ETSI TR 101 584 V2.1.1 (2013-12)

**Machine-to-Machine Communications (M2M);
Study on Semantic support for M2M Data**

Reference

DTR/SmartM2M-00017ed211

Keywords

interworking, M2M, semantic

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://ipr.etsi.org).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Report (TR) has been produced by ETSI Technical Committee Machine-to-Machine communications (M2M).

The present document may be referenced by other TRs and Technical Standards (TS) developed by ETSI TC M2M. The present document is a TR and therefore, the content is informative, but when this TR is referenced by a TS, the referenced clauses may become normative with respect to the content of the referencing TS.

1 Scope

The present document is motivated by the fact that within the ETSI M2M System semantic information needs to be available on M2M data that is transferred within the M2M system. Through such semantic information M2M data can be discovered by applications that do not have prior knowledge on them. The capability of the ETSI M2M System to enable applications to discover, interpret and use M2M data from different sources is considered essential for creating high-level M2M services and to develop open markets for M2M data.

- In this study pre-normative work is conducted in order to facilitate normative specification work in ETSI M2M Rel.-2 or later.
- The study analyses benefit, feasibility and potential requirements for the support of semantic information on application related M2M Resources in the M2M system.
  The ETSI M2M system would, however, only provide a means to create and handle such semantic information in the ETSI M2M system; ETSI M2M continues to stay independent of 'vertical' markets who in general would define the semantics of M2M data related to their field of expertise.
- The study creates use cases that illustrate provisioning and usage of such semantic information and that demonstrate the benefit for the M2M ecosystem.
- It investigates on the kind and amount of semantic information that would become available in the M2M system, keeping in mind a trade-off between complexity and usability.
- It investigates discovery mechanisms for semantic information in the ETSI M2M System. This should take into account how existing solutions from other standards or research could be used within the ETSI M2M architecture.
- It considers on issues of ownership/responsibility for application related M2M Resources in the case that the M2M system can provide semantic information on them. This needs to take into account the need for support of different levels of data privacy and confidentiality.

This study relates to WI 0014 (TR 102 966 [i.11] - Interworking between the M2M Architecture and M2M Area Network technologies), as a further step in the abstraction of LAN technologies and devices. Existing relevant standards are taken into account and the study aspires to benefit from inputs of related research projects.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are necessary for the application of the present document.

Not applicable.

2.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI TR 102 725: "Machine-to-Machine communications (M2M); Definitions".

[i.2] [Guarino, 98] Formal Ontology in Information Systems.

NOTE: Available at: http://www.loa.istc.cnr.it/Papers/FOIS98.pdf.

[i.3] W3C: XML Technology.

NOTE: Available at: http://www.w3.org/standards/xml/.

[i.4] W3C: Semantic Web.

NOTE: Available at: http://www.w3.org/standards/semanticweb/.

[i.5] W3C: OWL Web Ontology Language.

NOTE: Available at: http://www.w3.org/standards/techs/owl.

[i.6] W3C: SPARQL Current Status.

NOTE: Available at: http://www.w3.org/standards/techs/sparql#w3c_all.

[i.7] W3C: Resource Description Framework (RDF).

NOTE: Available at: http://www.w3.org/RDF/.

[i.8] W3C: W3C Semantic Sensor Network Incubator Group.

NOTE: Available at: http://www.w3.org/2005/Incubator/ssn.

[i.9] IETF RFC 1738: "Uniform Resource Locators (URL)".

[i.10] ETSI TS 102 690: "Machine-to-Machine communications (M2M); Functional architecture".

[i.11] ETSI TR 102 966: "Machine-to-Machine Communications (M2M); Interworking between the M2M Architecture and M2M Area Network technologies".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TR 102 725 [i.1] and the following apply.

NOTE: A term defined in the present document takes precedence over the definition of the same term, if any, in TR 102 725 [i.1].

Abstract Application Information Model: information model of common functionalities abstracted from a set of Device Application Information Models

Abstraction: process of mapping between a set of Device Application Information Models and an Abstract Application Information Model according to a specified set of rules

Application Information Model: information model of an Application, including data and methods

NOTE: An Application Information Model may have Representations expressed in specific operational protocols.

attribute: characteristic of an Entity

NOTE 1 Also called "property", "characteristics", "characteristics".

NOTE 2: The state of an Entity is determined by its Attributes

EXAMPLE: Name, Time, Location.

Concept: fundamental category of existence

NOTE: Also called "entity type", "category", "subsystem", "class".

EXAMPLE: Person, Car, 4-Wheel-Drive, Ford Explorer.

Device Application Information Model: technology (e.g. ZigBee®) specific Information Model of the physical device

entity: an instance of a Concept

Management Plane: part of the ETSI M2M System that carries the operations and administration traffic required for network management

EXAMPLE: Software download, Statistic collection, Set/Get of parameters that provide policy or configuration settings. Any and all network management functions are provided by a service provider.

ontology: formal specification of a conceptualization, that is defining Concepts as objects with their properties and relationships versus other Concepts

Operational Plane: part of the ETSI M2M System that offers capabilities (methods, data) that are needed for delivering the intended services (M2M Services and Application Services) from/by the Network Domain, Devices and Gateways

physical entity: tangible element that is intrinsic to the environment, and that is not specific to a particular M2M application in this environment

NOTE: Depending on the environment, the physical entity may be an appliance, a piece of furniture, somebody, a room of a building, a car, a street of a city, etc. To be part of the M2M/IoT architecture, a physical entity does not need to be connected through a direct network interface, or even to be identified through a universal identification scheme such as RFID/EPC global, provided it can be sensed by sensors that are supposed to be deployed in this environment, and possibly acted upon by actuators.

physical entity proxy: contextually identifies and represents a physical entity; it implements an executable version of the informational model of the physical entity and serves as an intermediary towards applications, providing them an interface for control and monitoring of this entity with the primitives defined in the corresponding model

NOTE: Proxies representing individual physical entities are all distinct software components at the same hierarchical level and do not contain one another, even if the corresponding entities have such containment relationships; for example the proxy of a room will not contain the proxy of a an appliance, even if this appliance is inside the room.

relation: specifies how Concepts (objects) are related to other objects stating a relationship among Concepts

NOTE: Also called "relationship", "interrelation" relations are part of an Ontology.

EXAMPLE: "is-part-of", "is-subtype-of".

Representation: expression of an Application Information Model in terms of the operational protocol of a specific technology (e.g. ETSI M2M, ZigBee®...)

Representation Interworking: process of mapping and synchronizing multiple Representations of an Application Interface

Set Of Things Representation: group of Thing Representations that share a common property or functionality

NOTE: A Thing Representation can belong to several Set Of Things Representations.

EXAMPLE: It can contain Thing Representations of:

- Things that radiate heat (radiators, electric appliances and even human beings);
- Things that provide lighting (lights, display screens and windows).

Thing: element of the environment that is individually identifiable in the M2M system

Thing Representation: instance of the informational model of the Thing in the M2M System

NOTE A Thing Representation provides means for applications to interact with the Thing.

Translation: combination of Abstraction and Representation Interworking

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in TR 102 725 [i.1] and the following apply.

NOTE: An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in TR 102 725 [i.1].

