%203%20High%20Level%20Data%20Models%20and%20Message%20Structures%20V1.0.pdf.

Overall description

TIBUCON[62] (self-powered wireless sensor network for HVAC system energy improvement – Towards Integral BUilding CONnectivity) is a European 7th Framework R&D project which proposes a solution for an easy to deploy and easy to maintain building environment monitoring. It is based on a Wireless Sensor Network (WSN) that consists of spatially distributed autonomous sensors with the objective of monitoring physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, and cooperatively send their data through the network to a main location. D2.3 describes the standards used at the upper levels of the WSN protocol stack. Only the application layer is in the scope of our analysis. TIBUCON is a collaboration of Mostostal Warszawa S.A., Tekniker-IK4, Giroa, University of Southampton, Katholieke Hogeschool Kempen, and E&L Architects. D2.3 was edited by Tekniker-IK4.

Description of the semantic coverage

The Tibucon approach uses DPWS (Devices Profiles for Web Services) and SensorML schemas. Section 3 should elaborate on the TIBUCON data models, but it only gives an overview of SensorML purposes and no data models are actually presented. Annex-A gives 3 examples of SensorML XML schemas adapted for TIBUCON in which some headers "have been omitted for simplicity". These schemas represent temperature sensor, humidity sensor and remaining battery monitoring. The rest of the semantic coverage is completely overlapping with SensorML. The semantic coverage is shown in Figure 36.

[62] www.tibucon.eu

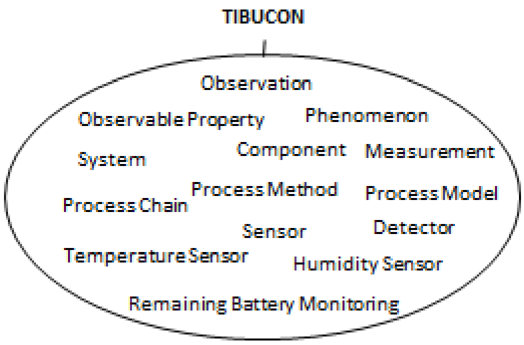


Figure 36: Visual representation of the semantic coverage of TIBUCON

3.4.22 VoCamp

Overall description

VoCamp (Vocabulary Camp)[63] is a series of informal events where people can spend some dedicated time creating lightweight vocabularies/ontologies for the Semantic Web/Web of Data. The emphasis of the events is not on creating the perfect ontology in a particular domain, but on creating vocabs that are good enough for people to start using for publishing data on the Web. The 5th VoCamp took place in Kaiserslautern (Germany) from 13-14 June 2013, where scientific foundations for standardization of M2M communication for energy management of Energy using and producing Products (EupP) in buildings and its environments were discussed. Major contributions were from research projects (FP7) SmartCoDe (TU Kaiserslautern), Adapt4EE, industry stakeholders, ETSI, and CENELEC as standardization bodies, and the European Commission. Presentations are mad available by the conference organizer, but have not yet been uploaded to the conference website, http://cps.cs.uni-kl.de/vocamp.

63 http://vocamp.org

4. Visual representation of key terms

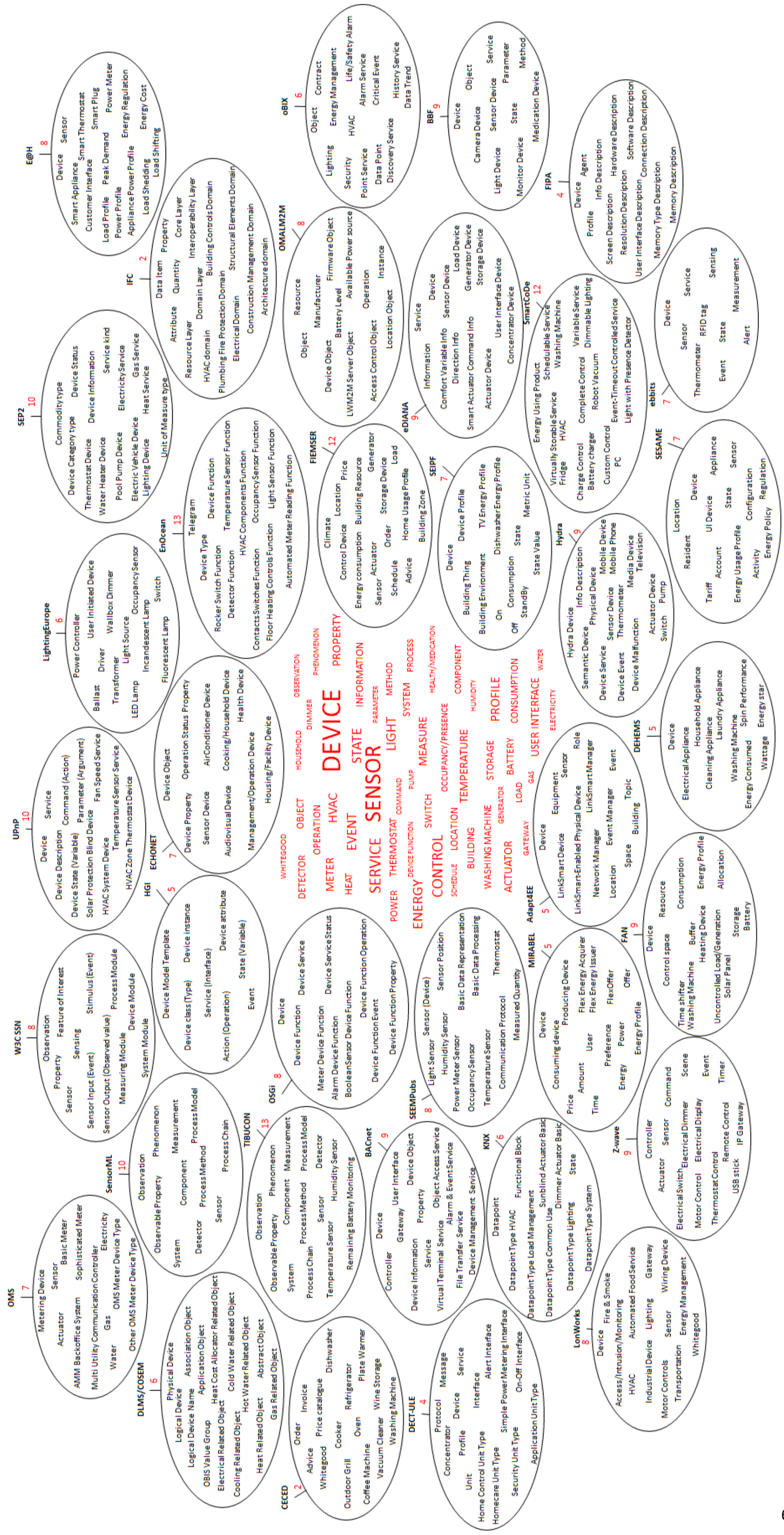
The term asset is used in this report in a broad sense, since it refers to a source that can present a project, a set of documents, an ISO standard, a working group, a committee, a paper, a homepage (of a wiki, or of any other website) that is somehow related to energy management and/or home appliances. Therefore, an asset may refer to one well-defined single ontology, but in most cases is a pointer to a set of multiple documents, several related standards and distinct articles on a web site or wiki, from which a single ontology should be derived.

In order to support the stocktaking task with an overall representation that could help the reader in visualizing the key terms used by different assets , we have created the visual representation in Figure 37, which schematically depicts between 10 and 15 key terms for each asset[64]. This visualization is intended to be an initial step towards the definition of a common semantics for the smart appliances domain. As such, it should be considered as a preliminary and partial result that will be taken as input in the following task of our study, namely creating ontologies based on an accurate semantic analysis. Figure 37 also shows some overlaps (in red and capital letters, in the middle of the figure), Figure 38 provides an enlargement of the overlap. We acknowledge that this visualization does not provide a precise semantic representation of the considered assets nor of their overlaps and differences. Ambiguities and inconsistencies are possible. For example, two assets may use the same term with a different meaning and consequently they will result in the overlaps of Figure 37, although in reality they do not share the same semantics. These ambiguities issues were addressed when creating accurate ontologies for the assets in the short list. Figure 37 only gives an indication of the linguistic overlap between key terms used by different assets, but with no guarantee that multiple uses of the term *Device*, for example, actually mean the same thing.

The overlaps in Figure 37 were identified by comparing the key terms used by different assets. The comparison was incremental, namely we started by comparing key terms of two assets and identifying the overlaps, then we compared the key terms of a third asset with the existing overlaps (i.e., the intersection of assets 1 and 2), but we also compared this third asset with asset 1 and asset 2 individually in order to find eventual new overlaps, and so forth for all the assets. If a term can be found in Figure 38, it means that at least two of the considered assets used that term. If a term is represented in an individual asset, but not in the middle of Figure 37, this shows a difference in the terminology used by other assets.

Every time an overlapping term was identified, the corresponding font in in the middle of Figure 38 was increased. Therefore, the bigger the terms, the more recurrent these terms are among the assets, for example, *Device*, *Sensor*, *Service* and *State*. Overlapping terms were added in the middle of Figure 37 if there was an almost exact match (at least of the stem). For example, the term *Device* matches exactly the term *Device* in several individual maps (exact match), but it is also used to match the term *Device object* in the ECHONET map and the term *Device Category type* in the SEP2 map (almost exact match). As another example of matching, the term *Measure* matches the stem of term *Measurement* of the SensorML map (stem match).

[64] If there are less than 10 terms, then the information about that asset was not sufficient to reach the lower bound of 10 terms.



[65]

Figure 37: Visual representation of key terms of the semantic assets considered in the stocktaking task

[65] This figure can be found at https://docs.google.com/file/d/0B2nnxMhTMGh4aEk3RFhXWHoyQkU

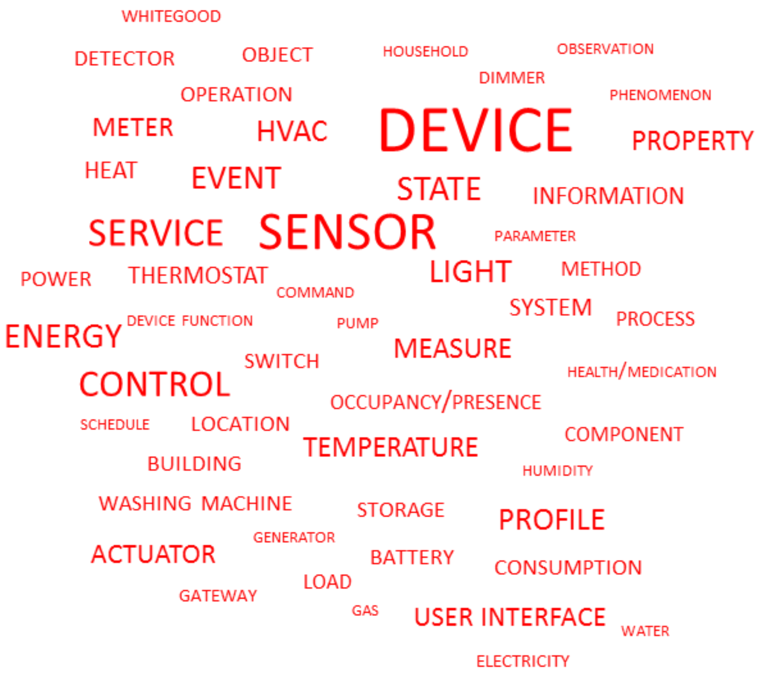


Figure 38: Visual representation of overlap (enlargement of middle part of figure 37)

Each of the assets representations in Figure 38 shows a number (in red) next to the name of the corresponding asset. For example, the UPnP map has a number *10* and the EnOcean map has a number *13*. This number indicates how many key terms used by that specific asset recur in the overlaps in the middle of Figure 38. In other words, the UPnP map has 10 terms overlapping with other maps, namely includes 10 of the most recurring terms among all assets.

5. Ontologies

5.1 Approach

The first step was to translate the 20 semantic assets in the first version of short list into ontologies in OWL in order to formally capture their semantics and be able to automatically reason about their content. Out of the 20 assets:

- 4 assets were already expressed in OWL, namely eDIANA, Hydra, PowerOnt (previously SEIFP), and W3C SSN. However, only W3C SSN provided an URL to the corresponding OWL file, while the other assets provided detailed documentation about their OWL ontologies, but not an URL. Therefore, we contacted the authors of these ontologies in order to acquire the original OWL file, which are essential for us to make sure that the reference ontology is based on the actual models that were defined in the projects or organizations, and not on our own interpretations of the available documentation.
- 16 assets needed to be translated into OWL from scratch.
    - Some of these assets provided UML-like data models from which we could (more or less) straightforward create a corresponding ontology. Some of these data models were built reflecting specific structures of the underlying implementation languages. We therefore needed to make an extra effort to abstract from these implementation details and capture the actual semantics.
    - Other assets consisted of technical specifications in terms of natural language descriptions, often supported by tables, from which we had to capture the semantics originally intended for these assets by their creators. While some of these specifications were clear and well-structured, supporting us to a great extent in our translation task, other specifications were ambiguous and inconsistent, requiring a major effort in the translation task.
    - Although having XML schema representations was one of the criteria we adopted to select the assets in the short list, we did not use these schemas, when available, to automatically generate OWL from XML. This was a specific design choice we made, given the resources that could be allocated for the translation task. In fact, it was more effective to create the OWL ontologies top-down, extracting the semantics from the natural language descriptions and tables provided by the assets, rather than bottom-up, by first automatically generating OWL from XML, and then having to edit the result in order to make a proper OWL ontology out of it. Notice that this choice still allows future extensions in terms of mappings from the OWL ontologies to XML for the interested stakeholders.

In this way, for some assets we have reused existing work created by other renowned bodies and organizations, rather than creating new ontologies from scratch, according to best practices in ontology engineering. When translating the other 16 assets into OWL, we took care of expressing the intended semantics correctly so that others, both machines and humans, could properly understand and reuse the ontologies being created. Particularly, we have created the ontologies in this document with the following best practices in mind:

1. include basic metadata that allow others to correctly understand and properly reuse the ontology being built, such as *creator*, *date of issue*, *title*, *description* and *source* of the ontology;
2. make the ontology self-descriptive by using labels, definitions and comments for each class or property;
3. provide proper documentation not only using label and comments, but creating a human-readable description that explains the main classes and properties;
4. make the ontology accessible for a long period by providing some guarantee of maintenance;
5. publish the ontology at a stable URL to guarantee persistent access, and facilitate reusability.

According to these principles, the second step of our approach was to create a human-readable description that reflects the content of the OWL files, in order to support the reader in understanding the main concepts and navigating the ontologies. These descriptions are presented in Chapter 4, together with important observations concerning the development of the ontologies and suggestions for future extensions.

The third step was to publish the ontologies at a stable URL[66] in order to guarantee persistent access and facilitate their (re)usability in the smart appliances community. The smart appliances website provides a page for each of the 20 assets in the short list, with the URL to download the corresponding ontology and a human-readable explanation to describe the main classes and properties. In order to guarantee transparency during the process and take into account the feedback of the stakeholders, especially the “owners” of the considered assets, each page includes a tab for posting comments (available when logged on to the website with a Google-account).

In parallel to the steps described above, the fourth step was to extend the long list of 43 assets with assets that resulted to be missing after the discussions that took place at the 1st stakeholders' workshop. Therefore, we contacted the authors/owners of these assets to obtain relevant material, and, if existing, to acquire the original OWL files.

5.2 Ontologies description

The following sections present the ontologies that we have created[67] providing their *title*, the *source* document used as a main reference to create the ontology, a *description* of the main classes and properties, and, eventually, *observations* necessary to better understand the choices underlying the ontology design and suggestions for its future extension. Notice that the project does not have the resources to elaborate every ontology in OWL in all possible detail. However, we think this is not necessary as, to achieve the final goal of the project, we only need to find the commonalities between the various ontologies. Moreover, having learned from our approach, every stakeholder can now do the work himself and improve/extend the ontology to his liking given the open character of our results. These ontologies have to be considered as an intermediate result that allows us to achieve the final goal of the project, namely provide a reference ontology for the smart appliances domain to create semantic interoperability among assets from different stakeholders, but they are not the ultimate result of this project themselves.

[66] https://sites.google.com/site/smartappliancesproject/ontologies

[67] These ontologies are expressed in OWL-DL and serialized in Turtle (therefore, their file extension is “.ttl”), which is a compact syntax alternative to RDF/XML. These ontologies can be opened with any ontology editor, such as TopBraid Composer, Protégé and NeOn.

These ontologies and their descriptions are also published online, where the "owners" of the corresponding assets can check them to validate whether the meaning they originally intended for their assets is actually reflected in our ontologies.

Table 2. Ontologies overview

Ontology title	Source Language	URL	Number of classes	Number of Object properties	Number of Datatype properties
Dectule	Natural Language + Tables + XML	https://sites.google.com/site/smartappliancesproject/ontologies/dect_ule-ontology	76	6	4
Echonet	Natural Language + Tables	https://sites.google.com/site/smartappliancesproject/ontologies/echonet-ontology	187	27	2
Ediana	OWL	https://sites.google.com/site/smartappliancesproject/ontologies/ediana-ontology	70	25	12
Enocean	Natural Language + Tables	https://sites.google.com/site/smartappliancesproject/ontologies/enocean-ontology	240	4	3
Fanfpai	Natural Language + Javadoc	https://sites.google.com/site/smartappliancesproject/ontologies/fan-ontology	31	16	17
Fiemser	Natural Language + UML	https://sites.google.com/site/smartappliancesproject/ontologies/fiemser-ontology	47	5	34
Fipa	Natural Language + Tables + UML	https://sites.google.com/site/smartappliancesproject/ontologies/fipa-ontology	14	14	18
Hydra device	OWL	https://sites.google.com/site/smartappliancesproject/ontologies/hydra-ontology	66	9	14
Knx	Natural Language + Tables	https://sites.google.com/site/smartappliancesproject/ontologies/knx-ontology	20	1	3
Mirabel	Natural Language + Tables + UML	https://sites.google.com/site/smartappliancesproject/ontologies/mirabel-ontology	24	16	0
Omalwm2m	Natural Language + Tables + XML	https://sites.google.com/site/smartappliancesproject/ontologies/oma-lightweight_m2m-ontology	30	4	7
Oms	Natural Language + Tables	https://sites.google.com/site/smartappliancesproject/ontologies/oms-ontology	67	0	1

Osgidal	Natural Language + Javadoc	https://sites.google.com/site/smartappliancesproject/ontologies/osgi_dal-ontology	17	9	19
Poweront	OWL	https://sites.google.com/site/smartappliancesproject/ontologies/dogpower-ontology	945	41	60
Seempubs	Natural Language + Tables	https://sites.google.com/site/smartappliancesproject/ontologies/seempubs-ontology	44	9	4
Sep2	Natural Language + Tables +UML + XSD	https://sites.google.com/site/smartappliancesproject/ontologies/sep2-ontology	39	5	12
Smartcode	Natural Language + Tables	https://sites.google.com/site/smartappliancesproject/ontologies/smartcode-ontology	32	3	4
Upnp	Natural Language + XSD + XML	https://sites.google.com/site/smartappliancesproject/ontologies/upnp-ontology	8	11	23
W3C SSN	OWL	https://sites.google.com/site/smartappliancesproject/ontologies/w3c_ssn-ontology	116	137	6
Z-Wave	Natural Language + Code lists	https://sites.google.com/site/smartappliancesproject/ontologies/z-wave-ontology	77	6	0

5.2.1 DECT ULE

Ontology title

Dectule: Digital Enhanced Cordless Telecommunications (DECT) Ultra-Low Energy (ULE)

Source

'HF-Protocol', 'HF-Service', 'HF-Interface', 'HF-Profile', version1.0, 23 January 2014, available at (for free after registration) http://www.ulealliance.org/registration.aspx?f=11.