OWL Web Ontology Language
RDF Resource Description Framework

4 Introduction to semantic information on M2M data

4.1 Problem description

Release 1 of ETSI M2M defines a Service Capability Layer which is enabling transport of M2M data between devices or gateways and network applications. Release 1 provides an abstraction layer hiding the heterogeneity of M2M access networks and provides means for secure data transport. By design choice, the ETSI M2M Rel.1 SCL is handling only data containers without any knowledge of the data contained. The advantages of this approach are:

- a clean separation of data transport from data handling;
- focus on the generic, commonly needed functions of an SCL - thus avoiding applications-specific functionality to be included into the ETSI M2M standards.

More precisely, the current release assumes that applications know all details of the device installation they interact with, e.g.:

- how devices are identified as being useful/relevant for the application;
- what actions are performed by the device (e.g. what kind of data can be provided by the devices);
- how to interpret the data delivered by the devices.

While Rel. 1 of ETSI M2M already opens a lot of opportunities in the M2M area, there are a number of **limitations**:

- 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;
- device and application need to agree beforehand on a common definition of the exchanged containers as well as on the contained data. This makes re-use of M2M data across different applications difficult;
- there are only very limited functions in Rel.1 to discover which data are available in an SCL;
- 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;
- limited chances for ETSI M2M compliant platform providers to enable value-added services re-using M2M data;
- limited opportunities for treating different kinds of M2M data with different Quality-of-Service or by charging differently for them.

For operators and providers of an ETSI M2M compliant platform this is limiting their ability for offering new and innovative business models.

To overcome these limitations

- data transmitted in M2M services need both, the semantics and the level of abstraction, that could make it possible to provide them as a pool of common data available in a given environment and to share them between different applications, without these applications needing to know beforehand the specifics of these data (units, metadata, context, etc.);
- the physical entities that are sensed and acted (e.g. appliances, people, cars, rooms of a building, or more generally self-contained subsystems of a larger system that makes up the target environment) need to be modelled with a level of abstraction allowing the M2M system to treat them as generic entities, intrinsic to the environment and not tied to a specific M2M application.

4.2 Benefits of semantic annotation

By providing means to understand M2M data, the available business models can be greatly enhanced. For example, through offering additional semantic information about the data, platform provider can enable (and potentially charge for) the discovery of devices and data by semantic specification. Another possible business that can be provided would be to provide derived information from the provided raw data through intelligent processing, e.g. analysing the data, aggregating data across many different data sources, or to provide interpreted data as an additional service.

Support for semantically annotated M2M data and related advanced operations like discovery or resolution from real-world entities to sensors/resources and vice versa will for example enable the following:

- re-use of M2M data by many applications - data can be "brokered" by the M2M Service Provider;
- "write-once run-anywhere" applications (which automatically adapt to the specific device installation);
- simplified configuration of M2M applications and more intelligent adaptation to changing situations;
- easy adaptation in case of failures/changes of the available sensor sets;
- easy creation of generic tools (e.g. for visualization, data processing).

Such 'intelligent' applications will require some notion and modelling of the 'real world' in which they provide services. For example M2M applications are not interested in sensors and actuators themselves, but in what is being sensed by the sensors, or acted upon by actuators. The relevant level of abstraction for M2M data pooling should thus not be confined to individual sensors and actuators. It should rise to the level of physical entities that are being sensed by sensors and acted upon by actuators. Depending on the environment, these entities may be appliances, people, cars, rooms of a building, or more generally self-contained subsystems of a larger system that makes up the target environment. These entities are generic, intrinsic to the environment and not tied to a specific M2M application. They can be legacy appliances or completely passive "things" that need not be directly connected through a network interface, or not even to be identified through a universal identification scheme (such as RFID/EPC global).
However, in the context of ETSI M2M only those entities need to be considered, that can interact with the ETSI M2M System. For example a "room" entity can be sensed by the sensors in that room.

4.3 What constitutes semantic information and how can it be structured

4.3.1 Heuristic view

Figure 1 shows the hierarchy of semantic information, ranging from raw data to real-world context information. As the picture explains, we are looking at two kinds of data. First the data handled by an ETSI M2M System, e.g. a measured value. Secondly, the data that is describing a specific sensor including information on what kind of measurement is returned and what information in which context is actually measured. Both kinds of information are needed for the semantic support of M2M data.

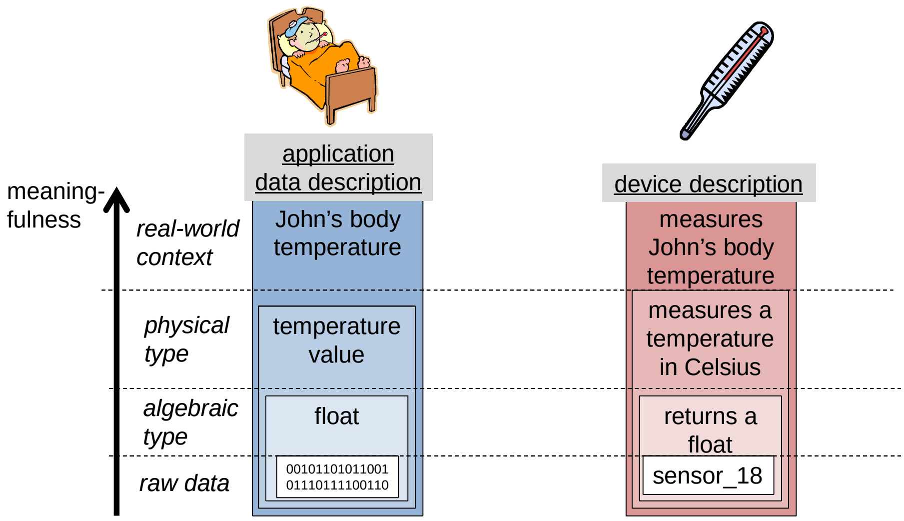

Figure 1: semantic information hierarchy

Note that the top level of the semantic hierarchy assumes a model of real-world context. Typically, such a context model would consider the world to be composed of physical ***entities***, and the state of the world is determined by the ***attributes*** of these entities. In figure 1 the entity would be "John", while the attribute would be "body temperature".

The support of ETSI M2M release 1 for semantic annotation of data is limited to the specification of a MIME type of data containers, which corresponds to the second lowest level. On the device description side, there is no support for semantic annotations at all.

Adding semantic information to a system can be done in various forms. In the following we discuss a few possible options. The goal is to explain in this study what can be done and through that to enable a decision how best to add semantic information. (See for example the Semantic Web lecture at http://www.sti-innsbruck.at/teaching/curriculum/semantic-web/ for a good overview about the semantic web).

4.3.2 Option 1: Standardized Data Types with implicitly defined semantics

Traditionally, communication systems have standardized data types to be commonly used between various applications. The semantic of this information might be defined through the respective standard. Unfortunately, this implicit mapping of a data type to a specific semantic meaning is not always given and might lead to errors.

EXAMPLE: The traditional email address "user@domain.tld" has a clear implicit semantics. Email addresses can be used when composing and sending emails. Unfortunately, email adresses have been used as identifier for user and devices in computer systems that do not necessarily provide a mailbox with the given email address. So the implicitly given semantics ("you can send an email to this address") is broken. In an M2M system, this relying on an implicitly given semantics can lead to errors.

4.3.3 Option 2: Standardized Data Types with some defined semantics

RFC 1738 [i.9] defines the syntax and semantic of the Uniform Resource Locator (URL). A URL has the format:

<scheme>:<scheme-specific-part>

The <scheme> part defined the usage of the URL and also determines how the <scheme-specific-part> is interpreted. The following schemes are reserved and define a mapping between the scheme and respective defined Internet protocols:

`ftp`	`File Transfer protocol`
`http`	`Hypertext Transfer Protocol`
`gopher`	`The Gopher protocol`

mailto	Electronic mail address
news	USENET news
nntp	USENET news using NNTP access
telnet	Reference to interactive sessions
wais	Wide Area Information Servers
file	Host-specific file names
prospero	Prospero Directory Service

As an example, the URL "mailto:user@domain.tld" specifies an address to which mails can be sent. The syntactically similar "ftp:user@domain.tld" defines an FTP server that should be accessed using the account "user". These examples show that semantics can be explicitly given as part of the naming or addressing schemes.

4.3.4 Option 3: eXtensible Markup Language (XML)

XML is a framework for defining markup languages. It is used to describe arbitrary complex data structures. XML defines a common syntax to represent data in an ordered, labelled tree. Similar to the Option 1, an XML-based markup language first defines the syntax of the data, not the semantics. XML Schemas enable a syntactical verification that a certain XML document is adhering to the defined schema.

As XML is more expressive with its ordered, labelled tree, there are again the possibilities of explicitly defining semantics through labels or annotations in the XML tree. As XML is a framework for defining markup languages, these newly defined languages can define the semantic aspects of its elements.

See [i.3].

4.3.5 Option 4: Ontologies

An ontology represents knowledge as a set of concepts and the relationship between the concepts. Thus using an ontology semantic concepts can be described. Annotating information with ontology references links the information to these concepts. On this basis, semantic tools and mechanisms like reasoning can be employed.

The following technologies are typically used as a basis for describing ontologies:

- **Uniform Resource Identifier (URI):** identifies things and concepts in the semantic web
  →URIs are used to identify resources in the semantic web. see [i.4];
- **RDF: Resource Description Framework:** an XML-based markup-language to represent the triples (subject; predicate; object) see [i.7]
  →RDF is used to make statements about resources;
- **OWL (Ontology Web Language):** a language to represent Ontologies in the Web (see [i.5]);
- **SPARQL:** Query language for RDF triples (see [i.6]).

For further details please refer to the respective W3C standards [i.3] to [i.7].

4.3.6 A concrete approach for ETSI M2M

An M2M system is set up to acquire data from a large-scale physical system and control it in return. Shared sensors and actuators are distributed as monitoring and control points through this physical system.

A set of entities, subsystems of the overall physical system, are defined as the components of the overall physical system that are relevant for controlled and monitored by the targeted application. These subsystems are distinct physical entities which are fully-fledged physical systems in their own right. The overall physical system is made up of the composition of these physical subsystems for what is relevant to the application at hand. The sensors and actuators are not target entities themselves they are used just as transparent intermediaries.

A target subsystem is defined as a self-contained subset of the overall target physical system or process that can be individually monitored and actuated, either indirectly through sensors and actuators or directly, if it is equipped with its own network connection. These two possibilities are not actually exclusive, and a subsystem with a network connection may still need to be monitored/controlled through complementary external sensors and actuators because the network connection does not provide access to the required data or functionality.

If we take as an application example home/building energy management, the overall physical system is in this case the home or building itself, and examples of the target physical subsystems are:

- appliances and devices including all types of pieces of home or office equipment;
- rooms of the building;
- Components of the building (including walls, roof, openings, etc.), as long as it makes sense to deal with them individually rather than as parts of a larger subsystem.

These mostly non-digital subsystems have to be integrated in the supervisory M2M system in a way similar to what is done with regular networked entities. This means they have to be identified and matched to an existing model that can be specific or generic, exact or approximate.

Figure 2 shows a possible modelisation of a home subsystem, which would be composed of an Appliance-category and of a Room-category subsystem.

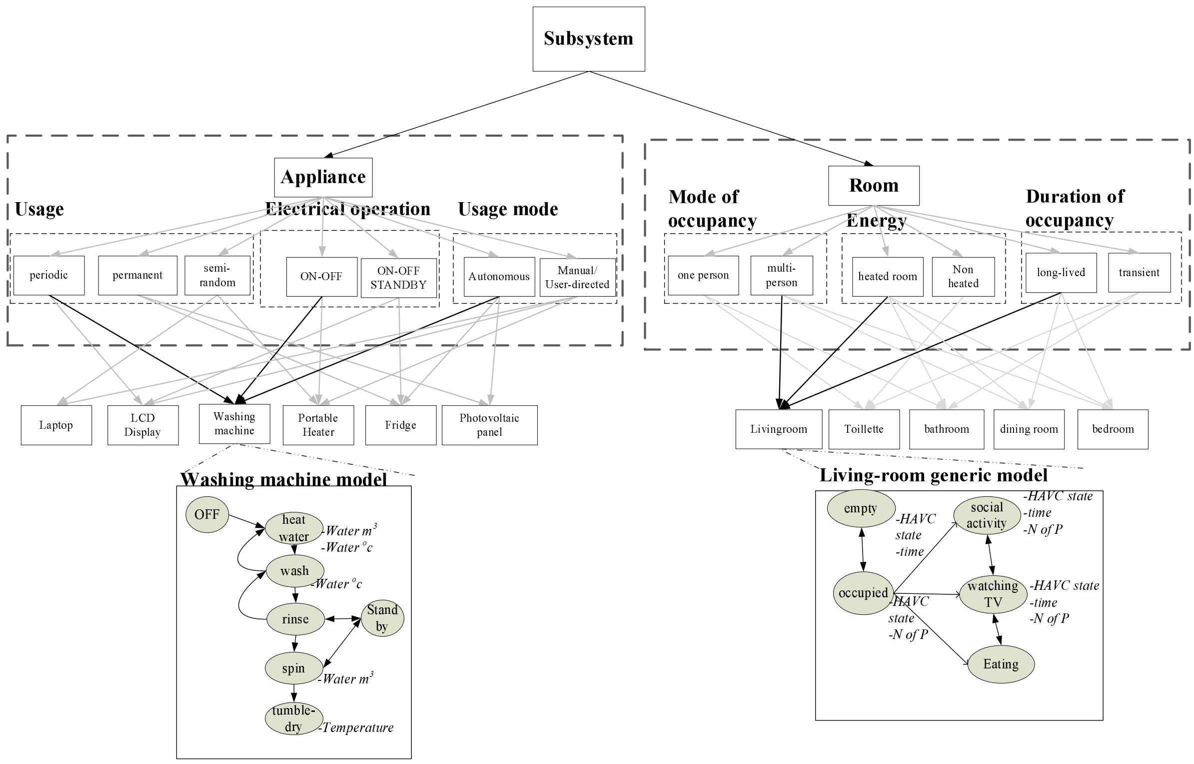

Figure 2: Ontology comprising two categories of subsystems (rooms and appliances) for the home environment

Architectural consideration

One of the goals of using semantic is to take into account 'things' in the M2M system. The idea is not to limit the M2M system to the consideration of a sensor network but to extend into an infrastructure that supports un-digitized "thing-computer" interfaces inspired from ambient and context-aware "human-computer" interfaces as if they were regular devices.

Devices can be identified by approximation to a very generic model, and the system should be able to integrate them on the basis of this minimal information.