Ontology description

The DECT ULE HF standard is based on a star network topology of network entities represented by the `HFNetworkEntity` class in the DECT ULE ontology. A `HFNetworkEntity` can be a `HFConcentrator`, which is the network's master device, or a `HFDevice`. There are up to thousands of devices supported by the concentrator and connected to it. The HF protocol supports several types of HF messages exchanged between network entities (i.e., commands, requests, responses), and each of these messages has a message type code. A `HFMessage` is structured in 3 fields (i.e., network, transport and application layers).

A `HFNetworkEntity` implements one or more services. A `HFService` is either fundamental for the correct operation of a HF network, or provides advanced network features that may be useful on certain applications. Services from the latter category are optional to implement, while fundamental services are mandatory. Mandatory services for the `HFConcentrator` are the `AttributeReporting`, `BindManagement`, `DeviceInformation` and `DeviceManagement` services. Mandatory services for a `HFDevice` are the `AttributeReporting`, `DeviceInformation` and `DeviceManagement` services.

An `HFUnit` is a conceptual entity inside a `HFDevice` that instantiates the functionality of a specific type. There are several unit types, namely `HomeControlUnitType` (Simple On-Off Switchable, Simple On-Off Switch, Simple Level Controllable, Simple Level Control, Simple Level Controllable Switchable, Simple Level Control switch, AC Outlet, AC Outlet with Simple Power Metering, Simple Light, Dimmable Light, Dimmer Switch, Simple Door Lock, Simple Door Bell, Simple Power Meter), `SecurityUnitType` (Simple Detector, Door Open Close Detector, Window Open Close Detector, Motion Detector, Smoke Detector, Gas Detector, Flood Detector, Glass Break Detector, Vibration Detector, Siren), `HomecareUnitType` (Simple Pendant), `ApplicationUnitType` (User Interface, Generic Application Logic), and `ProprietaryUnitType`. Each `HFUnit` has a unique identifier.

An `HFInterface` is a conceptual entity inside a `HFUnit` that defines a collection of commands and attributes, allowing for units to understand each another. Interfaces can be mandatory or optional to implement by a unit, and they have a role – client or server – associated with them. An `HFInterface` has attributes.

Observations

- The attributes of the `HFInterface` class are not defined in this version of the ontology because their level of granularity is too fine. However, this ontology can be extended to cover also the attributes by adding them under the `Attribute` class, according to the HF-Interface specification.

5.2.2 ECHONET

Ontology title

Echonet: Energy Conservation and HOmecare NETwork (ECHONET) for Device Objects

Source

ECHONET Specifications Appendix 'Detailed Requirements for ECHONET Device Objects' Release C, 31 May 2013, available at http://www.echonet.gr.jp/english/spec/pdf_spec_app_c_e/SpecAppendixC_e.pdf.

Description

The Echonet ontology represents Echonet device objects and their properties. A `Device` defines one or more `DeviceObject`. Device objects represent mechanical functions of a device and aim at facilitating controls and status verification through communications between devices. There are general properties applicable to any device object, such as `hasOperationStatus`. These general properties are defined as subproperties of the `hasDeviceObjectProperty` property.

There are 7 groups of device objects, namely `AirConditionerRelatedDevice`, `AudiovisualRelatedDevice`, `CookingHouseholdRelatedDevice`, `HealthRelatedDevice`, `HousingFacilitiesRelatedDevice`, `ManagementOperationRelatedDevice` and `SensorRelatedDevice`. Each group has a corresponding code (`hasGroupCode` property) and is characterized by the `hasOperationStatus` property, which indicates whether the function native to this group of objects is operating or not (ON/OFF). The CodeList class defines enumerations used to represent admitted values for some properties, for example, the OperationStatus class defines the instances On and Off as admitted values for the hasOperationStatus property.

Each of the 7 groups mentioned above is further refined in device object subclasses with a specific code (hasClassCode property). For example, the AirConditionerRelatedDevice group includes the AirCleaner, Dehumidifier, ElectricHeater, and HomeAirConditioner classes, among others. Specific properties that characterize a certain device class, but not any device object, are defined as subproperties of the hasClassSpecificProperty property. For example, the hasOperationModeSetting property characterizes the HomeAirConditioner class.

Observations

- The general properties applicable to any device object, such as hasOperationStatus are defined globally under the hasDeviceObjectProperty property. The amount of DeviceObject classes was such that it was not possible to restrict (the cardinality of) these properties for all classes. For the future it is advised to restrict these properties locally at the level of each DeviceObject subclass.
- The amount of hasClassSpecificProperty properties specific to device classes, but not applicable to any device object, was such that it was not possible to define them all. However, we have defined three properties (hasDetectionThresholdLevel, hasOpenCloseSetting, and hasOperationModeSetting) that should be used as example to further populate the hasClassSpecificProperty according to the ECHONET Device Objects specification.

5.2.3 eDIANA

Ontology title

Ediana: Embedded Systems for Energy Efficient Buildings (eDIANA) ontology for Device Awareness

Source

'D2.2-A Ontology for Device Awareness', 30 November 2009, available at http://s15723044.onlinehome-server.info/artemise/documents/D22A_Ontology_for_Device_Awareness_m10_IMSML.pdf.

Ontology description

The eDIANA ontology defines the universe of concepts and their relations in the domain of eDIANA Platform Architecture, related to device awareness. The ontology defines three main classes, namely the Information, Service and Device classes. The Information class contains the different categories of information that will be referenced by the elements defined in the Service and Device classes. It includes Direction_Information, Comfort_Variable_Information (such as Humidity_Information, Luminosity_Information, Noise_Information, and Temperature_Information), and Smart_Actuator_Command_Information (such as Change_Configuration_Command_Information, Delayed_Turn_Off_Command_Information, Delayed_Turn_On_Command_Information, Turn_Off_Command_Information, and Turn_On_Command_Information).

The Service class specifies the different interfaces at a very high level,. They are divided in External_Services and Internal_Services. The concrete definition of these interfaces is recommended as future work in the document used as source of the ontology.

The Device class contains different categories of devices that compose the eDIANA platform to enable device awareness services and plug-and-play services, by characterizing the devices, their properties and their interfaces. Devices include Concentrator, Actuator, Appliance (including Generator, Load, Storage), Sensor (including Video_Camera, Airflow_Sensor, Gas_Sensor, Humidity_Sensor, Light_Sensor, Power_Sensor, Sound_Sensor, Sun_Radiation_Sensor, Temperature_Sensor, Fire_Sensor, Movement_Sensor and Smoke_Sensor), and User Interface.

Observations

- We did not create the eDIANA ontology. We are reusing the OWL version that was provided to us by the authors of the 'D2.2-A Ontology for Device Awareness' document.

5.2.4 EnOcean

Ontology title

Enocean: EnOcean Alliance Equipment Profile (EEP)

Source

'EnOcean Equipment Profiles (EEP) ', Version 2.6, 17 December 2013, available at http://www.enocean-alliance.org/eep/.

Ontology description

The Enocean ontology specifies the user data embedded in the structure of a radio telegram as defined by the EnOcean Equipment Profile (EEP). Therefore, the ontology defines an EEP_profile class. Through the hasElement property, the EEP_profile class is characterized by 3 elements:

- the RORG class, which represents the ERP radio telegram type using a code, for example, the value F6 represents an RPS telegram type;
- the FUNC class, which represents the basic functionality of the data contained in a radio telegram, for example, TemperatureSensor, AutomatedMeterReading, Detector, and HVAC_component; and
- the TYPE class, which represents the specific characteristics of a device type, for example, a temperature sensor with range between -10°C and 30°C (TemperatureSensor_range-10Cto30C class).

The ontology defines 4 types of telegrams according to the EEP profile, namely RPS, 1BS, 4BS and VLD, which are represented by the corresponding classes TelegramRPS, Telegram1BS, Telegram4BS, and TelegramVLD, respectively. Each telegram has a RORG (hasRORG property), and can have several device functions (hasFUNC property) and types (hasTYPE property).
Each RORG class, FUNC class and TYPE class has a code (hasRorgCode property, hasFuncCode property and hasTypeCode property, respectively). These codes are used to assemble the 3 field code that characterizes a specific telegram. For example, the code A5_02_04 characterizes a 4BS telegram (hasRorgCode with value A5) with a temperature sensor function (hasFuncCode with

value `02`) and a temperature sensor type with range between -10°C and 30°C (`hasTypeCode` property with value `04`).

Observations

- The TYPES are defined completely for the `TelegramRPS` and `Telegram1BS` classes. For the `Telegram4BS` class the TYPES are defined until and including the `A5_10` subclass. For the `TelegramVLD` class the TYPES are not defined at all. For completeness, it is advised to add the remaining TYPES in the future.
- The EEP document defines enumerations that are used to further characterize the specific TYPE of telegrams. These enumerations are too many and too detailed to be included in the current version of the ontology. However, the ontology could be extended in the future to cover also this aspect of the EnOcean Equipment Profile.
- The source used to create the ontology is a secured pdf from which the information could not be automatically copied. As a consequence, comments that could better explain the telegrams are missing in the ontology.

5.2.5 FAN

Ontology title

Fanfpai: Flexible Power Alliance Network (FAN) Flexible Power Application Infrastructure (FPAI) ontology

Source

'Interface description: Interface report', Version 1.0 (final), 7th January, 2014, available at (for free after registration) http://www.flexiblepower.org/downloads/.

Ontology description

The Fanfpai ontology describes the resources (appliances) used in the Flexible Power Application Infrastructure (FPAI) . These resources are defined in the Resource Abstraction Interface (`RAI` class), which is used to express the energetic flexibility that appliances can offer and how this flexibility should be exploited. The RAI is an interface layer between:

- the Resource Abstraction Layer (`RAL` class) that monitors and controls the appliances and knows how much flexibility they can offer. The RAL consists of two main components: the resource manager (`ResourceManager` class) and the resource driver (not considered in this ontology);
- the energy apps (`EnergyApp` class) that are typically provided by a third party and exploit the flexibility that appliances have to offer. An energy app is only interested in exploiting energetic flexibility and not in the details of a specific appliance, such as a washing machine, for instance.

A `Resource` represents an appliance within a household or a building that can provide flexibility with regard to consumption, storage and production of energy. There are several type of resources defined in FPAI:

- `TimeShifter` resources, which are a category of appliances that produce or consume energy according to a predetermined energy profile and whose flexibility comes from their ability to shift the start time of this profile. Typical examples of time shifting appliances are `WashingMachine`, `DishWasher`, `AutomaticVacuumCleaner`;
- `Buffer` resources, which are a category of appliances that can provide electrical flexibility. With a buffer appliance one can choose to consume/produce more energy now (within certain

operational constraints) so that it consumes/produces less energy later, or the other way around. Most buffers are thermal. Examples of such appliances are `Refrigerator`, `Freezer`, `HeatingSystem`;

- `EnergyStorage`, which is category of appliances similar to buffers, but with the main difference that with buffers the electrical energy only flows in one direction: it is either consumption or generation, while with a storage appliance the electrical energy flow is bidirectional. The storage category includes self-discharging batteries (e.g. Li-In/NiMH batteries), chemical storage batteries with conversion loss (e.g. flow batteries), and mobile storage in electrical vehicles;
- `UncontrolledLoadOrGeneration`, which is a category reserved for appliances that cannot be controlled and, as a consequence, cannot offer flexibility. It is however important to know how these appliances behave energetically to make informed decisions about the usage of flexibility in the other three categories. A `SolarPanel` is an example of an uncontrolled generation appliance, which generates energy that cannot be controlled since it depends on external natural conditions (i.e. the weather), whereas `Lighting` represents an uncontrolled load.

Important concepts in the ontology are `ControlSpace`, which is used to describe the energetic flexibility of a particular resource/appliance, and `Allocation`, which indicates how this flexibility should be used. The `ResourceManager` constructs and communicates a `ControlSpace`. In response to a `ControlSpace` communication, a `ResourceManager` can receive an `Allocation`, which contains a precise `EnergyProfile` that the `ResourceManager` should try to follow as closely as possible. There are several control spaces that correspond to the different categories of resources, namely `TimeShifterControlSpace`, `BufferControlSpace`, `EnergyStorageControlSpace`, and `UncontrolledControlSpace`. Each control space is characterized by specific properties. The ontology also defines the `Energy`, `Power`, `Duration` classes and their corresponding units of measure.

Observations
None.

5.2.6 FIEMSER

Ontology title

Fiemser: Friendly Intelligent Energy Management Systems in Residential Buildings (FIEMSER) ontology

Source

'D5 FIEMSER Data Model', February 2011, available at http://www.fiemser.eu/wp-content/uploads/2011/12/D5_FIEMSER-data-model_m9_CSTmb_REVIEW.pdf.

Ontology description

The Fiemser ontology describes the main classes of the Energy-focused BIM model and WSN-related data that are part of the FIEMSER data model. The ontology describes the building space organization in terms of the `Building`, `BuildingPartition`, `BuildingSpace` and `BuildingZone` classes. A building partition defines a part of a building managed by either a dweller (e.g., a flat) or a facility manager (e.g., a common building area). A building space defines the physical spaces of the building. A building zone defines a functional area in the building that will be controlled as a unique zone. A building `consistsOf` some building partitions, a building partition `consistsOf` some building spaces, a building zone `consistsOf` some building spaces. The Fiemser ontology also describes the devices (`Device` class) used in the building in terms of `HomeEquipment` and `ControlledDevice`.

A `HomeEquipment` is any home appliance or mechanism to increase building energy efficiency, such as `Generator`, `Load`, `Mechanism` and `Storage`. Generators represent devices that provide part of the energy required by the building, for example, `PV` (of type `ElectricalGenerator`) and `Boiler` (of type `ThermalGenerator`). Loads represent devices that consume energy and offer a service to the user, for example, `TV` (of type `ElectricalLoad`) and `Radiator` (of type `ThermalLoad`). Mechanisms represent devices that are installed in the home to increase its energy efficiency, but don not generate or consume energy by themselves, for example, a `Blinder`. Storage devices represent devices that store energy and can be used to provide convenient energy management strategy, for example, `Battery` (of type `ElectricalStorage`) and `Tank` (of type `ThermalStorage`).

A `ControlDevice` represents a device directly connected to the FIEMSER control infrastructure and used to monitor and/or control the environment and its appliances. A control device `consistsOf` some `ControlComponent` that can be a hardware component (`Sensor` or `Actuator` or `CommDevice`) and a software component. An `Actuator` is any actuating hardware installed in a control device, such as a `Dimmer`, `Switch` and `Controller`. A `Sensor` can be a `MeasurementSensor` (e.g., thermostat) or `StateSensor` (e.g., presence). A communication device (`CommComponent`) identifies the communication devices used for data exchange and uses a specific Network protocol (`NetProtocol` class).

Observations

- The Fiemser ontology describes the main classes of the Energy-focused BIM model and WSN-related data that are part of the FIEMSER data model. Although the other 6 models of the FIEMSER data model contain relevant information also, we decided not to include them in the

current version to keep the size of this ontology balanced in comparison with the other ontologies. It is therefore advised to do so as part of future work.
- The source used to create the ontology is a secured pdf from which the information could not be automatically copied. As a consequence, comments that could better explain the ontology may be missing.

5.2.7 FIPA

Ontology title
Fipa: Foundation for Intelligent Physical Agents (FIPA) Device Ontology

Source
'FIPA Device Ontology Specification', document number SC00091E , 3 December 2002, available at http://www.fipa.org/specs/fipa00091/SI00091E.pdf.

Ontology description
The Fipa ontology describes a device ontology that aims at enabling interoperability between software agents, as defined by the FIPA Device Ontology Specification. This ontology can be used by agents when communicating about devices: when agents pass profiles of devices to each other, these profiles can be validated using the information contained in this ontology.

The main class of the ontology is the `Device` class, which defines a device and its general properties. A device has some `InfoDescription`, such as the name, vendor and version of the product under consideration, and has some hardware and software properties. Software properties include the details of the device's operating system (`hasOperatingSystem`), such as its name, vendor and version. Hardware properties are the type of connection that the device uses (`hasConnection`), the amount of memory that it requires (`hasMemory`), the user interfaces offered by the device (`hasUserInterface`), and the type of central processing unit (`hasCPU`). The connection type is expressed in terms of name, vendor and version of the connection provider (`hasConnectionInfo`). The `MemoryTypeDescription` class defines the unit of measure of the memory (`hasMemoryUnit`), and its usage type, namely application, storage, or both application and storage (`hasMemoryUsageType`). The `UIDescription` class defines the information that characterize the screen of the device (`hasScreen`), such as its width (`hasWidth`), height (`hasHeight`), resolution (`hasResolution`), and the measurement units (`hasWidhtHeightUnit`). The ontology also defines the `RequestDeviceInfo` function that can be used in the FIPA framework by an agent to make a query to request the device information contained in the ontology.