For example, a camera is a single sensor that acquires data. The latter are analyzed by a "thing" recognition and monitoring software. We can consider then that every individual thing or physical entity within the field of view of this camera becomes a "networked thing", provided it can be recognized and monitored by this software This means it can have a presence on the M2M system, without requiring an RFID tag or even a digital optical code (such as a 1D or 2D barcode) for this. Thus the range of things that may become indirectly connected to the network can extend much further than sensor devices themselves, to all things that are individually identifiable by a sensor.

As the counterparts of sensors, actuators transduce numerical variables into physical ones. They enact modifications of the physical environment and the effects of these are either sensed directly by sensors, or indirectly, through passive things which are modified by the actuators. These new physical "actuator-to-thing" links complement the "sensor-to-thing" links.

The notion of bilateral "sensor to thing" links presented above is a simple abstraction of the multilateral reality of context sensing that should apply for the identification and monitoring of things.

Networked "things" may then comprise all "stuff" that can be sensed by pattern recognition software operating on top of these federated sensors *working together*, potentially overcoming their individual limitations.

When using basic sensors such as passive infrared, door contact or electrical sensors, both the detection of temporal coincidence of events from different sensors (as potentially coming from the same physical entity) and the application of simple filtering rules to these multi-sensor events are used. The consolidation of these events will then depend on the corresponding model of the originating physical entity.

An ICT system is set up to acquire data from a large-scale physical system and may control it in return. Shared sensors and actuators are distributed as monitoring and control points through this physical system; they are not target entities themselves, but rather some transparent intermediaries. The ICT system will "shadow" each of these nodes individually through matching software components (proxies) that will offer the required interfaces to the Things in this environment. The ICT system should have the capability to provide an automatic association with the entity proxy of the interfaces to the subset of sensors and actuators used as intermediaries for the monitoring and control of a given entity.

This can be illustrated through the reference architecture in figure 3.

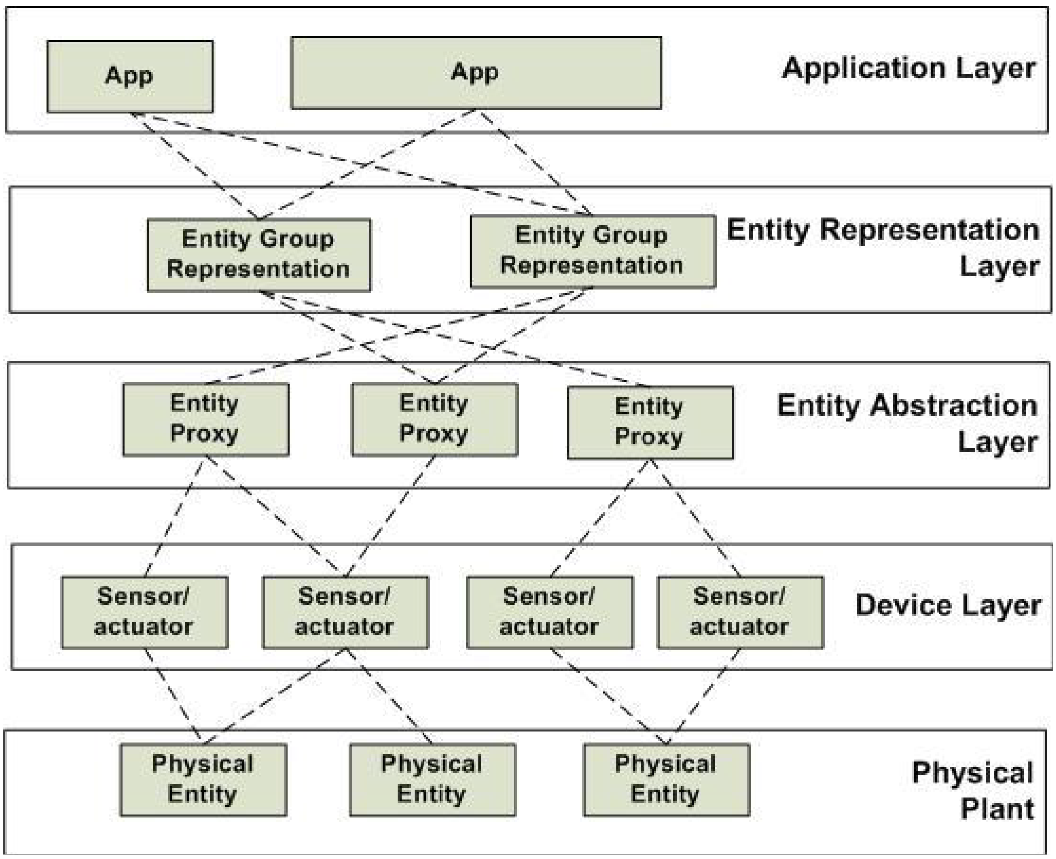

Figure 3: Reference architecture of the ICT system incorporating physical entities

An additional separate layer of "entity groups" is needed to represent aggregation or containment relationships between physical entities, with a 1-to-n or n-to-1 mapping to the physical entity layer. An entity group representation may thus link to several physical entities or a physical entity may link to several entity group representations.

If we take home energy management as an example, examples of the physical entities would be:

- appliances and devices of all types, including all pieces of legacy home equipment, such as a lamp, PC, etc.;
- rooms of the home;
- energy-relevant components of the home such as walls, windows.

The physical entity "lamp" may then be mapped to two entity group representations:

- "heating entities" group representation;
- "entities producing light" group representation.

For monitoring an entity, an application can obtain the instantaneous state of this entity as the discrete state of the corresponding entity proxy. This discrete state is estimated as a result of the fusion, aggregation, consolidation and classification of data from sensors associated with the entity.

For control purposes, an application can effect a change in the state of an entity through the entity proxy that relays this high-level state-change control order to low-level control data for the associated actuators.

Non-digital entities such as pieces of furniture, pets, or the home occupants themselves have to be identified and matched to an existing model that can be specific or generic, exact or approximate.

Example for monitoring of real-world entities that have mains connection:

- As for legacy appliances whose only available interface is that of their mains connection, this interface makes it possible to identify these appliances through the characteristic features of the patterns exhibited by their electric power consumption through an electric power sensor (like e.g. an oven showing a steady plateau pattern whereas a washing machine has characteristic peaks and troughs). When these appliances are identified and enrolled into the extended home network in this way, it becomes possible to monitor and control them as specific or semi-generic entities, even though this control is limited to the mains interface.

4.3.7 Summary

The described options above are some selected ways of adding semantics to information. For ETSI M2M we need to identify the right way of describing information and attaching the needed semantic information to it. In this study we need to identify the requirements and derive the design goals for an ETSI M2M support of semantic data.

For example, the implicit assumption of ETSI M2M is that most of the handled data is generated by small and simple sensors. In such a scenario it would be overkill to use extensive semantic descriptions just to describe the sensor information.

4.4 Semantic M2M and OBIX mapping

Introduction

In the subsequent section we describe a possible semantic data model for ETSI M2M. The design of this model has been guided by **two main principles:**

1) **Separation of the abstract information model from its representation in the ETSI M2M resource structure.** The information model proposed in this contribution is abstract in the sense that it is not yet defined how to represent it in the form of ETSI M2M resources. This separation of concerns hopefully simplifies the discussion about the semantic data model. Furthermore, the semantic data model will not need to be redesigned when future versions of ETSI M2M are released, only the mapping might have to be updated. Finally, an abstract model also serves as a set of requirements for future ETSI M2M versions - these future versions should be able to represent the abstract model as accurately as possible.