Observations
- We have created an OWL version of the FIPA ontology according to the FIPA device ontology specification. This specification refers to some classes defined in other FIPA ontologies, namely the FIPA-Nomadic-Application and FIPA-Agent-Management ontologies. These ontologies are out of the scope of this study and, therefore, have not been translated to OWL. However, our Fipa ontology can be extend to consider the FIPA-Nomadic-Application by using the `AgentPlatform` class, and the FIPA-Agent-Management ontologies by using the `QoS` class.

5.2.8 HYDRA

Ontology Title

Hydra: Heterogeneous physical devices in a distributed architecture (HYDRA) ontology

Source

'Deliverable D6.6 Updated MDA Design Document', version 1.0, 20 August 2009, http://www.hydramiddleware.eu/hydra_documents/D6.6_Updated_MDA_Design_Document.pdf.

Ontology description

There are several ontologies developed in the Hydra project, but for the purpose of this study we are mainly interested in the Device ontology, which consists of the following modules:

- Basic Device Information
- Device Services
- Device Events
- Device Malfunctions
- Device Capabilities and State Machine

The Basic Device Information module represents general device information. The `HydraDevice` is the main ontology class, which is further divided in the `PhysicalDevice` and the `SemanticDevice` classes. Physical and semantic devices share common device properties, such as `deviceId` or `inLocation`, but have different semantic interpretation and behaviour. The `HydraDevice` class refers to the `InfoDescription` class using the `info` property. The `InfoDescription` class contains basic information about device `friendlyName`, manufacturer data, i.e., `manufacturerName` and `manufacturerURL`, and device model data, i.e., `modelName`, `modelDescription` and `modelNumber`. An important part of the basic device information is the representation of device type modelled as sub classes of the `PhysicalDevice` concept, such as `SensorDevice`, `ActuatorDevice`, `MediaDevice` and `MobileDevice`. Further, the `hasEmbeddedDevice` property of the `SemanticDevice` class recursively refers to `HydraDevice` concept. This property enables the creation of models of composite devices, such as in case of the `HeatingSystem` device, which can be, for example, composed of `Thermometer` and `Pump` devices.

Observations

- We did not create the Hydra ontology. We are reusing the description of the ontology in the 'Deliverable D6.6 Updated MDA Design Document' and the OWL files provided by the authors of this document.

5.2.9 KNX

Ontology Title

Knx: KNX ontology

Source

- 'KNX System Specifications Interworking Datapoint Types', Version 01.09.01, 18 September 2014
- 'KNX Advanced Course- Interworking_E1209b'

Ontology description

The Knx ontology represents the several types of data point defined in the KNX specification, namely Datapoint types for common use (`DatapointType4CommonUse` class), Datapoint types for HVAC (`DatapointType4HVAC` class), Datapoint types for Load Management (`DatapointType4LoadManagement` class), Datapoint types for Lighting (`DatapointType4Lighting` class), and Datapoint types for Systems (`DatapointType4System` class). Examples of `DatapointType4CommonUse` are `DPT_ControlDimming`, `DPT_Scaling`, `DPT_Step`, `DPT_Switch` and `DPT_UpDown`. Each data point type has an identifier with an allowed value range.

Combinations of data point types in a device are called functional blocks (`FunctionalBlock` class). Many functional blocks have been standardized by KNX, but we take into account in the Knx ontology only two as example, namely the `DimmerActuatorBasic` and `SunblindActuatorBasic` functional blocks.

The `DimmerActuatorBasic` functional block combines the `DPT_ControlDimming`, `DPT_Scaling` and `DPT_Switch` data point types. This functional block can be in one of the 3 states `{"On", "Off", or "Dimming"}`, which are defined under the `DimmingActuatorBasicState` class. The change from one state to another is triggered by the so-called Events with one of the values listed under the `DimmingActuatorBasicEvent` class.
The `SunblindActuatorBasic` functional block combines the `DPT_Step` and `DPT_UpDown` data point types. This functional block can be in one of the 4 states `{" Stopped", "InMotion", or "StepUp", "StepDown"}`, which are defined under the `SunblindActuatorBasicState` class. The change from one state to another is triggered by events with one of the values listed under the `SunblindActuatorBasicEvent` class.

Observations

- A very large amount of Datapoint types are defined. Unfortunately, we could not represent them all in this initial version of the ontology. However, we have defined the `DatapointType4CommonUse` that were relevant to define the `DimmingActuatorBasic` and `SunblindActuatorBasic` functional blocks, which are used here as an example. The other data points need to be added eventually according to the KNX datapoint type specification used as source of this ontology.
- A very large amount of Functional Blocks are standardized in the KNX specifications (i.e., 157 functional blocks and more coming soon), but in this initial version of the ontology we only represented two of them (`DimmingActuatorBasic` and `SunblindActuatorBasic`) as an

example for the KNX ontology users. Additional functional blocks can eventually be added according to the KNX specifications.

5.2.10 MIRABEL

Ontology Title

Mirabel: Micro-Request-Based Aggregation, Forecasting and Scheduling of Energy Demand, Supply and Distribution (MIRABEL) ontology

Source

'D7.5 MIRABEL-ONE: Initial draft of the MIRABEL Standard', version1.0, 22 December 2011, available at http://wwwdb.inf.tu-dresden.de/miracle/publications/D7.5.pdf.

Ontology description

The Mirabel ontology defines how actors can express in the form of user preferences their energy flexibility for a specific device with respect to amount, time and price. Each device has an energy profile that describes the amount of energy consumed and/or produced over a certain time span. A flex offer is issued by an actor and combines the user preferences with the corresponding device energy profile.

The `User` class represents the person using the device. This person may own the device or not. Depending on the device, the user can take the role of a consumer, producer or prosumer. A user expresses his user preferences with respect to a device. A user `uses` a device and `specifies` some preferences. A `Device` is an electricity consuming and/or producing appliance. Three types of devices can be identified, namely energy production, consumption and storage devices (`ProducingDevice`, `ConsumingDevice` and `StorageDevice` classes, respectively). A `Preference` describes the minimum demand by a user for the electrical consumption/production of a device. A `Preference` is expressed with respect to time, price and amount constraints (consists of `exactly 1 Amount`, `exactly 1 Price`, `min 1 TimePoint`, `max 2 TimePoint`). Time can be expressed as one point in time (`TimePoint`) or as several points in time (`TimeInterval`). `TimePoint` is either expressed as `CalendarTime` or as a `RelativeTimePoint`. The `Price` represents the minimum/maximum price that the user is willing to pay for energy production/consumption. The `Amount` can be an absolute amount (`AbsoluteAmount`) or a percentage of this amount (`RelativeAmount`).

An `EnergyProfile` describes the energy load production and/or consumption of a device over a time span. It specifies the profile in terms of time (`consistsOf exactly 1 TimeInterval`) and indicates whether or not there can be breaks/ interruptions between the end of one interval and the start of the next interval. The energy profile also specifies power and/or energy (`consistsOf max 1 Power`, `max 1 Energy`) in terms of an `Amount`.

An `Offer` combines the user `Preference` and the `EnergyProfile` of a device. It can either be a `SingleOffer` or `CompositeOffer`. A single offer is a `FlexOffer` that consists of a unary expression with only one operand. A composite offer is a `FlexOffer` that consists of a binary expression with 2 operands connected using the conditional elements "AND, OR". Through a `FlexOffer` the flexibility in energy supply and demand can be offered on a marketplace. The

FlexOffer is issued by a FlexEnergyIssuer (this can be the same person as the user) and submitted to a FlexEnergyAcquirer.

Observations
None

5.2.11 OMA Lightweight M2M

Ontology Title
Omalwm2m: Open Mobile Alliance (OMA) Lightweight (LW) Machine-to-Machine (M2M) ontology

Source
'OMA Lightweight Machine-to-Machine Technical Specification', candidate version 1.0, 10 December 2013, available at http://technical.openmobilealliance.org/Technical/release_program/docs/LightweightM2M/V1_0-20131210-C/OMA-TS-LightweightM2M-V1_0-20131210-C.pdf.

Ontology description
The OMA LWM2M architecture is based on a client component, which resides in the LWM2M Device, and a server component, which resides within the M2M Service Provider or the Network Service Provider. Each piece of information made available by the client is a resource. A client may have any number of resources and these resources are organized into objects. Each resource supports one or more operations. The Omalwm2m ontology describes the resources, objects and operations supported by the OMA LWM2M architecture.

An object (Object class) consists of one or more resources (Resource class). Different resources are organized into an object. Each Object has a unique identifier (hasObjectID property), and each Resource has a unique identifier within the object it belongs to (hasResourceID property). Both objects and resources can be mandatory or optional (isMandatory property). Both objects and resources can have a single instance (hasInstance min 0 and hasInstance max 1 constraints) or multiple instances (hasInstance min 0 constraint), represented by the ObjectInstance and ResourceInstance classes, respectively. Each object instance and resource instance have a unique identifier (hasObjectInstanceID and hasResourceInstanceID properties). A Resource supports one or more operations (hasOperation property). Examples of operations are Read, Write, and Execute.

According to the OMA Lightweight Machine-to-Machine Technical Specification, the Omalwm2m ontology defines several type of objects, namely the LWM2MSecurity, LWM2MServer, LWM2MAccessControl, LWM2MDevice, LWM2MConnectivityMonitoring, LWM2MFirmware, LWM2MLocation, and LWM2MConnectivityStatistics objects. Each object has a unique identifier (hasObjectID property). The ontology further details the LWM2MDevice object and its corresponding resources, such as, for example, the Manufacturer, PowerSourceCurrent, and AvailablePowerSource resources, among others. The LWM2MDevice object hasObjectID with value "3", supports a single object instance, and is a mandatory object. The Manufacturer resource hasResourceID with value "0", supports a single resource instance, supports the "Read" operation, and is optional for the LWM2MDevice object.

Observations

- Not all the OMA LWM2M objects as defined in the 'OMA Lightweight Machine-to-Machine Technical Specification' are in the scope of our study. Therefore, we have chosen to include details in our ontology of only the `LWM2MDevice` object and its corresponding resources. Following the same approach, the ontology can be extended with the details of the other objects (`LWM2MSecurity`, `LWM2MServer`, `LWM2MAccessControl`, `LWM2MConnectivityMonitoring`, `LWM2MFirmware`, `LWM2MLocation`, and `LWM2MConnectivityStatistics`), which are currently "open" classes, i.e., these classes are defined in the ontology, but they may be further detailed.

5.2.12 OMS

Ontology Title

Oms: Open Metering System (OMS) ontology.

Source

'Open Metering System Specification Vol.2 – Primary Communication Issue 4.0.2', 27 January 2014, available at http://oms-group.org/fileadmin/pdf/OMS-Spec_Vol2_Primary_v402.pdf.

Ontology description

The Oms ontology is a taxonomy of the devices supported by the Open Metering System (OMS) specification. The `Device` class represents these devices. Each device has a corresponding code (`hasCode` property). There are two type of devices: the devices that are actually supported (`SupportedDevice` class) and the devices that are not certifiable (`NotCertifiableDevice` class). The OMS specification covers the devices under the `SupportedDevice` class, which are further classified in `OMSMeter` and `OMSDevice`. The `OMSMeter` class covers the meter type of devices, such as `ElectricityMeter`, `GasMeter`, `HeatMeter` and `WaterMeter`, among others. The `OMSDevice` class covers other types of devices, such as `Breaker`, `BidirectionalRepeater` and `CommunicationController`, among others. The devices under the `NotCertifiableDevice` class may also be integrated in the Open Metering System. However these devices cannot be approved by the OMS-Compliance Test. Therefore the interoperability for the devices under the `NotCertifiableDevice` class cannot be guaranteed.

The Oms ontology also provides a taxonomy of the DataPoints according to the OMS-Data Point List (OMS-DPL), such as, for example, the current in Ampere (`CA` class), the energy in Joule or Watt hour (`EJ` and `EW` classes, respectively), the temperature in °C (`TC` class), the volume in cubic meters (`VM` class) and the voltage in Volt (`VV` class).

Observations

- The Oms ontology focuses on the type of metering devices and does not consider the architecture of the Open Meter System, such as the Multi Utility Communication Controller (MUC), which is the hardware system used to readout different metering devices and to transfer subsets of this data to AMM back office systems for billing, servicing or other purposes.

– The DataPoint classes are "open" classes, i.e., these classes are defined in the ontology, but they may be further detailed. As future work, the ontology can be extended to take into account the separation of OMS-Datapoints in M-Bus tags and VIB-type lists.

5.2.13 OSGi DAL

Ontology Title

Osgidal: OSGi Device Abstraction Layer (DAL) ontology

Source

'RFC-196 OSGi Alliance Device Abstraction Layer, Draft', 30 January 2014, available at https://raw.githubusercontent.com/osgi/design/a71f2871f4ed0b97c4da79cf756a15876a61a347/rfcs/rfc0196/rfc-0196-DeviceAbstractionLayer.pdf.

Ontology description

The Osgi_dal ontology focuses on the concepts of 'device' and 'function' that are central in the OSGi architecture. A device (Device class) represents a physical device or a functional part of it in the OSGi service registry. A device is characterized by a mandatory unique identifier (hasDeviceUID property) and a set of properties, most of which are optional, namely the device type, such as DVD or TV (hasDeviceType property), model (hasModel property), serial number (hasSerialNumber property), driver (hasDriver property), firmware and hardware vendor and version (hasFirmwareVendor, hasFirmwareVersion, hasHardwareVendor and hasHardwareVendor properties). Moreover, a device has a status (hasStatus mandatory property) that can assume one of the values "Removed", "Offline", "Online", "Processing", "NotInitialized", or "NotConfigured". Optionally, the reason of the current device status can be defined using the hasStatusDetail property, which can assume fixed values, such as "Connecting", "Initializing", or "DeviceBroken", among others. A device can support zero or more functions, which are described by the Function class.

A function is an atomic functional entity that characterizes a device. A function is registered in the OSGi service registry. There are 8 functions defined by OSGi, namely Alarm, BooleanControl, BooleanSensor, KeyPad, Meter, MultiLevelControl, MultiLevelSensor and WakeUp. The ontology also defines several function types (FunctionType class), such as Light, Occupancy and Temperature, which further specifies a certain function. For example, one can have a temperature sensor or an occupancy sensor represented by a BooleanSensor function with Temperature or Occupancy function type, respectively. Each function provides a set of operations and properties (hasPropertyName and hasOperationName properties). For example, the Meter function has an operation "resetTotal" and two properties, namely "current", which contains the current consumption, and "total", which contains the total consumption measured since the last call of the "resetTotal" operation or the device initial run. Finally, the ontology defines units of measure in the UnitOfMeasure class.

Observations

– The ontology defines the status of a device (Status class), but not the status transitions to go from one state to another, namely the dynamic behaviour of the device, which is represented by state diagrams in the OSGi specification.

- Operations and properties that correspond to a certain function are currently defined as `hasPropertyName` and `hasOperationName` properties. Eventually, one could extend the ontology by defining the `Operation` and `Property` classes under which those can be further detailed.
- The ontology includes some example units under the `UnitOfMeasure` class, but more units should to be added for completeness according to the OSGi DAL specification (see page 72 of the 'OSGi Alliance Device Abstraction Layer').
- Events are also part of the OSGi specification, but they are out of scope for our study. However, the ontology could be extended by defining a `FunctionEvent` class and its corresponding properties.

5.2.14 PowerOnt (previously SEIPF)

Ontology Title

PowerOnt: Power Profiling for Intelligent Domotic Environments (imports DogOnt: Ontology Modeling for Intelligent Domotic Environments)

Source

- Dario Bonino, Fulvio Corno, 'DogOnt - Ontology Modeling for Intelligent Domotic Environments', 7th International Semantic Web Conference. October 26-30, 2008. Ed. Springer-Verlag, Lecture Notes on Computer Science, pp. 790-803, available at http://www.cad.polito.it/db/iswc08.pdf
- DogOnt website, Politecnico di Torino, available at http://elite.polito.it/dogont
- PowerOnt ontology, available at http://elite.polito.it/ontologies/poweront/poweront.html

Ontology description

The PowerOnt ontology provides energy consumption information for different appliances in the house using the underlying DogOnt ontology, which models the domotic system of a house supporting intelligent operations. The DogOnt ontology consists of the following main classes: `BuildingThing`, which models available things, either controllable or not; `BuildingEnvironment`, which models the place where things are located; `State`, which models the stable configurations that controllable things can assume; `Functionality`, which models what controllable things can do; and `Command`, which models the way a given device property can be modified (e.g., light intensity) and the values it can assume.

The `BuildingEnvironment` class supports a coarse representation of domestic environments, as whole architectural units, including several types of `Room`, the `Garage` and the `Garden`. The `BuildingThing` concept represents all the elements that can be located or that can take part in the definition of a `BuildingEnvironment`. DogOnt defines a clear separation between objects that can be controlled by a domotic system (`Controllable` class) and all the other objects that can be found in a home (`UnControllable` class). Controllable objects can be appliances (`Appliance` class) or can belong to house plants such as the HVAC3 plant. Appliances and are further subdivided in `WhiteGoods` and `BrownGoods`, according to the EHS taxonomy. House plants include `HVACSystems`, `ElectricSystems` and `SecuritySystems`. Uncontrollable objects are all the home components that cannot be directly controlled by a domotic system. They are mainly subdivided in `Furniture` and `Architectural` elements. Furniture models all the elements usually adopted as furniture like chairs, cupboards, desks. Architectural elements model all the elements that define a living environment such as Walls and Floors. All the objects that are usually referred to as "device", in the DogOnt ontology are objects belonging to the Controllable class. Each device class is associated to a set of different functionalities, by means of the hasFunctionality relationship. Each functionality defines the Commands to modify a given device property (e.g., light intensity) and the values they can assume. Functionalities are divided in different classes on the basis of their goals: ControlFunctionality models the ability to control a device or a part of it. NotificationFunctionality represents the ability of a device to autonomously advertise its internal state and in particular the ability of detecting and signalling state changes. QueryFunctionality encompasses the capabilities of a device to be queried, or polled, about its condition, e.g., failure and internal state values. States are classified according to the kind of values they can assume: continuously changing qualities are modelled as ContinuousStates, while qualities that can only assume discrete values (e.g., On/Off, Up/Down, etc.) are classified as DiscreteStates.

The PowerOnt ontology adds the PowerConsumption class, which encodes the power consumed by the appliances defined in DogOnt in a given state (StateValue class).

Observations

- We did not create the DogOnt and PowerOnt ontologies. We are reusing the OWL version that was provided to us by the authors of the 'DogOnt - Ontology Modeling for Intelligent Domotic Environments' article.