2) **Separation of domain-specific knowledge from instance-specific data.** ETSI M2M is designed as a standard to enable interoperability across different M2M domains. To achieve this goal also on the level of data understandability, the information model should be restrictive enough that any piece of information is expressed in a unique way, determined by (a) the **structure** of the information model and (b) **domain-specific** knowledge. Certain parts of the information model are always specific to application domains, e.g. naming of devices and commands. This separation of domain-specific knowledge from the data structure allows e.g. to find all the possible operations on a certain Thing (derived from the structure of the information model) without the need to have a-priori knowledge on e.g. naming conventions for operations of the domain (domain-specific knowledge).
   The information model proposed here imposes a rather strict structure on both instance data and domai-specific knowledge, and it defines how these two parts interrelate.
   The rationale between a strict separation of instance and domain knowledge is that the latter is exactly what needs to be agreed upon by the specific application domain in order to achieve interoperability within the domain. When separation of instance and domain knowledge is achieved, then the instance-specific data can be understood without any further a-priori knowledge, and any two applications connected through the ETSI M2M SCL can interact with each other just on the basis of the common domain model.
   In a representation of the abstract model in the ETSI M2M resource structure, the domain-specific knowledge could be stored centralized in one resource, while the instances are rather represented as resources of an application (e.g. for some physical Thing there is a resource representing it).

Further notes: The exact models are one possible approach that needs further discussion. In particular, the following points are subject to discussion:

- considering also instances of operations, in order to be able to monitor their execution
- modelling also lists of Things and values

Modelling semantic data in ETSI M2M

The information model proposal is shown in figure 4. Instances of this schema represent instance-specific data.

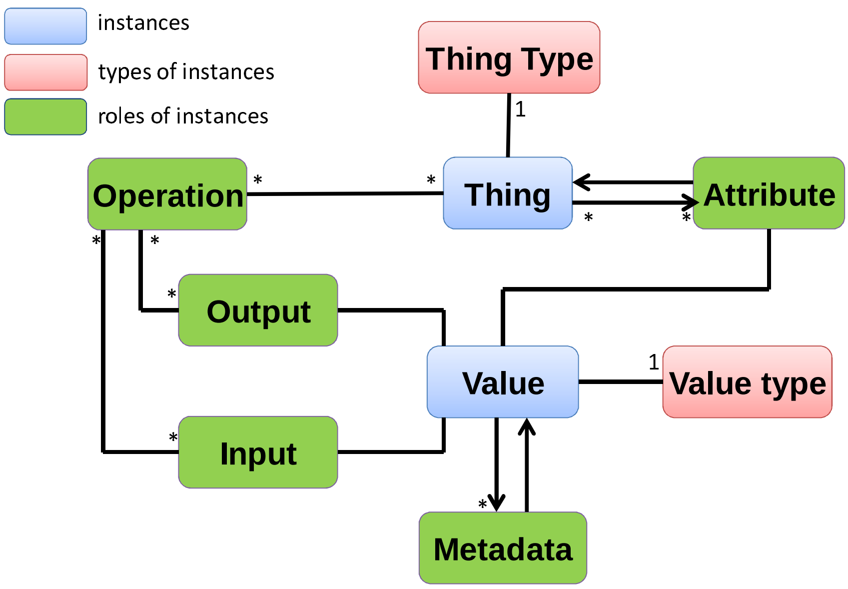

Figure 4

The central concept in this information model is the notion of *Thing*. Things are meant to represent arbitrary physical or virtual objects that might be relevant for M2M applications. Examples for Things are devices residing in local access networks, but also non-technical objects like rooms, persons, etc. can be modelled as Things. Things have associated a *type* and a number of *attributes* and *operations*. An attribute is a property of a Thing, and some of the attributes have a *value*. Values are used for "primitive" data types like integer, reals, or Booleans. Alternatively, an attribute of a Thing could point to another Thing. For example, a Thing "myDog" could have an attribute "age" which has a value, and another attribute "owner" which would point to a Thing that represents a human. While attributes of Things can potentially be read or written, operations are meant to be executed. Each operation needs a set of *inputs* and produces a set of *outputs*. The inputs and outputs again have a value. Values have a *value type* and possibly a set of *metadata*.

Metadata represents information about the value; examples are "accuracy" or "timestamp". Also each piece of Metadata has a value.

In figure 4, the two blue boxes (Things and values) represent information that is *specific to instances*. All other concepts - operation identifiers, input and output specifications, attribute specifications, metadata specifications, as well as Thing and value types - are *domain-specific* in the sense that every application domain might define its own set of Thing types, operation names, metadata identifiers, etc., which can then be re-used among all applications of the domain. These domain-specific definitions and their interrelationships would represent a typical *domain information model*. The structure of such domain information models is to a large degree determined by the overall information model in figure 4: a domain information model makes statements about which types of Things have which attributes and operations, which attributes can have which types of values, etc. The structure of this kind of domain-specific knowledge is depictein figure 5. Instances of this schema represent domain-specific knowledge.

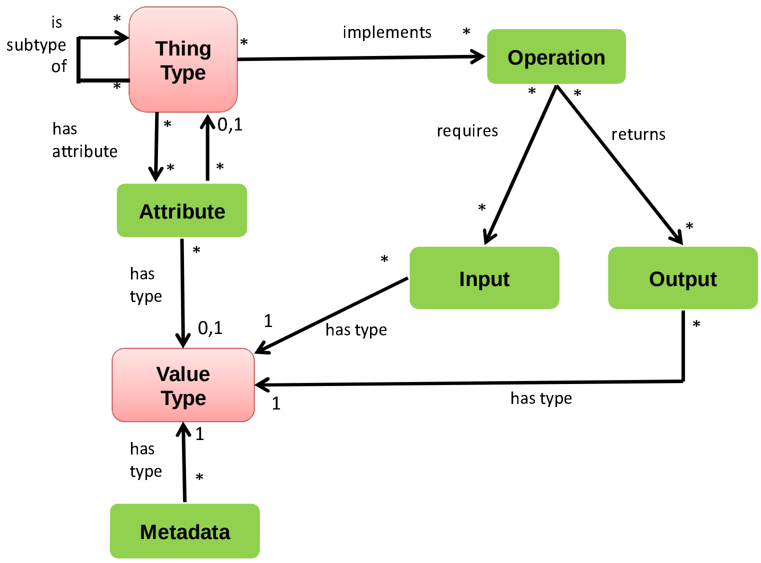

Figure 5

The information represented by an application domain model is what is typically expressed by ontologies. It is indeed recommended that identifiers of Thing types, operations, etc. should be realized as Uniform Resource Identifiers, and the typical triples known from ontologies can be used to express the above relations like "implements", "requires" etc.

The above information model is designed such that the usage of application domain models is optional, and also the level of detail of the domain model is flexible. In the simplest case there is no domain model, so each application uses its own identifiers for Thing types, attributes, etc. An intermediate solution could be that the application domain model serves as a "dictionary" for Thing types, attributes, operations, etc., but does not exactly specify which types of Things have which attributes and operations. It should be clear that the level of detail of the domain information model has a great influence on interoperability across different applications or even application domains.

Mapping OBIX "Open Building Information Exchange" to the Information Model

The purpose of this section to exemplarily show how existing M2M information models (e.g. OBIX based) can be mapped to the abstract model (denoted "ETSI M2M semantic model" or "ETSI model" for short in the following) proposed above. The feasibility of such mappings is important not only to demonstrate the wide applicability of the ETSI M2M semantic model, but also to be able to concretely make it the basis of an abstraction layer for various M2M access networks.

On OBIX

OBIX stands for "Open Building Information Exchange" and is a standard designed for information exchange in embedded software systems. Unlike the name might suggest, it does not contain special concepts that would limit its applicability to the domain of building automation, but it can be used in the context of a wide range of M2M information systems. It is also not tied to a specific binding, nor is it tailored to any specific M2M area network technology like ZigBee®/KNX/etc.

OBIX is based on XML, and its syntactic rules are kept as simple as possible. In what follows we explain the concepts of OBIX on an abstract level, without going too much into detail about how the concepts are expressed in XML.

The OBIX standard has three main ingredients:

1) An extremely simple basic object model.

2) A number of semantic conventions on top of the object model i.e. objects with special meaning.

3) Some syntactic sugar to ease the usage of these special purpose objects.

The **basic object model:** The world of OBIX consists of interlinked objects. Objects are identified by a unique identifier (a URI). Objects can contain other objects, and when Object B is contained in Object A, the role of B in the context of A is described by the *name* of B in A. Furthermore, there is the concept of inheritance - an Object can 'be' another Object. However, unlike in strongly typed programming languages, there is no separation between "Types" or "Classes" on one side and "Instances" on the other side - both are just Objects in OBIX. In the example above, B would inherit all Objects that are contained in A. In OBIX syntax, it is said that A is a *contract* of B, and B *implements* the contract A.

In addition to contracts, there are a number of pre-defined *facets* objects can have. Facets are special attributes used to express various metadata about objects. Examples are value ranges, human-readable names, or information about if the object is read-only.

Semantic conventions and syntactic sugar: There are a considerable number of special-purpose objects that serve for a wide range of purposes. Most of them are expressed by the usage of special xml tags (whereas for all user-defined objects only the standard <obj> tag can be used). The following list of pre-defined objects with special semantics is non-exhaustive.

- Objects representing primitive data types like integer, string, or date. For example, objects that implement the contract "obix:int" are regarded as integers.

- The Object "obix:ref" represents reference to other objects. Objects that implement this contract are regarded as references.

- Operations are objects which implement the "obix:op" contract.

- Objects for data exchange purposes like subscriptions, batch operations, server information, etc.

Mapping of OBIX to Semantic Model

As virtually any aspect in OBIX is modelled as an object, there certainly is no concept in the ETSI M2M semantic model that would in general correspond to OBIX objects. However, some of the predefined objects like operations or primitive data types have a direct correspondence in the ETSI model. Other special-purpose objects in OBIX do not need to be represented in the ETSI model, because they are used for aspects that are out of scope of that information model, e.g. subscriptions and other means of information exchange organization.

The fact that there is no distinction at all between types/classes and instances of objects in OBIX makes it difficult to define a purely syntactical mapping between OBIX and the ETSI model - additional knowledge is required about which objects are used as types and which objects are used for modelling instances. However, in practice it will be easy for domain experts to tell which objects are used as types.

Table 1

Obix objects	ETSI M2M semantic model correspondence
Objects serving as contracts of others	Thing types
Object A containing object B	Thing A has an attribute pointing to Thing B
Object serving as instances	Things
Operation objects	Operations
Contract objects implementing contract objects	Inheritance in the domain model ("is-subtype-of")
Error objects	(out of scope)
Primitive data type objects like <int> etc.	Value types
Objects implementing contracts of primitive data type objects	Values
"Unit" Objects describing physical types	Special kinds of metadata defined in domain model
Objects for: Server lobby, batch processing, watches, history, alarming, scheduling, security	(out of scope)

4.5 How is semantic content introduced into the ETSI M2M System

In general, ontologies are classified into four categories (from [i.2] Guarino, 98 Formal Ontology in Information Systems, as explained in the lecture from STI Innsbruck):

- **Top Level O., Generic O., Core O., Foundational O., High-level O., Upper O.:**
  describe very general concepts like *time, event, action, or dependency*, which are independent of a particular problem or domain.

- **Domain Ontology:**
  describe the vocabulary related to a certain domain by specializing the concepts introduced in the top-level ontology.

- **Task and Problem-solving Ontology:**
  describe the vocabulary related to a certain task or activity by specializing the top-level ontologies.

- **Application Ontology:**
  the most specific ontologies. Concepts in application ontologies often correspond to roles played by domain entities while performing a certain activity.

Giving this classification, there will be specific stakeholders that will provide specific kind of ontologies and related data models.

Upper Ontologies will be most likely provided by well-recognized organizations, e.g. standardization bodies, industry fora, or widely adopted ad-hoc standards. Although a number of such general ontologies have been proposed, there is none that has been accepted widely enough to be considered as a standard. Agreeing on a base ontology will help to enable horizontal integration and overcome vertical silos. Note that in the case of ETSI M2M it is conceivable that the base ontology not only covers the most general concepts, but also concepts that are specific to M2M communication while still being independent from application domains. One concrete initiative to mention here is the Semantic Sensor Network Incubator Group of W3C [i.8].

Domain Ontologies may be provided by vertical industries, while Task and Problems-solving Ontologies could be provided by service providers which provide re-usable solutions or services. Finally, Application Ontologies would need to be provided by specific applications to define certain aspects of their business logic.

Although ETSI M2M stays independent from concrete verticals and applications, the annotation of M2M data using ontologies of the latter three categories can be supported in a generic way.

4.6 Existing work on semantics that could apply to ETSI M2M

4.6.1 Overview

The following works or projects can be related to the present study, directly or indirectly, because they handle semantic data, support the development of semantic data, or allow the extension or exploitation of semantic services.

4.6.2 Existing semantic projects

SEALS

The SEALS Project is developing a reference infrastructure known as the SEALS Platform to facilitate the formal evaluation of semantic technologies. This allows both large-scale evaluation campaigns to be run (such as the International Evaluation Campaigns for Semantic Technologies) as well as ad-hoc evaluations by individuals or organizations.

Use of the SEALS Platform and associated technologies is free of charge and all code is Open Source (Apache 2.0).

Web site: http://www.seals-project.eu/.

Knowledge Web

Knowledge Web (KW) is a 4 year Network of Excellence project funded by the European Commission 6th Framework Programme. Knowledge Web began on January 1st, 2004. Supporting the transition process of Ontology technology from Academia to Industry is the main and major goal of Knowledge Web.

Woowl

Woowl.net is offering the possibility to get semantic information about words in several languages, by offering machine-readable word sets, based on synonyms and antonyms semantic relationships. It proposes also different functions applying on words and ontologies, including ontology alignment.

Web site: http://www.woowl.net/

5 Use cases for creation, management and usage of semantic information in the ETSI M2M System

5.1 Overview

5.2 Detailed use cases

5.2.1 Use Case 1 - Home Control

5.2.1.1 General Use Case Description

This use case demonstrates co-operation between two independent M2M applications. The co-operation is made possible because one application can find the other application through semantic information about the application's resources. This semantic information is available in the M2M System.

One application is a building management system (BMS) for a big apartment house. The BMS is operated by a building manager, e.g. the owner of the apartment house. BMS has knowledge about the blueprints of all the apartments in the house, e.g. it knows which heater is located in which room (heaters are assumed to be equipped with temperature sensors/actuators).