5.2.15 SEEMPubs

Ontology Title

Seempubs: Smart Energy Efficient Middleware for Public Space (SEEMPubS) ontolology

Source

'Deliverable D5.1, Data Format Definition', version 1.0, 30 September 2012, available at http://seempubs.polito.it/images/stories/documents/WP5/D.5.1.pdf .

Ontology description

The Seempubs ontology describes the sensors and data that have been used in the use cases of the SEEMPubS project to control the building services, and monitor the indoor conditions and energy consumptions in some rooms of the Politecnico di Torino Campus and the Valentino Castle in Italy. The Sensor class represents the different type of sensors that have been used, namely Controller, IndoorTemperatureHumiditySensor, IndoorTemperatureSensor, LightSensor, OccupancySensor, OutdoorTemperatureSensor, PowerMeter4Lightingsystem, PowerMeter4Appliance, SuppliedAirTemperatureSensor, Switch and Thermostat. The data recorded in the use cases can be classified as related to indoor comfort conditions, related to energy consumption (electrical or thermal consumption), and related to the use of spaces and building services. Therefore, each sensor belongs to one of the categories: ComfortCondition, EnergyConsumption or UseOfSpaceAndBuildingService. Moreover, a sensor has a certain position in the room (hasPosition property) in which a number of sensors is positioned (hasSensorNumber property), is associated to a certain protocol, such as, for example, "EnOcean" or "BACnet" (hasProtocol property), and requires a certain communication time to transmits its data (hasCommunicationTime property).

Each sensor measures some quantity (MeasuredQuantity class). For example the IndoorTemperatureSensor measures Temperature , the OccupancySensor measures Presence or Absence, and the Controller measures FanCoilStatus1, FanCoilStatus2, or FanCoilStatus3. The MeasuredQuantity class can have a unit (UnitOfMeasure class), a mode (OperationMode class) and a status (Status class). For example, Temperature has unit CelsiusDegree, Presence has mode Present, and FanCoilStatus1 has status FanVelocity1. Each sensor processes data in certain terms, such as, for example, average values or individual single data, which are enumerated in the DataProcessing class. Each sensor also represents data in a certain manner, for example, daily, monthly, according to an annual trend or a cumulative frequency, as enumerated in the DataRepresentation class.

The LightSensor class provides an example of how all the classes and properties mentioned above can be instantiated for a specific sensor.

Observations

– Only the LightSensor class is fully detailed with the values provided in the source of the ontology ('Deliverable D5.1, Data Format Definition'). The other type of sensors need to be detailed analogously (i.e., Controller, IndoorTemperatureHumiditySensor, IndoorTemperatureSensor, OccupancySensor, OutdoorTemperatureSensor, PowerMeter4Lightingsystem, PowerMeter4Appliance, SuppliedAirTemperatureSensor, Switch and Thermostat).

5.2.16 SEP2

Ontology Title

Sep2: ZigBee Smart Energy Profile 2.0 (SEP2) ontology

Source

' Zigbee Alliance/HomePlug Alliance Smart Energy Profile 2 Application Protocol Standard, ZigBee Public Document 13-0200-00', April 2013, available at http://www.zigbee.org/Standards/ZigBeeSmartEnergy/ZigBeeSmartEnergy20Standard.aspx.

Ontology description

The ZigBee SEP-2 ontology is a taxonomy that represents the SEP-2 resources and function sets. Resources are classified in resources that provide operational information or services to manage and support the end devices of an SEP-2 network (SupportResource class), resources that provide general purpose and non-domain specific functionality (CommonResource class), and resources that are specific to the domain of Smart Energy (SmartEnergyResource class). Examples of support resources are represented by the EndDeviceResource and DeviceStatusResource classes, common resources by the DeviceInformationResource, PowerStatusResource and TimeResource classes, and Smart energy domain resources by the MeterReadingResource class. Each resource can be further detailed with its specific properties. The ontology further details the DeviceStatusResource and the DeviceInformationResource class. For example, the DeviceStatusResource is characterized by the time at which the reported values were recorded (hasChangedTime property), the number of times that the device has been turned on (hasOnCount property), the device operational state (hasOpState property), and the total time device has operated (hasOpTime property).

A function set (FunctionSet class) is a logical grouping of resources that cooperate to implement SEP-2 features, such as, for example, metering (MeteringFunctionSet class). Therefore, a function set groups a number of resources (groups property), while a resource is grouped in a certain function set (isGroupedIn inverse property). For example, the EndDeviceResourceFunctionSet class groups the EndDeviceListResource, EndDeviceResource, RegistrationResource and DeviceStatusResource classes.

Under the TypesPackage class, the ontology represents some data types that are relevant to describe the considered resources, such as DeviceCategoryType (e.g., Water Heater, Sauna, Hot tub, Smart Appliance, Irrigation Pump, etc.), PowerSourceType (e.g., battery, local generation, emergency, etc.) and UnitType (e.g., kWh, kW, Cubic Meters, etc.).

Observations

- The Sep2 ontology presents examples of resources, function sets and package types that are in the scope of our study. The ontology also describes in detail the DeviceStatusResource and theDeviceInformationResource classes, which can be used as example to further detail other resources defined in the SEP 2 specification. In fact, the SEP 2 specification contains a large number of resources, function sets and package types that are not considered here, but can be eventually added to extend the current version of the ontology.

5.2.17 SmartCoDE

Ontology Title

Smartcode: Smart Control of Demand for Consumption and Supply to enable balanced, energy-positive buildings and neighbourhoods (SmartCoDE) ontology

Source

'Deliverable D1.1.2 -Model of local energy resource cluster', 31 December 2012, available at https://www.fp7-smartcode.eu/system/files/page/d-1.1.2.pdf.

Ontology description

The Smartcode ontology presents a classification of Energy using Products (EuPs) into seven categories, namely variable services (VARSVC class), thermal services (THMSVC class), schedulable services (SCDSVC class), event-timeout services (ETOSVC class), charge control (CHACON class), complete control (COMCON class), and custom control (CUSCON class). These products have some parameters, such as Configuration, OnlineInput and SensorInput. Each product is characterized by an energy management strategy (hasEnergyManagementStrategy property) and its cost profile can be of interest of not for energy management purposes (isCostProfileInteresting property).

The VARSVC class includes appliances that provide a user-variable service that is balanced with sensor input. For example, Blind, DimmableLighting and LightingIlluminaceControlled are variable services included in this class.

The THMSVC class includes appliances that provide an inert, thermal service that can serve as a virtual storage. For example, Freezer, Heating and WaterBoiler are thermal services included in this class.

The SCDSVC class includes appliances that provide a service that can be scheduled within a certain time-frame. For example, BakingMachine, Dryer and WashingMachine are schedulable services included in this class.

The ETOSVC class includes appliances that are controlled by sensor events and time-outs. For example, LightingPresenceControlled is an event-timeout service included in this class.

The CHACON class includes appliances that charge a possibly removable device. For example, BatteryCharger, Emergency and HandHeldVacuum are charge controls included in this class.

The COMCON class includes appliances that charge a possibly removable device, like CHACON, but the usage of the charged power can also be controlled. For example, RobotVacuum is a charge control included in this class.

The CUSCON class includes appliances that do not fit into other classes or have too high user interaction to be controllable. For example, Hifi, Oven and PC are appliances included in this class.

Observations

- The enumerations under the Parameter class contain the values defined in the source of the ontology ('Deliverable D1.1.2 -Model of local energy resource cluster'). However, these enumerations can be extended with new values, if necessary.

5.2.18 UPnP

Ontology Title

Upnp: Universal Plug and Play (UPnP) ontology

Source

'UPnP Device Architecture 1.1,', 15 October 2008, http://upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf.

Ontology description

The UPnP ontology represents the devices and services defined by the UpnP device architecture specification. A device (Device class) represents a logical device and is a container that embeds one or more services (Service class)and may embed other logical devices (Device class). A device must have a description (DeviceDescription class), which contains all relevant information about the device filled in by the vendor, such as manufacturer name, model name, model number, serial number, and URLs for control, eventing, and presentation, among others. The device description also includes the services corresponding to that specific device (hasService property). A service exposes some actions (Action class), namely the commands supported by the service, and state variables that characterise the status of the service (StateVariable class). Actions and state variables are specified in the service description (ServiceDescription class). An action has arguments (Argument class), which are parameters that can be input or output of a service, and may have a return value. A state variable can trigger events (`Event` class) as notification of one or more changes in the state variables exposed by a service.

The ontology instantiates examples for the `SolarBlindProtection`, `HVAC_System` and `HVAC_ZoneThermostat` devices with their corresponding services. Consider the `SolarProtectionBlind` instance of the `Device` class, which embeds the `TwoWayMotionMotor` instance of the `Service` class. Some manufacturer details of the `SolarProtectionBlind` device instance are mandatory and should be filled in, but they are left out from this version of the ontology for the sake of simplicity. The `TwoWayMotionMotor` service instance contains a number of corresponding actions, such as `Close`, `Lock`, `Open`, `Stop`, `SetPosition`, etc., and some state variables, such as `OperationMode`, `Position` and `ServiceLocked`.

Observations

- The UPnP ontology presents instance of devices, services, actions, arguments and state variables that should be used as an example to further extend the ontology according to the source, namely the 'UPnP Device Architecture 1.1' document.

5.2.19 W3C SSN

Ontology Title

Ssn: Semantic Sensor Network (SSN) Ontology

Source

Semantic Sensor Network Ontology, available at http://www.w3.org/2005/Incubator/ssn/ssnx/ssn.

Ontology description

The SSN ontology is an OWL ontology that provides a framework to describe sensors, observations and related concepts. The official description of the ontology from W3C is available at http://www.w3.org/2005/Incubator/ssn/ssnx/ssn. The SSN ontology does not describe domain concepts, such as time and locations, since these concepts are intended to be included from other ontologies via OWL imports. A sensor is a specific device whose purpose is to report measurements and observation real world phenomena. A sensor is different in nature from other types of devices such as actuators, because of its event based behaviour and the temporal relationships that need to be considered. The SSN ontology is a basis for reasoning about the measurements that can ease the development of advanced applications. For instance, when reasoning about sensors, constraints such as power restriction, limited memory, variable data quality need to be taken into account. It is possible to reason either about individual sensors as well as about the connection of a number of sensors.

The SSN ontology is composed by several modules. Some modules in the scope of our study are:

- Skeleton module, which represents
    - `FeatureOfInterest`, i.e., an abstraction of real world phenomena, such as thing, person, event;
    - `Observation`, i.e., a Situation in which a Sensing method has been used to estimate or calculate a value of a Property of a Feature Of Interest;

- Property, i.e., an aspect of an entity that is intrinsic to and cannot exist without the entity and is observable by a sensor;
    - Sensing, i.e., a process that results in the estimation, or calculation, of the value of a phenomenon);
    - Sensor, i.e., any entity that can follow a sensing method and thus observe some Property of a Feature Of Interest. Sensors may be physical devices, computational methods, a laboratory setup with a person following a method, or any other thing that can follow a Sensing Method to observe a Property;
    - SensorInput, i.e., an Event in the real world that triggers the sensor;
    - SensorOutput, i.e., a sensor outputs a piece of information (an observed value), the value itself being represented by an Observation Value),
    - Stimulus (an Event in the real world that 'triggers' the sensor. The properties associated to the stimulus may be different to eventual observed property. It is the event, not the object that triggers the sensor)
- Measuring module, which represents SensingDevice, SensorDataSheet;
- Measuring Capability module, which represents Accuracy, DetectionLimit, Drift, Frequency, Latency, MeasurementCapability, MeasurementProperty, MeasurementRange, Precision, Resolution, ResponseTime, Selectivity, Sensitivity;
- Data module, which represents ObservationValue;
- Time module, which represents end Time and startTime;
- Constraint Block module, which represents Condition;
- Device module , which represents Device;
- Energy Restriction module, which represents BatteryLifetime, OperatingPowerRange.

Observations
- We did not create the SSN ontology. We are reusing the OWL version that is available on the W3C website.

5.2.20 Z-Wave
Ontology Title
Zwave: Z-Wave Application Layer ontology

Source
' Z-Wave Technical Basics - Chapter 4: Application Layer', 1 June 2011, available at http://www.domotiga.nl/attachments/download/1075/Z-Wave%20Technical%20Basics-small.pdf.

Ontology description
The Z-Wave ontology covers the application layer of the 3-layer general model of wireless communication defined by Z-Wave. This application layer defines the messages to be exchanged, such as switching a light or increasing the temperature of a heating device. The Z-Wave ontology is a taxonomy of the supported type of devices (i.e., basic, generic or specific), the product categories to which these devices belongs to, and the type of functions, or commands, supported by these devices. Each Device class belongs to a ProductCategory class, such as ElectricalDimmer,

ElectricalSwitch, ThermostatControl, MotorControl and Sensor. Moreover, devices can be classified in basic devices (BasicDevice class), namely the basic category to which every device must belong, generic devices (GenericDevice class), which allows to specify the general function common to a certain type of devices, and specific devices (SpecificDevice class), which allows to further specialize the functions of a certain generic device. For example, each basic device must be a Controller, Slave or RoutingSlave. Examples of generic devices are a thermostat, meter, and alarm sensor, which are represented by the ThermostatGeneric, MeterGeneric and AlarmSensorGeneric classes, respectively. The generic device thermostat can be further specialized in the ThermostatGeneralV2, SetbackScheduleThermostat, and SetbackThermostatclasses, which are examples of specific devices.

The ontology further represent the commands supported by the Z-Wave devices under the Command class. This class enumerates the commands supported by the standard according to the Annex A of the source used to create our ontology (namely, the ' Z-Wave Technical Basics' document). In case the Z-Wave device is assigned to a SpecificDevice class, it must support a set of mandatory commands as functions of this specific device class, (supportsMandatoryCommand property). Besides the mandatory commands, Z-Wave devices can further support further optional commands (supportsMandatoryCommand property), which may be useful, but the standard does not enforce the implementation of these commands.

Observations

The ontology is a rather simple taxonomy of devices and commands that was derived from the only publicly available document that we could find, which is not the official protocol specification (not available for free). Therefore, this ontology is intended as an initial representation of the main concepts defined by Z-wave and should be more accurately extended according to the original specification.

6. Mappings

The goal of the reference ontology that resulted from task 3 is to explicitly specify recurring core concepts in the smart appliances domain, the relationships between these concepts, and mappings to other concepts used by different assets/standards/models. These mappings allow translation from the reference ontology to specific assets, reducing the effort of translating from one asset to another, since the reference ontology requires one set of mappings to each asset, instead of a dedicated set of mappings for each pair of assets. Figure 39 shows the role of the reference ontology in the mapping.

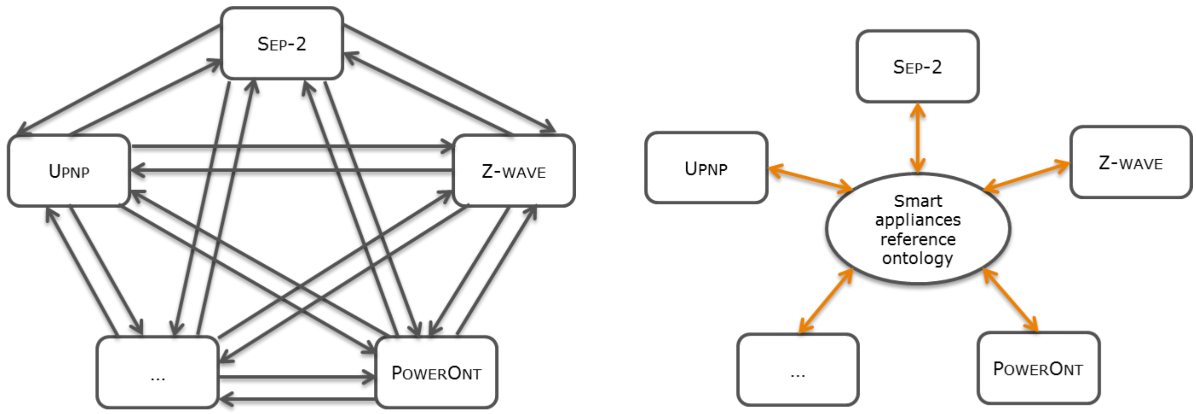


Figure 39: The role of the SAREF ontology in the mapping among different assets

We developed the reference ontology in parallel with task 2 in order to build the reference ontology incrementally while creating individual ontologies for the specific assets. In this way, not only were we able to include relevant concepts as soon as they turned out to be relevant for (several of) the specific assets, but we could also have a way to involve the expert group and stakeholders for validation in an early phase of the development of the reference ontology, instead of presenting our results only at the end of task 3. Therefore , when creating the ontologies for the specific assets in task 2, we identified relevant concepts in the scope of this study that could be part of the reference ontology. An initial proposal is to consider the following concepts:

- Device
- Device category
- Function
- Function category
- Service
- Command
- Parameter
- Mode/Status
- Energy profile
- Energy
- Power
- Time/Duration
- Building
- Sensor
- Actuator
- Meter
- Load
- Storage
- Generator
- Unit of Measure

These concepts have not been organized in any hierarchical relationship, nor relationships among them have been defined yet. These concepts should be considered as a means to present the mappings shown in this chapter (see mapping table on the next page), and as an input for early discussion with the expert group and stakeholders about what could be included in the reference ontology. The criteria used to select these concepts were:

1) whether a concept was recurring among several of the assets for which we created ontologies, showing therefore a certain (shared) degree of relevance for the smart appliances domain;
2) whether the same concept was in the scope of our study, as laid out in section 2.1. To facilitate this task, we used the recurring concepts initially proposed in task 1.

Please note that this selection represents the basis which has been used as a start. In chapter 7 additional concepts that are considered important have been added.

The following table presents an initial mapping of these concepts onto the ontologies described in this document. The table shows only the presence or absence of a certain concept and it is intended to give an overview to the reader in a visual and intuitive manner. The 'D-S4 - SMART 2013-0077 - Smart Appliances - Mapping SAREF to short list assets.xlxs'[68] file presents, the same mappings in more detail.