The other application is a home energy management system (HEMS). It has been subscribed by the tenant of one of the apartments. HEMS controls the heaters of the apartment (among other purposes).

Because HEMS can find the resources of BMS - e.g. the resource that represents the tenant's apartment and the heaters therein HEMS can configure itself automatically (and can adapt to changes over time) and doesn't require human configuration.

Finding the right resources in the M2M System is made possible through semantic annotation of the resources.

5.2.1.2 Stakeholders

- Building manager: is running a Building management system (BMS) for his apartment house.
- Tenant of an apartment: has subscribed to a home energy management system (HEMS) for his apartment.
- M2M service provider: is providing access to the M2M System for both applications, BMS and HEMS.

Building management system (BMS) is a M2M network application.

Home energy management system (HEMS) is a M2M network application.

5.2.1.3 Pre-conditions

The Building management system (BMS) is an M2M application that contains all the information needed to manage a large apartment house. In particular it contains the construction details of the tenant's apartment, where the doors and windows are located, where the heaters are, their capacity, etc. The BMS is used for overall control of the building, but information relevant for individual apartments (e.g. control of the heaters, built-in sensors for windows and doors) can be made available to authorized tenants. In case of fire, the complete blueprint of the house can be made available to fire-fighters.

In the M2M System the BMS makes its information available as M2M resources, similar to as if they were data transmitted by a device. E.g. the complete apartment, individual rooms, their heaters and windows could be represented as M2M resources.

A new tenant is renting an apartment in the house. As he is moving in, he also subscribes to a general-purpose home energy management system (HEMS) that promised a very efficient heater control. E.g. the HEMS always uses the best available electricity tariff and the heating is turned off when windows are open.

As part of the subscription, the HEMS is granted access to the respective resources used by the BMS in the M2M system. In particular, the building manager has permitted access of the tenant's HEMS to those resources of the BMS that are needed for energy management of the tenant's apartment (rooms, heaters, door-and window sensors, etc.). Other resources not needed for this task are not exposed to the HEMS.

5.2.1.4 Flow of the use case

The newly subscribed HEMS will immediately start discovering new devices in the apartment. Once the BMS has granted access, the HEMS will discover the resources of the BMS that are related to the apartment. Using the semantic description of the devices the HEMS can immediately find out about the available rooms, heaters, temperature sensors, etc. With this knowledge it can configure itself without any human intervention.

Since the BMS has configured its devices to be represented in the M2M System as abstract devices, the HEMS can use this information to immediately control the devices using the offered abstract command set. Consequently, HEMS does not have to understand the specifics (e.g. specific protocol) of a particular heater control.

Later, the building manager installs a new device into the tenant's apartment which can help in efficient energy management. This new device is also managed by BMS. Using the selection rule of the HEMS service, the new device will get immediately available to the HEMS. The HEMS will discover the new device and will use it to control the apartment's energy consumption.

5.2.1.5 Post-conditions

None.

5.2.1.6 Potential new requirements from this use case

- M2M system support for a common (e.g. per vertical domain) semantic data model (e.g. represented by Ontology) available to M2M application.
- 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…).
- M2M system provision of representation and discovery functionality of real-world entities (rooms, windows) that are not necessarily physical devices.
- M2M system ability support the mapping of control commands issued towards an abstract device to the concrete commands of a specific device.

5.2.2 Use Case 2 - Device plug and play

5.2.2.1 General Use Case Description

This use case applies with any verticals, below just take home automation as an example. The use case is about when a device is newly registered in a home, it will find its own character and its relationship with its neighbour devices and Things automatically based on semantic information within the M2M system without the interference of human being. For example, the house owner bought a lamp and a switch to the lamp for his house. Both the lamp and switch is enabled with wireless abilities to be able to communicate with the home automation gateway and other devices. The lamp is for the lobby and accordingly the switch is located near the entrance of the lobby. When the house owner has placed the lamp and the switch properly, a simple power-on would make the lamp and the switch work fine.

5.2.2.2 Stakeholders

- Home automation service provider: is providing home automation service by providing applications running on home automation devices such as gateway, lamp, switch, TV, air-condition etc.
- Home automation management system (HAMS): is a network application.
- Device manufacturer: produces devices as M2M nodes.
- M2M service provider: provides M2M service acts as a platform where all M2M nodes can register to.
- House owner: is a consumer of the home automation service.

5.2.2.3 Pre-conditions

The house owner has a contract with the home automation service provider for the home automation service. The home automation service provider has a business relationship with the M2M service provider and the device manufacturer. The home automation management system manages all the devices and their relationships registered in the house. Each device has its role and serves fixed services among all home devices.

5.2.2.4 Flow of the use case

When the house owner buys new devices for his house, the newly bought devices will register to the M2M service provider and expose to the M2M SP its role and functionalities including their semantic descriptions. According to such information, the HAMS will compare the semantic description of the new device with the semantic description of the existing devices in the house and judge their relationships by semantic inference. Then the HAMS will help establish the relationship between the new device and the device in the home and the relationship is maintained in the M2M SP. For example the HAMS finds that the lamp is to be controlled by the switch, it may then bind the status of the switch to the action of the lamp. If the status of the switch is ON, an "ON" command will be sent to the lamp automatically.

5.2.2.5 Post-conditions

None.

5.2.2.6 Potential new requirements from this use case

- 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.
- M2M entities ablility to expose their semantic description to the M2M System.
- 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.

6 Summary of all potential requirements

- M2M system support for a common (e.g. per vertical domain) semantic data model (e.g. represented by Ontology) available to M2M application.
- 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…).
- M2M system provision of representation and discovery functionality of real-world entities (rooms, windows) that are not necessarily physical devices.
- M2M system ability support the mapping of control commands issued towards an abstract device to the concrete commands of a specific device.
- 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.
- M2M entities ablility to expose their semantic description to the M2M System.
- 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.

7 Potential architecture alternatives

7.1 Device Abstraction

7.1.1 Architecture

Native devices (type d) can host several applications. For example, a ZigBee® device can have several on/off switches. Each switch is a distinct application and needs to be registered to the Gateway as well as the Network. As specified in the TS 102 690 [i.10] clause 6.1, the GIP capability provides interworking between non ETSI compliant devices and the GSCL.

Figure 6 shows a high-level architecture for supporting device abstraction. Native devices (e.g. ZigBee® devices) are first registered in the GSCL as native applications through the GIP capability. These native applications are then abstracted in corresponding abstract resources through a capability supporting device abstraction, which is called the Gateway Resource Abstraction (GRA) capability. Both native and abstracted applications are then registered (or announced) to the NSCL via mId interface. Both GSCL and NSCL have abstract resources in their resource tree.

This architecture provides both legacy M2M applications, which have access network specific knowledge, and standard M2M applications to have an access to native resources. The legacy M2M applications can access through the native applications while the standard M2M applications do through the abstracted resources.