[68] See https://docs.google.com/file/d/0B2nnxMhTMGh4UnVFMTh1S2R2cGc

	DEVICE	SERVICE	DEVICE CATEGORY	FUNCTION CATEGORY	FUNCTION	COMMAND	PARAMETER	MODE/STATUS	ENERGY PROFILE	ENERGY	POWER	TIME/DURATION	BUILDING	SENSOR	ACTUATOR	METER	LOAD	STORAGE	GENERATOR	UNIT Of MEASURE
Dect_ule	x	x		x	x	x	x				x	x		x	x	x				
DogOnt/Power	x			x	x	x		x			x		x							x
Echonet	x		x		x			x					x	x						
eDiana	x	x				x								x	x		x	x	x	
Enocean	x		x	x	x									x		x				
Fan_Fpai	x		x	x	x				x	x	x	x					x	x	x	x
Fiemser	x							x					x	x	x		x	x	x	
Fipa	x																			
Hydra	x													x	x					
Knx	x			x		x		x							x					
Mirabel	x								x	x	x	x					x	x	x	x
Omalwm2m	x					x	x													
Oms	x									x	x	x				x				x
Osgi_dal	x			x	x	x	x	x						x		x				x
Seempubs	x		x					x		x	x		x	x						x
Sep2	x	x			x			x			x	x				x				x
Smartcode	x		x				x										x	x	x	
Upnp	x	x				x	x	x												
W3C_ssn	x											x		x						x
Zwave	x		x	x	x	x								x	x					

7. Smart Appliances REFerence (SAREF) ontology

7.1 Principles

The Smart Appliances REFerence (SAREF) ontology is conceived as a shared model of consensus that facilitates the matching of existing assets in the smart appliances domain, reducing the effort of translating from one asset to another, since the SAREF ontology requires one set of mappings to each asset, instead of a dedicated set of mappings for each pair of assets.

From the analysis realized in the D-S1 and D-S2 deliverables, we could conclude that different assets share some recurring, core concepts, but they often use different terminologies and adopt different data models to represent these concepts. Using the SAREF ontology, different assets can keep using their own terminology and data models, but still can relate to each other through their common semantics. In other words, the SAREF ontology enables semantic interoperability in the smart appliances domain.

The SAREF ontology explicitly specifies recurring core concepts in the smart appliances domain, the main relationships between these concepts, and axioms to constrain the usage of these concepts and relationships. We have created the SAREF ontology based on the following fundamental principles:

- *Reuse and alignment* of concepts and relationships that are defined in existing assets. Since a large amount of work was already being done in the smart appliances domain, we have not invented anything new, but harmonized and aligned what was already there. The SAREF ontology is based on the core concepts that in the previous deliverables were identified as especially relevant to describe the existing assets. Despite the heterogeneity of these existing assets, when considering their semantic coverage, we could identify three main trends with focus on:
    1) devices, sensors and their specification in terms of functions, states and services,
    2) energy consumption/production information and profiles to optimize energy efficiency, and
    3) building related semantic models.

  In the SAREF ontology and the rest of this deliverable, we call these trends, *function-related*, *energy-related* and *building-related*, respectively. The SAREF ontology includes not only the necessary concepts and relationships to characterize these trends individually, but also to link these trends to each other. For example, the concept of *building space* links function-related assets to building-related assets, since a device designed to accomplish a certain function is located in a specific room of the home or office in a building. Another example is the concept of *profile* that links function-related assets to energy-related assets, since a device designed to accomplish a certain function can be associated with a certain energy/power profile that can be used for energy optimization purposes.
- *Modularity* to allow separation and recombination of different parts of the ontology depending on specific needs. The SAREF ontology provides building blocks that can be combined to accommodate different needs and points of view. The starting point is the concept of *device*, which is actually common to all assets considered in this study, although some assets may refer to it with different names, such as *resource* or *product,* but we provide mappings for that. For example, a "switch" is a device. A device is always designed to accomplish one or more *functions*, therefore, the SAREF ontology offers a lists of basic functions that can be eventually combined in

order to have more complex functions in a single device. For example, the switch mentioned above offers an actuating function of type "switching on/off". Each function has some associated *commands*, which can also be picked up as building blocks from a list. For example, the "switching on/off" function is associated with the commands "switch on", "switch off" and "toggle". Depending on the function(s) it accomplishes, a device can be found in some corresponding *states* that are also listed as building blocks, so that it is easy and intuitive to combine devices, functions and states. The switch considered in our example can be found in one of the two states "on" or "off". The SAREF ontology also provides a list of *properties* that can be used to further specialize the functioning of a device. For example, a "light switch" specializes the more general "switch" described above for the purpose of controlling the "light" property. An extensive explanation of the SAREF ontology, its classes and relationships is presented in the next section and is available online at http://ontology.tno.nl/saref/.

- *Extensibility* to allow further growth of the ontology. Different stakeholders can specialize the SAREF concepts according to their needs and points of view, add more specific relationships and axioms to refine the general (common) semantics expressed in the reference ontology, and create new concepts, as long as they explicitly link these extensions to at least one existing concept and/or relationship in the SAREF ontology. The minimum requirement is that any extension/specialization must comply with the SAREF ontology.
- *Maintainability* to facilitate the process of identifying and correcting defects, accommodate new requirements, and cope with changes in (parts of) the SAREF ontology. According to the extensibility criterion mentioned above, a new module/ontology can be created to further extend/specialize concepts of the SAREF ontology, but according to the maintainability criterion the creator of this module is responsible for its maintenance and versioning, independently from the SAREF ontology. Therefore, the maintenance of new modules is distributed to the creators of these modules. In contrast, in order to avoid inconsistency and confusion, the maintenance of SAREF is centralized to a single party (i.e., TNO until the end of the project in March 2015 and probably ETSI later) who also takes care of aligning SAREF with new modules when necessary.

7.2 Ontology creation process

Towards the creation of the SAREF ontology we have taken the following steps:

1) We have assessed various additional assets suggested during the stakeholders' workshops in Brussels and Sophia Antipolis (see [40]), and we concluded that in addition to the assets short-listed in D-S1, CENELEC, ZigBeeHA and Adapt4EE should also considered in the creation of the reference ontology. ZigBeeHA[69] and Adapt4EE were expressed in OWL, which allowed us to include them straightforwardly in our catalogue of OWL ontologies, while CENELEC only provided a pdf specification with associated XSDs. It is a major undertaking to translate the CENELEC specification to an OWL ontology, and we advise this to be done in future work. Nevertheless we were able to take the most relevant content of the CENELEC specification into account when constructing SAREF.
2) We have (qualitatively) validated the usability of our modular approach of using building blocks to create the SAREF ontology with some stakeholders (representatives of CENELEC, ETSI M2M,

[69] We use the ZigBeeHA ontology based on DogOnt provided by the Politecnico di Torino at http://elite.polito.it/index.php/research/research-topics/35-dogont?showall=&start=2

and HGI) in a dedicated session organized after the 2nd stakeholders workshop in Sophia Antipolis. The result was that a reference ontology built with such modularity in mind seems to be intuitive and well understood by different stakeholders. Moreover, from our analysis of the existing assets we noticed that several of these assets use a similar modular approach for combining devices, functions and commands, e.g., DogOnt (PowerOnt), OSGi DAL, CENELEC, DECT ULE, KNX, SeemPubs, UpnP and Zwave. Therefore, the building blocks of the SAREF ontology should be intuitive for these stakeholders.

3) We have performed an experiment in collaboration with Jerome Euzenat, member of the Smart Appliances expert group, based on the work he has carried out in the context of the READY4SmartCities project[70] (http://www.ready4smartcities.eu/). In this experiment, automatic mappings were performed using dedicated software for ontology matching[71] to support the manual mappings we have provided in the D-S2 deliverable. This experiment has taken as input the 20 ontologies in D-S2 (https://sites.google.com/site/smartappliancesproject/ontologies) and has produced some interesting preliminary matching results[72], which are hosted on the INRIA server (http://al4sc.inrialpes.fr/). Some of these results showed that the DogOnt, OSGi DAL, Fiemser and Seempubs ontologies present the highest number of exact matching among each other (see http://al4sc.inrialpes.fr/onid/1420470114201/6506). Therefore, these assets could be used as a solid common basis for creating our reference ontology. The results obtained with the automatic ontology matching were then checked against the *D-S2-SMART 2013-0077-Smart Appliances-Appendix A-Mappings.xlsx* file, consisting of the mappings we have derived manually in task 2. The conclusion was that this experiment validates the results we presented in the D-S2 deliverable, supporting our mappings in *D-S2-SMART 2013-0077-Smart Appliances-Appendix A-Mappings.xlsx* and the choice of core concepts proposed as basis to build the reference ontology. Unfortunately, there are no further resources in the Smart Appliances project to elaborate on these automatic mappings, but they can and should be further explored in a follow-up of the Smart Appliances project to provide tools for stakeholders for automatic mapping using the SAREF ontology.

4) We created the SAREF ontology starting from the core concepts presented in D-S2, namely: *Device, Device category, Function, Function category, Service, Command, Parameter, Mode/Status, Energy profile, Energy, Power, Time/Duration, Building, Sensor, Actuator, Meter, Load, Storage, Generator, Unit of Measure.* We have created explicit definitions in natural language for these concepts and, in parallel, we have organized them in hierarchical (vertical) relationships and defined horizontal relationships among them. We have also changed some names and refined some of these concepts in subclasses.

[70] READY4SmartCities was considered here since it presents similarities to the Smart Appliances project, although it focuses on another domain (i.e., smart cities). READY4SmartCities intends to increase awareness and interoperability for the adoption of ICT and semantic technologies in energy system to obtain a reduction of energy consumption and CO2 emission at smart cities communities level through innovative relying on RTD and innovation outcomes and ICT-based solutions. Similarly to our project, READY4SmartCities investigated and identified vocabularies and ontologies related to the domain of interest, and provided mappings among them. These mappings were created using automatic tools for ontology matching, instead of manually relating concepts from different assets, as we have done in the smart appliances project.

[71] Exact match algorithm, LogMapLite, YAM++

[72] http://al4sc.inrialpes.fr/onid/1420470368391/9235 => Smart appliances max-aggregated alignment network with exact match, LogMap, YAM++
http://al4sc.inrialpes.fr/onid/1420470148730/6339 => Smart appliances alignment network with YAM++
http://al4sc.inrialpes.fr/onid/1420470114201/6506 => Smart appliances alignment network with exact match

For example:

- the *Mode/Status* concept has been renamed as “State”;
- the *Time/Duration* concept has been renamed as “Time” and then refined in the two concepts of “Instant” and “Interval”;
- The *Sensor*, *Actuator*, and *Meter* concepts have been moved as subclasses of “Device category” (under the “Function-related” category), and also used as basis to create subclasses of the class “Function”;
- The *Load*, *Storage*, and *Generator* concepts have been moved as subclasses of “Device category” (under the “Energy-related” category);
- The concept of “Property” has been introduced to represent anything that can be sensed, measured or controlled in households, common public buildings or offices, such as “Energy”, “Power”, “Temperature”, “Humidity”, and so forth.
  - We have moved the *Energy* and *Power* concepts as subclasses of “Property”
- The concept of “Commodity” has been introduced to represent homogenous goods traded in bulk on an exchange and available at our homes such as “Electricity”, “Gas” and “Water”.
- The *Parameter* concept has been replaced by the two relations “has input Parameter” and “has output Parameter” that characterize the “Service” concept, which must specify the input and output parameters necessary for its operation.

The documentation of the SAREF ontology is available at http://ontology.tno.nl/saref and shows the complete list of concepts and their definitions.

5) In the process of creating the SAREF ontology, we have iteratively checked our intermediate results against the assets in the second version of the short list, mainly using the mappings in the *D-S2-SMART 2013-0077-Smart Appliances-Appendix A-Mappings.xlsx* file, in order to guarantee the link of the reference ontology with the existing assets. DogOnt, OSGi DAL and CENELEC were especially useful for creating the function-related part of the SAREF ontology, SSN for creating the part related to the sensing function and the observation of properties, Fiemser for defining the building-related part, while Fanfpai, Mirabel, PowerOnt and CENELEC provided support especially for creating the energy-related part of the SAREF ontology. Notice that this does not mean that we have neglected the other assets not mentioned above: we have extensively used them all and we acknowledge the value and contribution to the reference ontology in one way or another of all assets in our (extended) short list.

7.3 SAREF

The SAREF ontology focuses on the concept of device, which we define in the context of the Smart Appliances study as *"a tangible object designed to accomplish a particular task in households, common public buildings or offices. In order to accomplish this task, the device performs one or more functions*". Examples of devices are a light switch, a temperature sensor, an energy meter, a washing machine. A washing machine is designed to wash (task) and to accomplish this task it performs the start and stop function. The `saref:Device` class and its properties are shown in Figure 40.

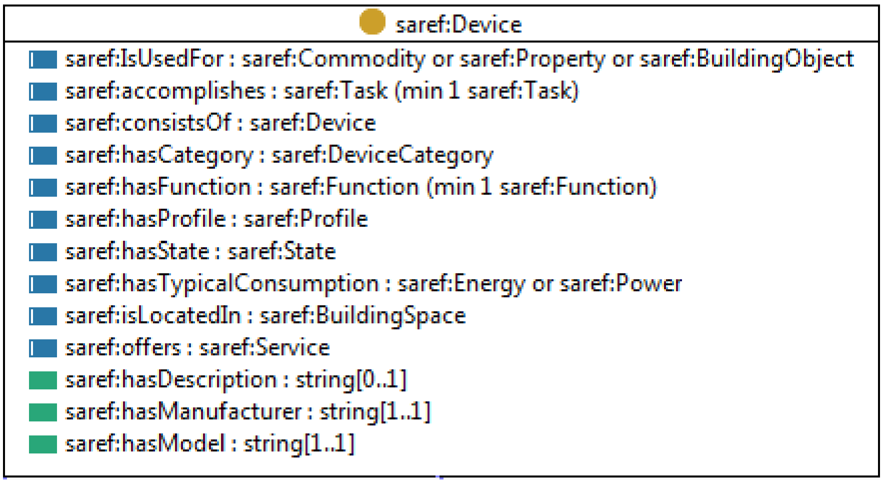


Figure 40: Device class and its properties

A `saref:Device` must have some properties that uniquely characterize it, namely its model and manufacturer (`saref:hasModel` and `saref:hasManufacturer` properties, respectively). Optionally, a description of the device can also be provided (`saref:hasDescription` property). These properties are depicted in Figure 40 using green rectangles that represent *OWL Datatype properties*, which are properties that relate a class (the `Device` class here) to data values, namely a *string* data value in this example. In contrast, *OWL Object properties* are represented using blue rectangles and relate a class to another class. For example, the `saref:isLocatedIn` object property in Figure 40 relates the `saref:Device` class to the `saref:BuildingSpace` class, whereas a building space defines the physical spaces of the building where a device is located, such as a kitchen or a living room. Figure 41 shows the `saref:BuildingSpace` class and its properties.

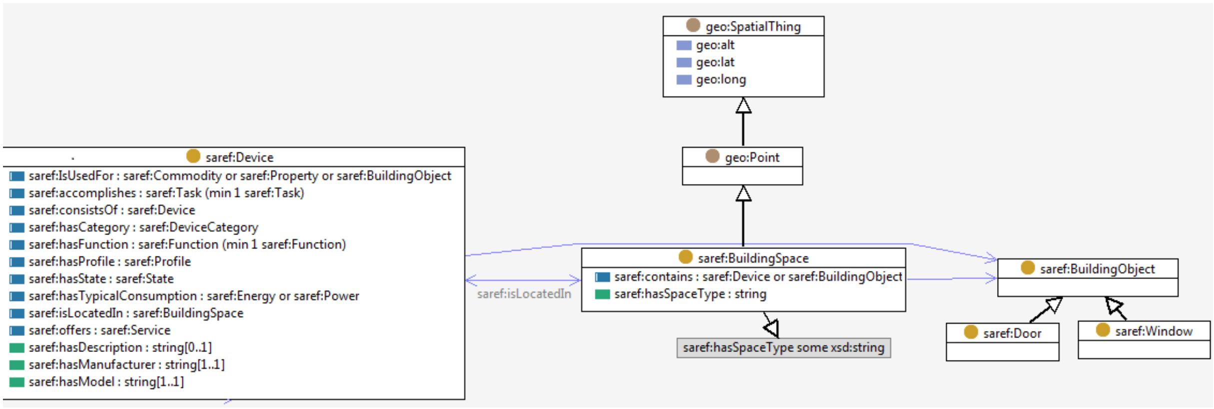


Figure 41: Building Space and Building Object classes

A building space contains devices or building objects (the `saref:BuildingObject` class), where building objects are objects in the building that can be controlled by devices, such as doors or windows that can be automatically opened or closed by an actuator. A building space has also a `saref:hasSpaceType` property that can be used to specify the type of space, for example, the living room or the bedroom. The `saref:BuildingSpace` class provides the link to the FIEMSER model that describes building related concepts, therefore, there is no need to further elaborate on these concepts in SAREF since they are covered elsewhere (see the FIEMSER ontology at https://sites.google.com/site/smartappliancesproject/ontologies/fiemser-ontology). Moreover, a building space is a `geo:Point` characterized by a certain altitude, latitude and longitude, which are provided by the W3C WGS84 geo positioning vocabulary[73] that we have imported in the SAREF ontology. Notice that the WGS84 geo vocabulary is referred to using the `geo:` prefix, which distinguish it from the classes and properties of the SAREF ontology, which are referred to using the `saref:` prefix.

The `saref:hasCategory` object property in Figure 40 relates the `saref:Device` class to the `saref:DeviceCategory` class, which provides a way to classify devices into certain categories. Notice that when analyzing the semantic assets in task 1 we have identified three main trends in the context of the Smart Appliances study with focus on 1) devices, sensors and their specification in terms of functions, states and services, 2) energy consumption information and profiles to optimize energy efficiency, and 3) building related data models. Therefore, according to these trends, we suggest in the SAREF ontology a classification of devices in three main categories that we have called `saref:FunctionRelated`, `saref:EnergyRelated` and `saref:BuildingRelated`, respectively. These categories are shown in Figure 42.

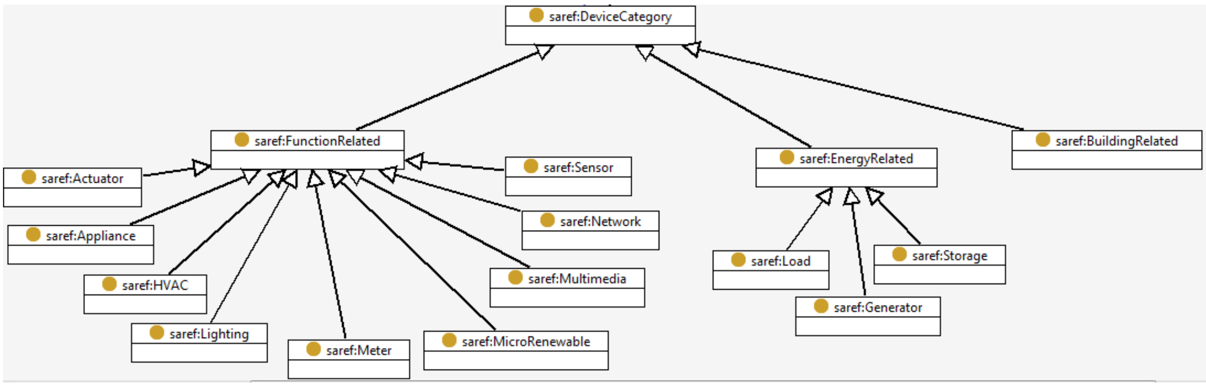


Figure 42: Device Category class

Depending on which trend a certain semantic asset focuses, this asset can be assigned to one of these categories. For example (see [40] and http://sites.google.com/site/smartappliances/ontologies for links to their respective ontologies), Echonet, EnOcean, OSGi DAL, SEP2, and UPnP could identify their devices with the category `saref:FunctionRelated`, FAN and Mirabel could be assigned to the category `saref:EnergyRelated`, while FIEMSER devices would better fit under the category `saref:BuildingRelated`. Moreover, some assets can belong to several categories, for example, PowerOnt[74] and CENELEC could be assigned to both the `saref:FunctionRelated` and

[73] http://www.w3.org/2003/01/geo/wgs84_pos
[74] https://sites.google.com/site/smartappliancesproject/ontologies/dogpower-ontology

saref:EnergyRelated categories. In any case, the assignment of devices provided by specific assets to a certain category is not mandatory and is completely flexible since the asset's owners are free to define a new category as a subclass of saref:DeviceCategory that suits better to their point of view.

The SAREF ontology is conceived in a modular way in order to allow the definition of any device from pre-defined building blocks, based on the function(s) that the device is designed for and the purpose for which it is used. Therefore, Figure 40 shows that a saref:Device must accomplish at least one function (saref:hasFunction min 1 saref:Function), and can be used for (saref:isUsedFor property) the purpose of i) offering a commodity, such as saref:Water or saref:Gas; ii) sensing, measuring and notifying a property, such as saref:Temperature, saref:Energy and saref:Smoke, respectively; or iii) controlling a building object, such as a saref:Door or a saref:Window. Moreover, a device may consists of other devices (saref:consistsOf property). For example:

- a washing machine is a device that has category saref:Appliance, accomplishes the task saref:Washing and performs an actuating function of type saref:StartPauseFunction. Notice that from an energy related perspective, a washing machine also belongs to the category saref:Load. This shows the flexibility of SAREF that trough the saref:DeviceCategory class allows the same device to be classified in different ways without creating inconsistencies;
- a sensor is a device that has category saref:Sensor and performs a saref:SensingFunction;
- a temperature sensor is a device that consists of a sensor, has category saref:Sensor, performs the saref:SensingFunction and is used for the purpose of sensing a property of type saref:Temperature;
- a smoke sensor is a device that consists of a sensor, has category saref:Sensor, performs the saref:SensingFunction and saref:EventFunction , and is used for the purpose of sensing a property of type saref:Smoke and notifying that a certain threshold has been exceeded;
- a switch is a device that has category saref:Actuator and performs an actuating function of type saref:OnOffFunction or saref:OpenCloseFunction;
- a door switch is a device that consists of a switch, has category saref:Actuator, performs the saref:OpenCloseFunction and is used for the purpose of controlling a building object of type saref:Door;
- a dimmer lamp is a device that has category saref:Lighting and saref:Actuator, performs an actuating function of type saref:LevelControlFunction and is used for the purpose of controlling a property of type saref:Light;
- a meter is a device that has category saref:Meter and performs a saref:MeteringFunction;
- an energy meter is a device that consists of a meter, has category saref:Meter, performs the saref:MeteringFunction and is used for the purpose of measuring the saref:Energy property;

More types of devices, sensors, actuators, etc. exist and can be defined to extend SAREF. The devices described above represent some examples that aim at explaining the rationale behind SAREF.

A function is represented in the SAREF ontology with the `saref:Function` and is defined as *"the functionality necessary to accomplish the task for which a device is designed"*. Examples of functions are the `saref:ActuatingFunction`, `saref:SensingFunction`, `saref:MeteringFunction` and `saref:EventFunction`. The `saref:Function` class and its properties are shown in Figure 43.

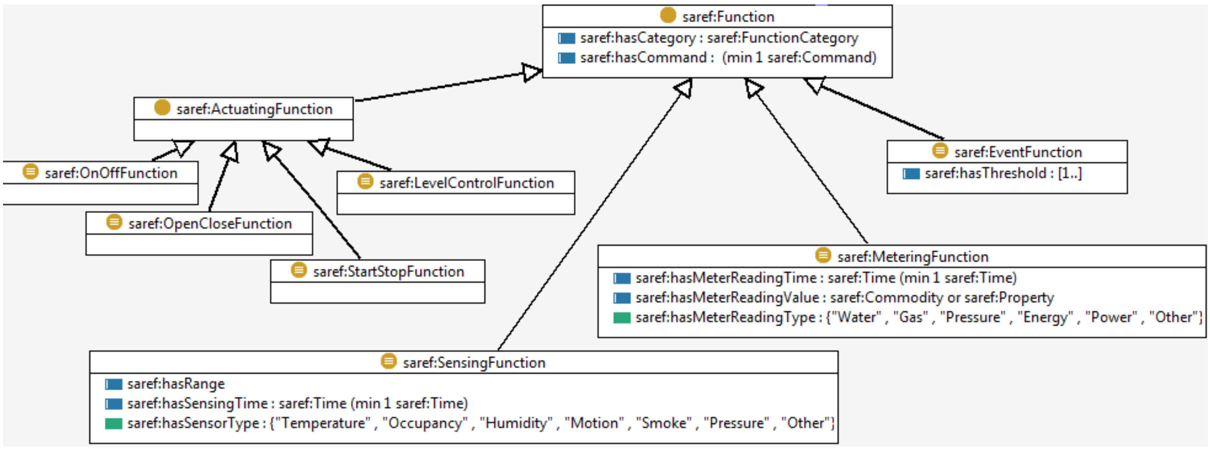


Figure 43: Function class and its properties

A `saref:Function` can belong to a function category (`saref:hasCategory` property). Analogously to the `saref:DeviceCategory` class, we decided to leave the `saref:FunctionCategory` class open in order to grant the asset's owners the flexibility to use their own categories. For example, OSGi DAL could map its `osgidal:FunctionType` class to the SAREF ontology, defining `osgidal:FunctionType` as a subclass of `saref:FunctionCategory`. Figure 43 further shows that a `saref:Function` must have at least one command associated to it (`saref:hasCommand min 1 saref:Command`). Figure 44 shows the list of commands currently available in the SAREF ontology. This list is used here for illustration purposes and can be extended with new commands.

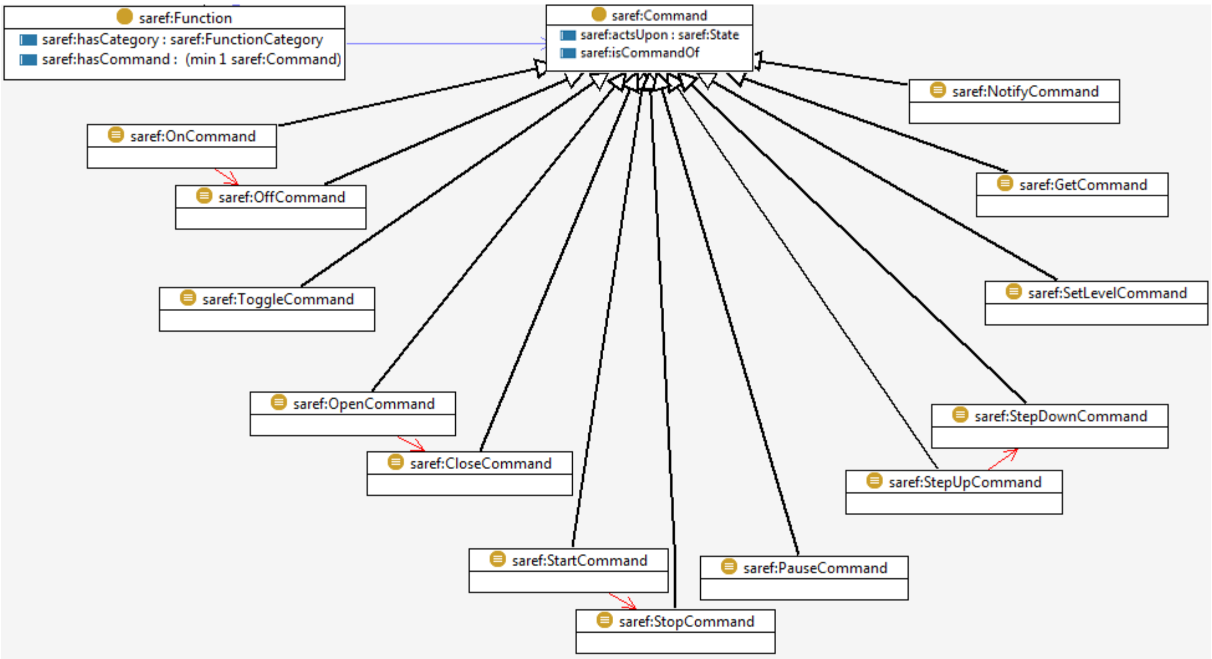


Figure 44: Command class

For example:

- the `saref:ActuatingFunction` allows to *"transmit data to actuators, such as level settings (e.g., temperature) or binary switching (e.g., open/close, on/off)"*
    - the actuating function of type `saref:OnOffFunction` in Figure 43 allows to "*switch on and off an actuator*". This function allows the commands `saref:OnCommand`, `saref:OffCommand` and `saref:ToggleCommand` shown in Figure 44, whereas the `saref:OnCommand` is disjoint from the `saref:OffCommand`.
    - the actuating function of type `saref:LevelControlFunction` in Figure 43 allows to *"do level adjustments of an actuator in a certain range (e.g., 0%-100%), such as dimming a light or set the speed of an electric motor"*. This function allows the commands `saref:SetLevelCommand` (which can be of type `saref:SetAbsoluteLevel` or `saref:SetRelativeLevel`), `saref:StepUpCommand` and `saref:StepDownCommand` shown in Figure 44, whereas the `saref:StepUpCommand` is disjoint from the `StepDownCommand`.
- the `saref:SensingFunction` in Figure 43 allows to "*transmit data from sensors, such as measurement values (e.g., temperature) or sensing data (e.g., occupancy)*". This function allows the command `saref:GetCommand` shown in Figure 44.
- the `saref:EventFunction` in Figure 43 allows to "*notify another device that a certain threshold value has been exceeded*". This function allows the command `saref:NotifyCommand` shown in Figure 44.

Figure 44 further shows that a command can act upon a state (`saref:actsUpon` relation) to represent that the consequence of a command can be a change of state of the device. Notice that a command may act upon a state, but does not necessarily act upon a state. For example, the `saref:OnCommand` acts upon the `saref:OnOffState`, but the `saref:GetCommand` does not act upon any state, since it only gives a directive to retrieve a certain value.

Depending on the function(s) it performs, a device can be found in a corresponding saref:State, as shown in Figure 45. For example, a switch can be found in the saref:OnOffState, which is characterized by the values ON or OFF (saref:hasValue property). Notice that SAREF is not restricted to binary states such as the saref:OnOffState , but allows to define also n-ary states (see, for example, the saref:MultiLevelState class).

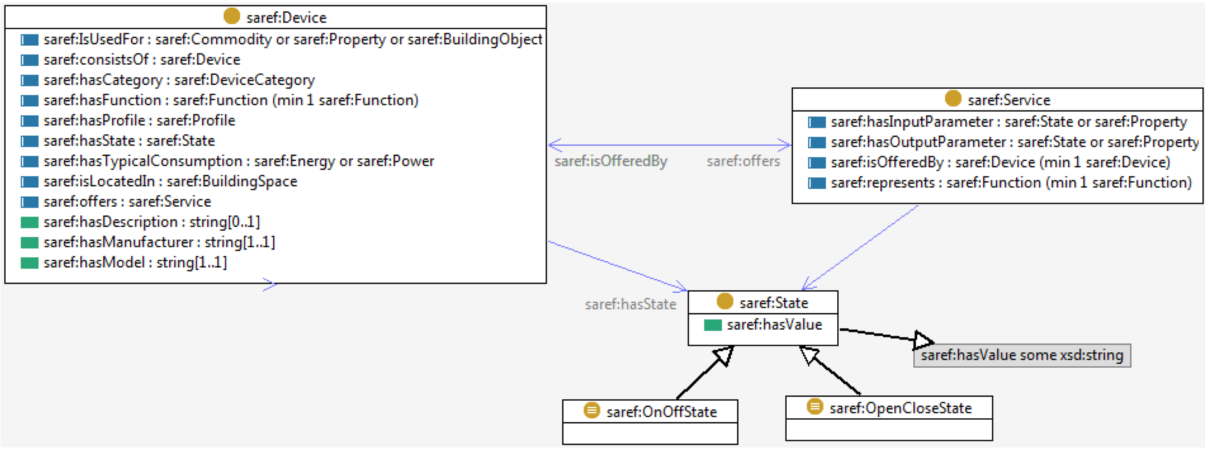


Figure 45: State and Service classes

Figure 45 further shows that a device offers a service (the saref:Service class), which is a representation of a function to a network that makes this function discoverable, registerable and remotely controllable by other devices in the network. A service must represent at least one function (saref:represents min 1 saref:Function) and is offered by at least one device that wants (a certain set of) its function(s) to be discoverable, registerable and remotely controllable by other devices in the network (saref:isOfferedBy min 1 saref:Device). Multiple devices can offer the same service. A service must specify the device that is offering the service, the function(s) to be represented, and the input and output parameters necessary to operate the service (saref:hasInputParameter and saref:hasOutputParameter properties). For example, a light switch can offer the service of remotely switching the lights in a home through mobile phone devices that are connected to the local network. This "remote switching" service represents the saref:OnOffFunction previously described, it must have a saref:State as input parameter, e.g., with value "ON" , and it must have a saref:State has output parameter, namely with value "OFF" in this example since the input state value was "ON".
Moreover, a device in the SAREF ontology can be characterized by a profile that can be used to optimize the energy efficiency in the home or office under consideration. Figure 46 shows the saref:Profile class and its properties.

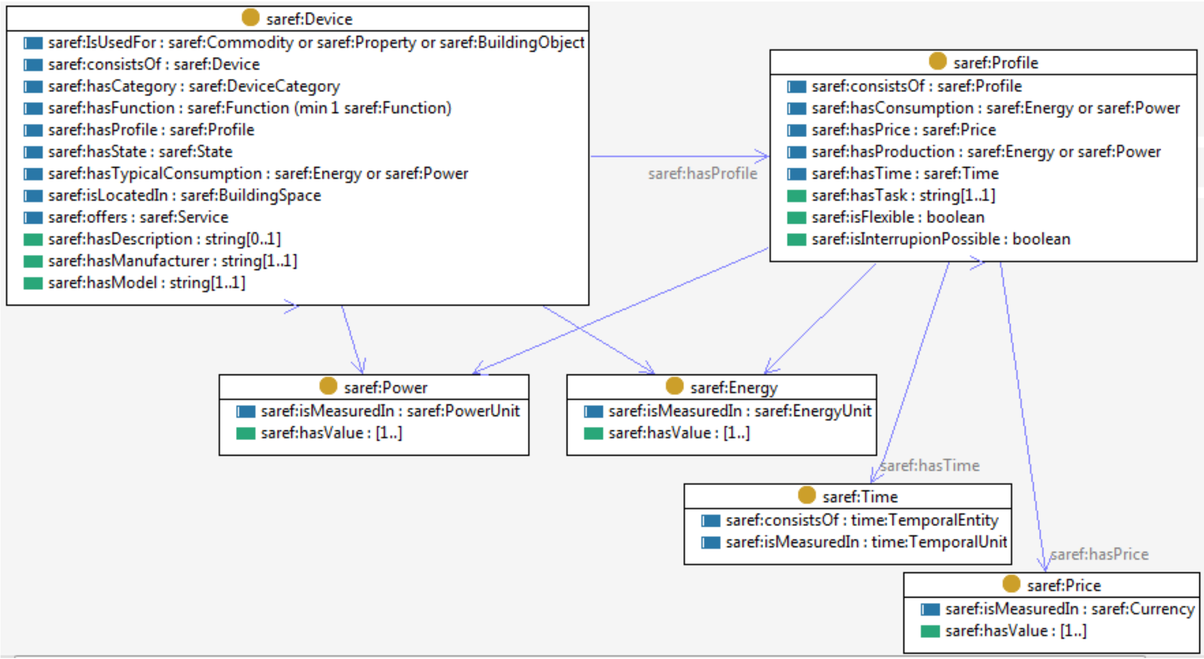


Figure 46: Profile class

The `saref:Profile` class allows to describe the energy (or power) production and consumption of a certain device using the `saref: hasProduction` and `saref:hasConsumption` properties shown in Figure 46 . This production and consumption can be calculated over a time span (the `saref:hasTime` property) and, eventually, associated to some costs (the `saref:hasPrice` property).

The `saref:Power` and `saref:Energy` classes are characterized by a certain value (`saref:hasValue` property) that is measured in a certain unit of measure represented by the `saref:UnitOfMeasure` class, namely `Kilowatt` and `Kilowatt_Hour`, respectively. Analogously, the `saref:Price` class is characterized by a certain value (`saref:hasValue` property) and is measured using a certain `saref:Currency`, which is a subclass of the `saref:UnitOfMeasure` class.

The `saref:Time` class allows to specify the "time" concept in terms of instants or intervals according to the existing W3C Time ontology[75] that we import in our SAREF ontology to avoid defining this concept from scratch. The concepts of the W3C Time ontology that are useful for the purpose of the SAREF ontology are shown in Figure 47. We refer to W3C Time ontology with the `time:` prefix in order to distinguish from the classes and properties of the SAREF ontology, which are referred to using the `saref:` prefix.

[75] http://www.w3.org/TR/owl-time/

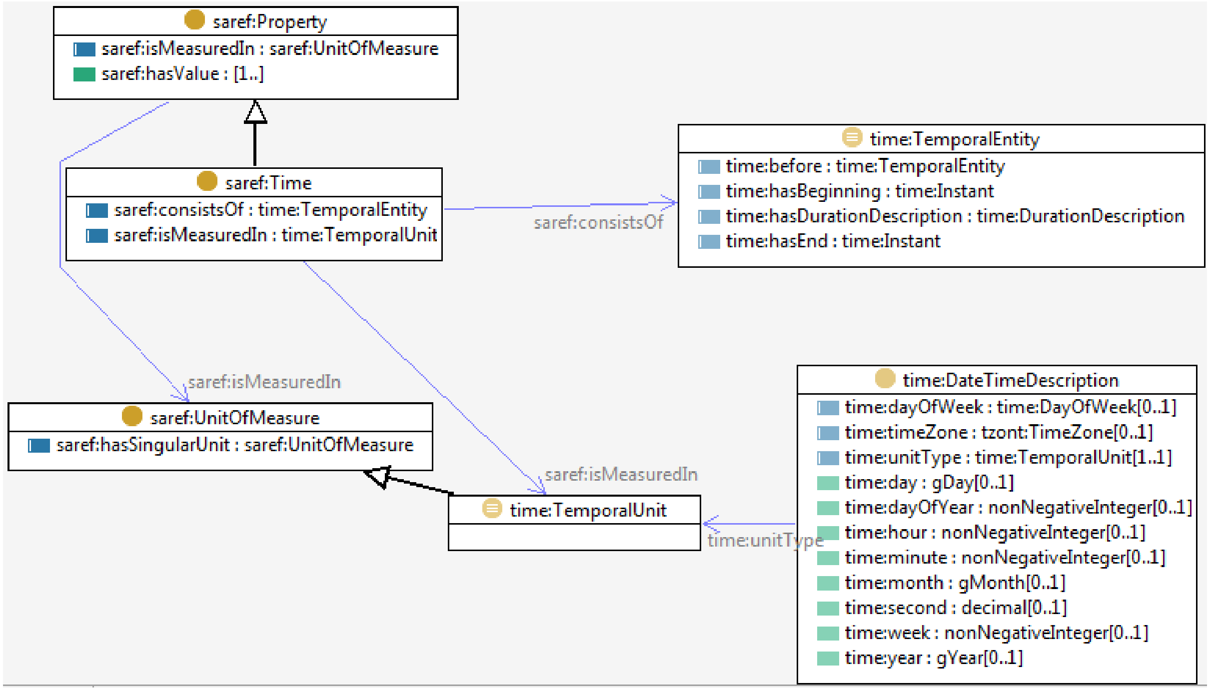


Figure 47: Time class

7.4 Observations about SAREF

- SAREF currently does not contain explicit references to upper ontologies such as DUL or SUMO. We acknowledge that the use of upper ontologies is a best practice in ontology engineering, but we argue that the smart appliances industry - main user of SAREF - is very pragmatic and is not acquainted with high-level upper ontologies. Introducing DUL would have unnecessarily complicated the understanding and, consequently, the adoption of SAREF by the smart appliances industry. SAREF has been built on a solid ontological foundation and can be related to DUL, but this will not be done at this early stage of SAREF in order not to confuse the smart appliances industry's users. Furthermore, SAREF currently has mappings to the W3C SSN ontology, which is in turn related to DUL. Therefore, SAREF currently includes an indirect reference to DUL through the W3C SSN ontology. In section 2.4 it is explained that we regard an ontology as an artifact that includes precise definitions of the ontology concepts in natural language e.g., 'an appliance is a tangible object designed to accomplish a particular task in households, such as cooking or cleaning. In order to accomplish this task, the appliance performs one or more functions'. In the "saref.ttl file" that contains the OWL version of SAREF (available at http://ontology.tno.nl/saref.ttl) these definitions can be found as `rdfs:comment` properties attached to the most important SAREF classes. Due to the large amount of concepts in SAREF, the definitions of self-explanatory concepts, e.g., `saref:OnCommand` class, are omitted. Table 3 shows a summary of the main SAREF definitions.

Table 3: Summary of main SAREF definitions

CONCEPT	DEFINITION
Building Object	A Building Object is an object in the building that can be controlled by devices, such as a door or a window that can be automatically opened or closed by an actuator

Building Space	According to FEIMSER, a Building Space in SAREF defines the physical spaces of the building. A building space contains devices or building objects.
Command	A Command is a directive that a device must support to perform a certain function. A command may act upon a state, but does not necessarily act upon a state. For example, the ON command acts upon the ON/OFF state, but the GET command does not act upon any state, since it gives a directive to retrieve a certain value with no consequences on states.
Commodity	A Commodity is a marketable item for which there is demand, but which is supplied without qualitative differentiation across a market. SAREF refers to energy commodities such as electricity, gas, coal and oil.
Device	A Device in the context of the Smart Appliances study is a tangible object designed to accomplish a particular task in households, common public buildings or offices. In order to accomplish this task, the device performs one or more functions. For example, a washing machine is designed to wash (task) and to accomplish this task it performs the start and stop function.
Device Category	A Device Category provides a way to classify devices according to a certain point of view, for example, the point of view of the user of the device vs. the device's manufacturer, or the domain in which the device is used (e.g., smart appliances vs. building domain vs. smart grid domain), etc.
Function	A Function represents the particular use for which a Device is designed. A device can be designed to perform more than one function.
Function Category	A Function Category provides a way to classify functions according to a certain point of view, for example, considering the specific application area for which a function can be used (e.g., light, temperature, motion, heat, power, etc.), or the capability that a function can support (e.g., receive, reply, notify, etc.), and so forth.
Profile	A Profile caracterizes a device for the purpose to optimize the energy efficiency in the home or office in which the device is located. The saref:Profile class allows to describe the energy (or power) production and consumption of a certain device using the saref: hasProduction and saref:hasConsumption properties. This production and consumption can be calculated over a time span (the saref:hasTime property) and, eventually, associated to some costs (the saref:hasPrice property).
Property	A Property is anything that can be sensed, measured or controlled in households, common public buildings or offices.
Service	A Service is a representation of a function to a network that makes the function discoverable, registerable, remotely controllable by other devices in the network. A service can represent one or more functions. A Service is offered by a device that wants (a certain set of) its function(s) to be discoverable, registerable, remotely controllable by other devices in the network. A Service must specify the device that is offering the service, the function(s) to be represented, and the (input and output) parameters

	necessary to operate the service.
State	A State represents the state in which a device can be found, e.g, ON/OFF/STANDBY, or ONLINE/OFFLINE, etc.
Task	A Task represents the goal for which a device is designed (from a user perspective). For example, a washing machine is designed for the task of cleaning.
Unit of Measure	The Unit of Measure is a standard for measurement of a quantity, such as a Property. For example, Power is a property and Watt is a unit of power that represents a definite predetermined power: when we say 10 Watt, we actually mean 10 times the definite predetermined power called "watt". Our definition is based on the definition of unit of measure in the Ontology of units of Measure (OM). We propose here a list of some units of measure that are relevant for the purpose of the Smart Appliances ontology, but this list can be extended.

- Usually, it would be common practice for an ontology developer to create hierarchies of device categories as subclasses (types) of the `saref:Device` class, as shown in Figure 48 (a). In contrast, we decided to adopt in SAREF a flat classification of devices under the `saref:Device` class - in other words, no hierarchies of device types - and provide device categories using the `saref:hasCategory` relation, as shown in Figure 48 (b). This was a specific design choice to simplify SAREF as much as possible for its users and keep it as much as possible independent from subjective choices. For example, most of the users would classify `TemperatureSensor` and `SmokeSensor` as subclasses of `Sensor`, as depicted in Figure 48 (a). But this is an easy example. If we need to add a new device of type `MobilePhone`, where the users of SAREF would add it? One could say it is a subclass of `MultimediaDevice`, but another user could argue that it is a subclass of `Sensor`. To make it even more difficult, what happens if we need to define a new type of combined sensor such as `TemperatureHumiditySensor`? Should it be a subclass of `TemperatureSensor` or `HumiditySensor` or a subclass of both? In order to avoid this type of issues, which require choices that are too specific/subjective and would harm the general applicability of SAREF, we decided to have a flat list of devices under the `saref:Device` class with no further hierarchy, even if it is in principle possible to create hierarchies. It is then possible to assign devices from this flat list to device categories using the `saref:hasCategory` relation to the `saref:DeviceCategory` class. Users can eventually define their own categories under the `saref:DeviceCategory` class and other users can simply ignore categorizations that are not relevant for them.

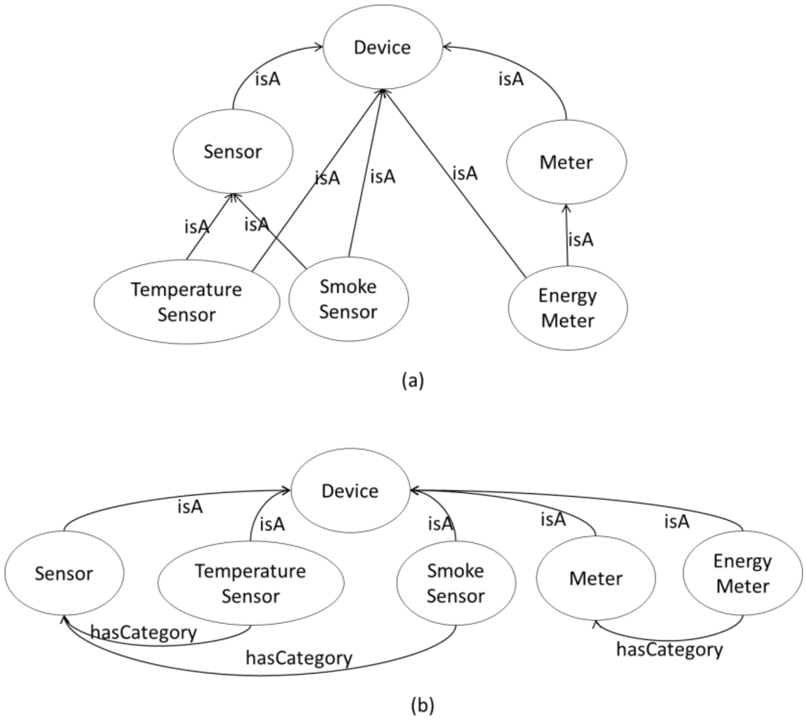


Figure 48: Device categories as subclasses of Device (a) vs. using the has Category relation (b)

- The scope of SAREF is currently limited to an indoor managed domain, such as a building managed by a building manager or an apartment managed by a user. This scope also includes the outdoor premises that belong to the considered indoor managed domain, in other words, a pergola that is part of the building is also within the scope, as well as a sensor located under that pergola. Please note that the smart city domain is currently not considered, i.e., if the same sensor under the pergola is also in a street, then the sensor in the street is out of the scope of SAREF. However, since in principle the sensor in the street can be also defined using the SAREF definition of device, nothing prevents us in the future to extend the scope of SAREF also to outdoor domains (e.g., smart cities) managed by managers different than building managers or apartment users considered here, such as for example an administrative manager of the city government.

8. Application of SAREF in relation to the ETSI M2M Architecture

8.1 ETSI Smart M2M Functional Architecture

ETSI M2M recently released its Functional Architecture [5], which describes the overall end-to-end M2M functional architecture, including the identification of the functional entities and the related reference points. The high-level architecture is shown in Figure 49. It includes a Device and Gateway Domain and a Network domain. The Device and Gateway Domain contains M2M Devices, M2M Gateways, and M2M Area Networks. M2M Devices run M2M Application(s) using M2M Service Capabilities. M2M Devices connect to the Network Domain either directly via the Access Network (xDSL, HFC, satellite, FttH, 3G, etc.) or indirectly via an M2M Area Networks and one or more M2M Gateways. Examples of M2M Area Networks include technologies such as Zigbee, Bluetooth, Wireless M-BUS and KNX.

The M2M Gateway is a gateway that runs M2M Application(s) using M2M Service Capabilities. The Gateway acts as a proxy between M2M Devices and the Network Domain. The M2M Gateway may provide services to other devices (e.g. legacy) connected to it that are hidden from the Network Domain. As an example an M2M Gateway may run an application that collects and treats various information (e.g. from sensors and contextual parameters).

The Network Domain is composed of Access Networks, Core Networks, and platforms running M2M Service Capabilities, M2M Applications, Network Management Functions, and M2M Management Functions. Network Management Functions consist of all the functions required to manage the Access and Core networks: these include Provisioning, Supervision, Fault Management, etc. M2M Management Functions consist of all the functions required to manage M2M Service Capabilities in the Network Domain.

The M2M Service Capabilities layer is arguably the most important part of the ETSI M2M Functional Architecture. The Service Capability Layer enables the transport of M2M data between devices or gateways and network applications. It provides an abstraction layer hiding the heterogeneity of M2M access networks and provides means for secure data transport. The M2M Service Capabilities:

- Provide M2M functions that are to be shared by different Applications
- Expose functions through a set of open interfaces
- Use Core Network functionalities
- Simplify and optimize application development and deployment through hiding of network specificities

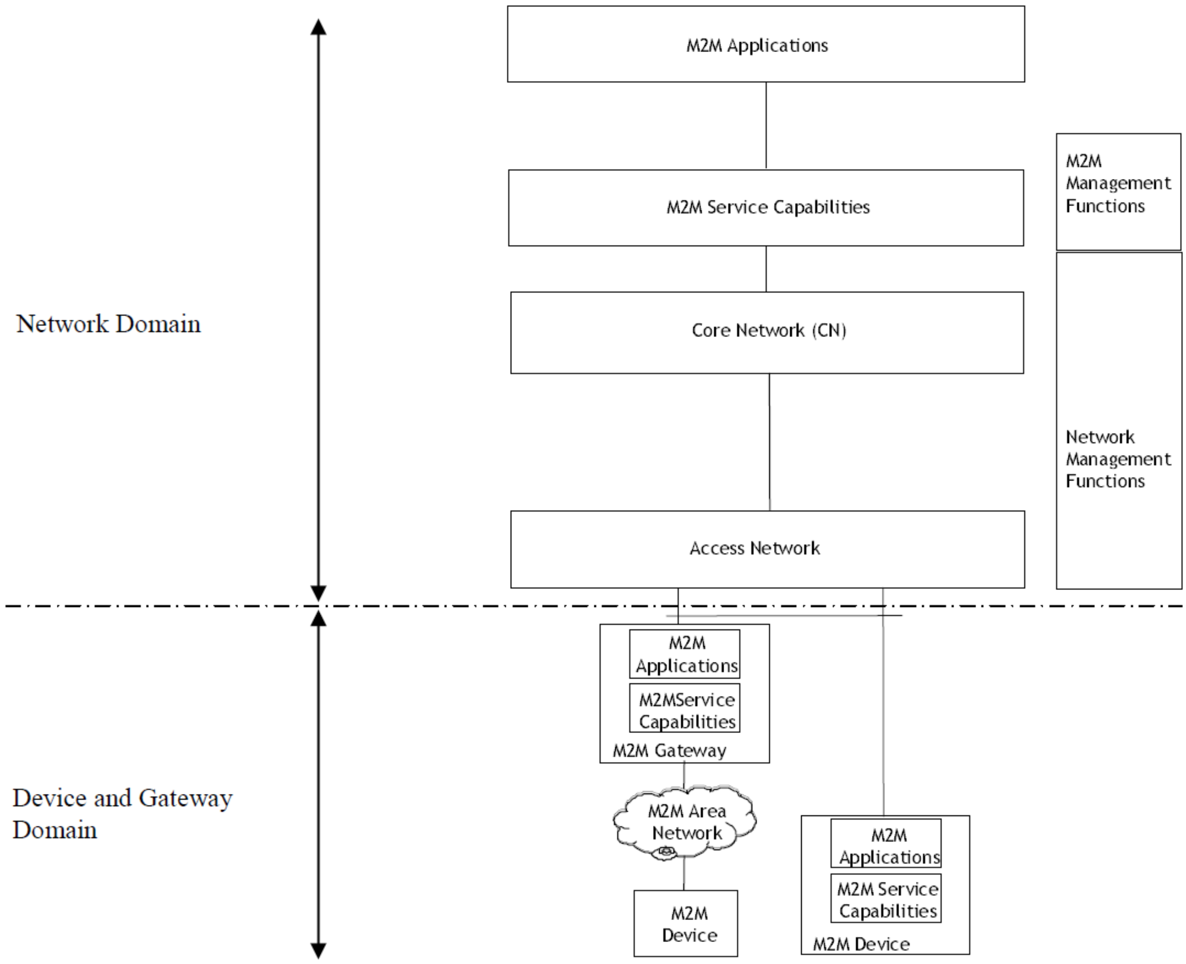


Figure 49: ETSI M2M High Level Architecture [5]

The M2M Applications run the service logic and use M2M Service Capabilities accessible via an open interface. In [5] this interface is called “dIa” for device applications and “mIa” for network applications. In Figure 50 they are denoted in the ETSI M2M High Level Architecture. It is dIa and mIa that SAREF applies to.

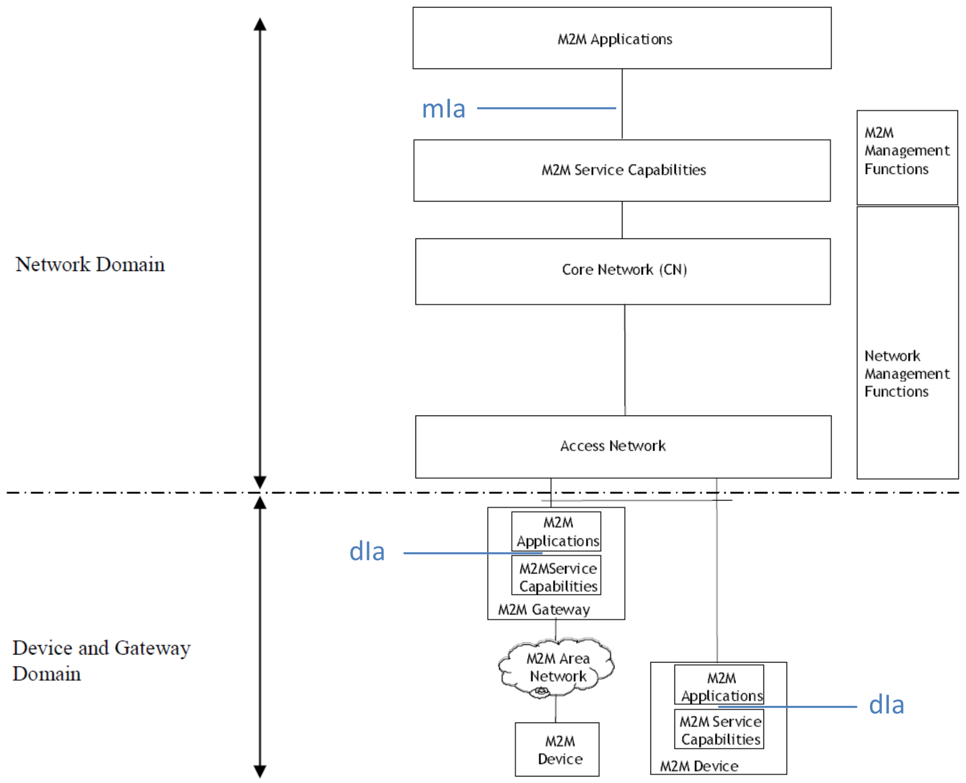


Figure 50: dIa and mIa interfaces in the ETSI M2M High Level Architecture

8.2 Current semantic support for M2M data

At the moment, the Service Capability Layer is handling only data containers without any knowledge of the data contained. As described in [6], this approach has a number of limitations, including:

- The common-place vertically integrated, but isolated M2M applications are now replaced by M2M applications which are re-using a common data transport, but which are still vertically integrated and isolated from each other;
- There is no support in the SCL to enable an open market of data, e.g. in which data owner publish (sell) their data and independent data users provide applications that make use of the data.

After studying various use cases and different approaches to semantics, reference [6] subsequently suggests the following potential requirements regarding semantics in a next release of the M2M Functional Architecture:

1. M2M system support for a common (e.g. per vertical domain) semantic data model (e.g. represented by Ontology) available to M2M application.
2. M2M system provision of discovery capabilities enabling the discovery of M2M resources based on their semantic information, e.g. semantic categories and relationship among them (e.g. all heaters and windows in a room; the room in which a window is located…).

3. M2M system provision of representation and discovery functionality of real-world entities (rooms, windows) that are not necessarily physical devices.
4. M2M system ability support the mapping of control commands issued towards an abstract device to the concrete commands of a specific device.
5. M2M system support of a semantic data model that is at least common to the vertical industry in which a Thing is used to describe Things registered in the M2M System.
6. M2M entities ability to expose their semantic description to the M2M System.
7. M2M System ability to re-use semantic information provided by external entities to create a virtual representation. System ability to describe the semantic relationship between Things.

The term M2M System indicates in a general way M2M entities like: device, gateway and network infrastructure, equipped with M2M Service Capabilities. A Thing is defined as an element of the environment that is individually identifiable in the M2M system.

SAREF can thus subsequently be applied by the industry to produce ETSI M2M compliant devices, or interoperability boxes to make existing, non-ETSI-M2M devices interwork with an ETSI M2M system. Ideally, the achieved interoperability would comply with the highest levels as defined by e.g. CENELEC [7], but it all depends on the richness of the protocol interfaces, and how well the already implemented data models translate into the unified ones.

In reference [6] and [8] ETSI Smart M2M elaborated some preliminary examples on how this interoperability could be achieved given a preliminary ontology.

8.3 Mapping SAREF into the ETSI M2M resource structure

SAREF is somewhat different and also more extensive than the preliminary semantic model as presented in [6]. Here, we will not discuss the differences in detail but use the methodology as provided in [6] and [8] to provide a mapping between SAREF and the ETSI M2M architecture.

A saref:Device obviously maps to an ETSI M2M Device. ETSI separately defines an M2M Gateway. In SAREF this should be (we have not defined it explicitly yet) a saref:FunctionRelated saref:DeviceCategory.

In ETSI a M2M Device is described in terms of its so-called resources it provides. They can be mapped on SAREF as shown in Table 4.

Table 4: Mapping of ETSI M2M Device Resources to SAREF

etsiSclMo	This is the management object of the service capability layer. SAREF has not yet considered remote management of devices.
etsiDevicInfo	Includes the saref:hasModel, saref:hasManufacturer properties and saref:FunctionRelated saref:DeviceCategory. There is no space for a free-format saref:hasDescription field.
etsiDeviceCapability	Maps to saref:Function except for the saref:EventFunction.
etsiBattery	No direct match. However, one saref:DeviceCategory is saref:EnergyRelated saref:Storage which in principle describes a battery function.
etsiMemory	No match
etsiTrapEvent	Maps to saref:EventFunction
etsiPerformanceLog	No match
etsiFirmware	No match
etsiSoftware	No match
etsiReboot	No direct match. However, there may be a relation with the saref:OnOffFunction.
etsiAreaNwkInfo	Possibly maps to saref:Network, but we have not elaborated this DeviceCategory any further yet.
etsiAreaNwkDeviceInfo	Possibly maps to saref:Network, but we have not elaborated this DeviceCategory any further yet.

There is no match between ETSI M2M and the SAREF saref:EnergyRelated and saref:BuildingRelated Device Categories, nor is there a relation yet with the saref:BuildingSpace location, the saref:Profile, or the saref:Time class.

There is possibly a relation between the concepts of M2M Applications and Services in SAREF, but this needs further study.

9. Conclusions

We have identified 47 semantic assets that needed to be included in our study given the scope as set out by the European Commission. That is 18 more than initially identified in the Invitation to Tender for this study. Of these 47 assets we were able to short-list 23 which provide a good basis for further reference ontology development. The short list was composed solely based on how well each asset was covering the scope of the project and if the asset provided concrete semantic specifications, preferably in the form of XML or OWL files.

The considered assets were heterogeneous when considering their semantic coverage. However, we could identify three main trends with a focus on:

- Devices, sensors and their specification in terms of services, functions and states (e.g., Echonet, eDIANA, EnOcean, OMALightweightM2M, OSGi DAL, SEP2, UPnP);
- Energy consumption information and profiles to optimize energy efficiency (e.g., FAN, FIEMSER, Mirabel, SESAME);
- Buildings related semantic models (e.g., DEHEMS, SEEMPubs, SEIPF).

Most assets mainly focused on one of these three trends and sometimes they did not show much (linguistic) overlap with assets covering one of the other trends. However, assets covering different trends could be connected starting from the most recurrent terms shown in the visual representation in Chapter 4, namely Device, Sensor, Service, State and Event.

Some of the considered assets did not provide sufficient information and/or documentation to define their semantic coverage (e.g., AIM*, ENERsip, CoAP, Agora).

The assessment of the items on the long list led to a short list of assets that we identified as the most relevant for building the reference ontology. These assets are in the core of the trends discovered and furthermore provide complete information in terms of data models and product specifications that we could use to build the reference ontology. Several semantic assets in the short list provided detailed documentation about the OWL ontologies they have built, but they did not provide a URL to the corresponding OWL files. These assets are eDiana, FIPA, Hydra, and SEIPF. The OWL files are essential for us to make sure that we based the reference ontology on the actual models that were defined in the projects or organizations, and not on our own interpretations of the documentation available for these ontologies.

We were able to translate the semantic assets in the short list to corresponding OWL ontologies, and created an initial mapping among these ontologies. The purpose of the mappings was to relate the 23 assets in the short list using their most recurring concepts, which then became the core concepts of the reference ontology in task 3. In order to perform the translation and mapping tasks, we have followed a systematic approach that allowed us to deal with the quantity of ontologies to be created and their complexity.

Out of the 23 assets:

- 6 assets were already expressed in OWL, namely eDIANA, Hydra, PowerOnt, ZigbeeHA, Adapt4EE and SSN. We contacted the authors of these ontologies in order to obtain the original OWL files.
- 16 assets have been translated into OWL from scratch. For each of these assets we have created an ontology expressed in OWL-DL and serialized in Turtle, therefore, they have a file extension 'ttl'.

All the ontologies are published online at https://sites.google.com/site/smartappliancesproject. The review of the ontologies by the "owners" has been actively solicited. We did not elaborate every ontology in all possible detail. However, we think that this is not necessary as, to achieve the final goal of the project, we only needed to find the commonalities between the various ontologies. Moreover, having learned from our approach, every stakeholder can now do the work himself, improving and/or extending the ontology to his liking given the open character of our results.

We acknowledge that our interpretation of the assets may not be always as intended by the "owners", therefore we will improve and update the (online version of the) ontologies according to the stakeholders' feedback until the end of the project in March 2015. Changes or extensions on these ontologies, together with new mappings to the reference ontology that may emerge in the future, should be realized by the interested stakeholders once the project will have ended. Concerning the reference ontology, its development is incremental and we will gradually improve it taking into account the reviews of the project's expert group and the stakeholders until March 2015. We recommend that further maintenance of the reference ontology will be facilitated by an industrial organization or standard development organization such as ETSI Smart M2M, and that the supporting research and development activities are stimulated, for instance via the Horizon 2020 call ICT 30.

We created a first version of the Smart Appliances REFerence (SAREF) ontology. SAREF explicitly specifies recurring core concepts in the smart appliances domain as given by the short-listed assets, the main relationships between these concepts, and axioms to constrain the usage of these concepts and relationships. SAREF is based on the fundamental principles of r*euse and alignment* of concepts and relationships that are defined in existing assets, *modularity* to allow separation and recombination of different parts of the ontology depending on specific needs, e*xtensibility* to allow further growth of the ontology, and *maintainability* to facilitate the process of identifying and correcting defects, accommodate new requirements, and cope with changes in (parts of) the SAREF ontology.

We subsequently mapped SAREF on the ETSI M2M Architecture, and found that there is a good correlation between the ETSI M2M Architecture and SAREF's function-related device categories. The mapping with energy-related and building-related device categories is still minimal. For further implementation of SAREF into ETSI M2M, the following actions need to be taken:

- SAREF needs to be extended with ETSI M2M specific functionality, such as M2M Gateway, and Remote Management functionality.
- ETSI resource description should be extended with (more) energy-related functionality and building-related functionality.

- The ETSI architecture should introduce a clear separation between functions (device capabilities) and services (the interface a device offers to a network).

The Smart Appliances Reference (SAREF) ontology is published online at http://ontology.tno.nl/saref.

References

[1] "Manual for statistics on energy consumption in households", ISSN 2315 - 0 815, Eurostat (2013).

[2] Frank den Hartog, "Consumer Networking Standardization: trends and research opportunities", IEEE Webcast Tutorial, http://www.comsoc.org/webcasts/view/consumer-networking-standardization-trends-and-research-opportunities, IEEE (2011).

[3] Invitation to Tender for a Study on "Available Semantics Assets for the Interoperability of Smart Appliances. Mapping into a Common Ontology as a M2M Application Layer Semantics" – SMART 2013/0077, http://ec.europa.eu/digital-agenda/en/news/invitation-tender-study-available-semantics-assets-interoperability-smart-appliances-mapping, EC (2013).

[4] David S. Watson, Mary Ann Piette, Osman Sezgen, and Naoya Motegi, "Machine to Machine (M2M) Technology in Demand Responsive Commercial Buildings", in Proceedings from the ACEEE 2004 Summer Study on Energy Efficiency in Buildings: Breaking out of the Box (2004).

[5] ETSI TS 102 690 v2.1.1, "Machine-to-Machine communications (M2M); Functional architecture", ETSI (2013).

[6] ETSI TR 101 584 V2.1.1, "Machine-to-Machine Communications (M2M); Study on Semantic support for M2M Data", ETSI (2013).

[7] CWA50560:2010 "IFRS (Interoperability Framework Requirement Specification)", CENELEC (2010).

[8] ETSI TR 102 966 V1.1.1, "Machine-to-Machine communications (M2M); Interworking between the M2M Architecture and M2M Area Network technologies", ETSI (2014).

[9] Chuck Eastman, Paul Teicholz, Rafael Sacks, Kathleen Liston, "BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors, 2nd Edition", ISBN: 978-0-470-54137-1, Wiley, 2011.

[10] ETSI EN 300 175 "Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI)"; parts 1-8, ETSI (2007).

[11] ETSI TS 102 939-1 "Digital Enhanced Cordless Telecommunications (DECT); Ultra Low Energy (ULE); Machine to Machine Communications; Part 1: Home Automation Network (phase 1)", ETSI (2013).

[12] ISO/IEC 14543-3-10:2012, "Information technology -- Home Electronic Systems (HES) -- Part 3-10: Wireless Short-Packet (WSP) protocol optimized for energy harvesting -- Architecture and lower layer protocols", ISO/IEC (2012).

[13] Vincent Gay, *et al*, "D4 – FIEMSER System Architecture", 15 February 2011, http://www.fiemser.eu/?page_id=40 .

[14] Marc Bourdeau, *et al*, “D9 – Interface modules”, 30 September 2011, http://www.fiemser.eu/?page_id=40 .

[15] ISO/IEC 14543-3-x, “Information technology -- Home Electronic Systems (HES) Architecture”, ISO/IEC (2006-2007).

[16] EN 50090-x-y, “Home and Building Electronic Systems (HBES)”, CENELEC (1994-2011).

[17] SML - Smart Message Language - Version 1.02, VDE (2008), http://www.vde.com/de/fnn/extras/tlz/documents/sml_080119_102.pdf .

[18] EN 13757-x, “Communication System for Meters and Remote Reading of Meters”, CENELEC (2003-2013).

[19] IEC 61968, “Common Information Model (CIM) / Distribution Management”, IEC (2003-2010).

[20] IEEEE 2030.5-2013 “IEEE Adoption of Smart Energy Profile 2.0 Application Protocol Standard”, IEEE (2013).

[21] ISO/IEC 29341-x-y “Information technology -- UPnP Device Architecture”, ISO/IEC (2008-2011).

[22] ISO 16484 “Building automation and control systems (BACS)”, ISO (2004-2007).

[23] M/490 EN “Smart Grid Mandate, Standardization Mandate to European Standardisation Organisations (ESOs) to support European Smart Grid deployment”, http://ec.europa.eu/energy/gas_electricity/smartgrids/doc/2011_03_01_mandate_m490_en.pdf .

[24] ISO/IEC 14908-x “Information technology -- Control network protocol”, ISO/IEC (2012), EN 14908-x “Open Data Communication in Building Automation, Controls and Building Management - Control Network Protocol”, CEN (2014).

[25] IEC 62056-5-3 ed1.0 “Electricity metering data exchange - The DLMS/COSEM suite - Part 5-3: DLMS/COSEM application layer”, IEC (2013).

[26] IEC 61850-x ed1.0 “Communication networks and systems in substations”, IEC (2003-2013).

[27] RFC 4944 “Transmission of IPv6 Packets over IEEE 802.15.4 Networks”, IETF (2007), http://datatracker.ietf.org/doc/rfc4944/; RFC 6282, “Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks”, IETF (2011), http://datatracker.ietf.org/doc/rfc6282/ ; RFC 6775 “Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)”, IETF (2012), http://datatracker.ietf.org/doc/rfc6775/.

[28] IEC 61334-4-41 “Distribution automation using distribution line carrier systems - Part 4: Data communication protocols - Section 41: Application protocol - Distribution line message specification”, IEC (1996).

[29] IEC 62056-6-1 ed1.0 “Electricity metering data exchange - The DLMS/COSEM suite - Part 6-1: Object Identification System (OBIS)”, IEC (2013).

[30] IEC 62056-6-2 ed1.0 “Electricity metering data exchange - The DLMS/COSEM suite - Part 6-2: COSEM interface classes”, IEC (2013).

[31] Kuo-Ming Chao, N. Shah, R. Farmer, A. Matei, Ding-Yuan Chen, H. Schuster-James, R. Tedd, “A Profile Based Energy Management System for Domestic Electrical Appliances”, in Proc. of the IEEE 7th International Conference on e-Business Engineering (ICEBE), 2010.

[32] EN 15500 “Control for heating, ventilating and air-conditioning applications - Electronic individual zone control equipment”, CEN (2008).

[33] EN 15232 “Energy performance of buildings - Impact of Building Automation, Controls and Building Management”, CEN (2012).

[34] ISO 16739:2013 “Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries”, ISO (2013).

[35]Daniele, L.M. and Ferreira Pires, L. (2013) An ontological approach to logistics. In: Enterprise Interoperability, Research and Applications in the Service-oriented Ecosystem, IWEI'13 Proceedings, 26 Mar 2013, pp. 199-213. ISTE Ltd, John Wiley & Sons, Inc.

[36]Guarino, N., Oberle, D., Staan, S., “What is an Ontology”. In: Handbook on Ontologies, pp. 1-17. Springer Verlag (2009)

[37]Gruber, T. R., “Ontology”. In: Encyclopedia of Database Systems. Springer Verlag (2009).

[38]Guizzardi, G., “Ontological Foundations for Structural Conceptual Models”. PhD Thesis, University of Twente, The Netherlands (2005).

[39]Uschold, M., Gruninger, M., “Ontologies: Principles, Methods and Applications”. In: the Knowledge Engineering Review, vol. 11(2), pp. 93-136. Cambridge University Press (1996).

[40]Daniele L., den Hartog, F., Roes, J., “Study on semantic assets for smart appliances interoperability, D-S2 Second interim report”, European Commission (2014).

[41]Bohring, H., Auer, S., “Mapping XML to OWL Ontologies”. In: Proceedings of the 13th Leipziger Informatik-Tage (LIT 2005), Lecture Notes in Informatics (2005).

Acknowledgements

This study is commissioned by the European Commission. The Project Officers Rogelio Segovia (until his retirement in September 2014) and Svetoslav Mihaylov (from October 2014) from the EC's DGCNECT department has been indispensable in creating the support of and attention from the smart appliances ecosystem needed to perform this study successfully and to embed it in the industry. The authors would also like to thank the members of the project's Expert Group for the interest shown in our work, their helpful guidance and review comments: Giancarlo Guizzardi, Jerome Euzenat, Frank van Harmelen, Patricia Martigne, Susan Schwarze, Marten van Sinderen, Andreas Sayegh, Lindsay Frost, Markus Eisenhauer, Enrico Scarrone, Thomas Kessler and Joachim Koss. We would further like to acknowledge the ETSI Smart M2M TC and other representatives of ETSI for their enthusiastic support of this study.

European Commission

Study on semantic assets for smart appliances interoperability
Luxembourg, Publications Office of the European Union

2015 – 164

Digital Agenda for Europe

European Commission

Digital Agenda for Europe