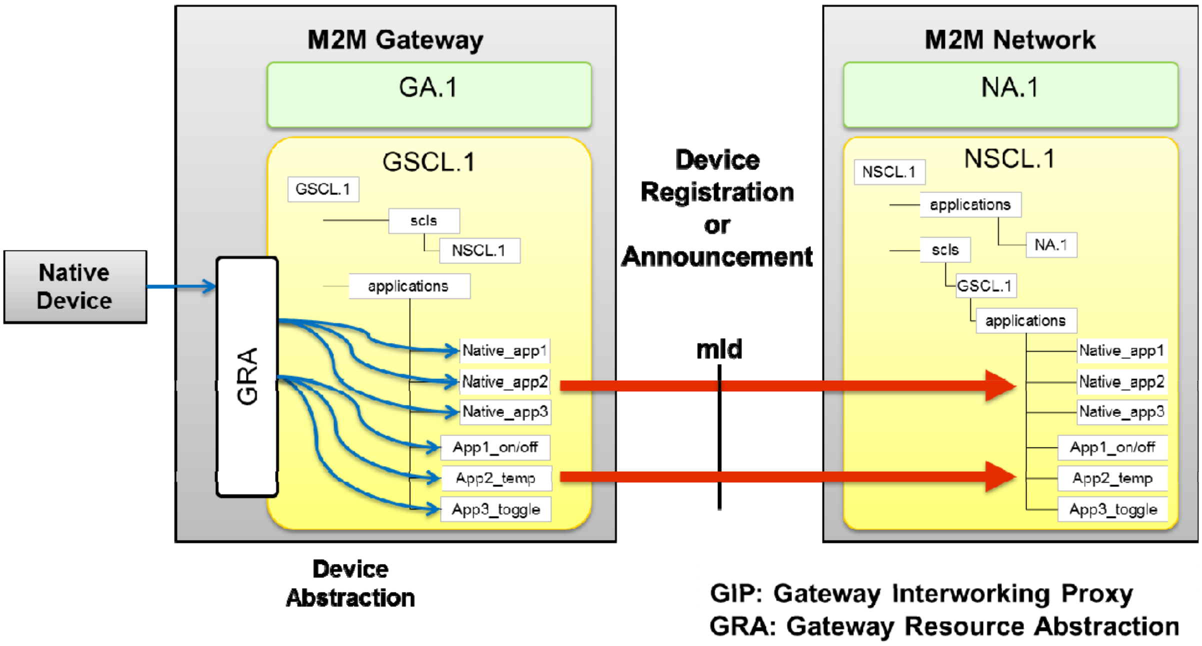

Figure 6: High-level architecture for supporting device abstraction

7.1.2 Interworking with legacy devices (d) through abstract devices

Figure 7 provides a resource-entity model that represents an M2M area network. In this model, each device in the network has native data and methods which are provided via access network-specific interfaces to applications. In order to provide interworking with M2M network applications that do not understand access specific technologies, the model defines an abstract application and linked it to its native application.

Since not all native applications are directly mapped to an abstracted application, the model provides 1 (native application) to 0..n (abstract application) relationship. All child entities of both native and abstract application such as interface, data field and method have the same 1 to 0.n relationship.

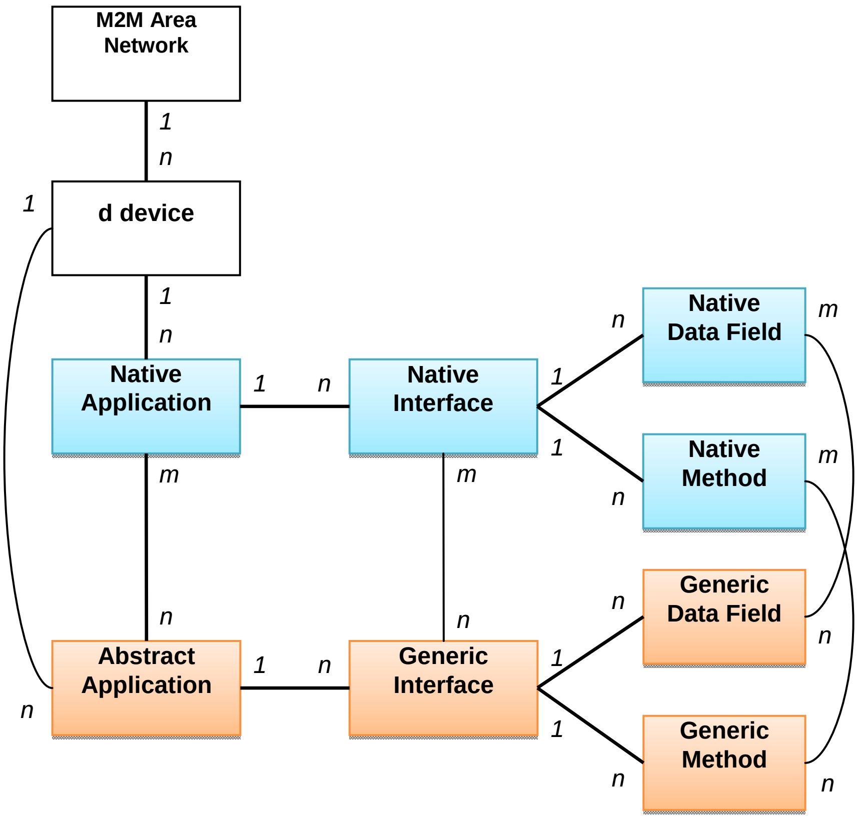

Figure 7: Generic entity-relation diagram for an M2M Area Network and its resources

This entity-relation diagram is applicable to the following M2M Area Networks:

- ZigBee®
- DLMS/COSEM
- Zwave
- BACnet™
- ANSI C12
- mBus

7.1.2.1 Native Resource

Native resource is an Application resource specified in the TS 102 690 [i.10] that will store network specific information about the Application. Same as application resource, native resource is created as a result of successful registration of an Application with the local SCL.

M2M network applications that understand network specific information can interwork with legacy devices (d) through this native resource.

7.1.2.2 Abstract Resource

An abstracted resource will point to the native resource hosted in another SCL or in the same SCL. The abstracted resource is a virtual resource which consists of a set of generalized attributes instead of local area network specific attributes, such as, the *searchStrings*, the *abstractLink* to the original resource, a set of *genericCommands*, which are visible to applications (e.g. toggle, on and off), and the *accessRight*. The purpose of the abstracted resource is to represent the original resource without any network-specific information, so that the issuer does not need to know about any prior knowledge of the used underlying network technology. An abstracted resource itself will be considered the same as other native resources that are located in the same SCL. When an abstracted resource is discovered, it returns a direct reference to the native resource.

7.1.3 Gateway Resource Abstraction (GRA) capability

At start-up of forming a local area network, the GIP capability detects new devices that have joined the network and creates original M2M resources on GSCL, which are specific to the local network technology. When the GIP capability creates the original resources, the GRA capability detects new resources, creates their corresponding abstract resources and registers them in proper SCLs.

The GRA Capability in the M2M Gateway is an optional capability, i.e. deployed when needed/required by policies.

The GRA Capability provides the following functionalities:

- Detects any additions of new native resources in the GSCL.
- Generates an abstracted resource from the native resource, which is non ETSI compliant resource.
- Links native resources to their corresponding abstract resources.
- Registers abstract resources to the NSCL.
- Subscribes to native resources to be notified any updates.
- Synchronize abstracted resources to their native resources.
- Provides functional mapping between the abstracted information (i.e. generic attributes and commands) and the underlying network specific information.
- GRA may either be an internal capability of GSCL or an application communicating via reference point dIa with GSCL. GRA can also be merged with the xIP (i.e. GIP, NIP and DIP) capability, so that provides resource abstraction and interworking capabilities together.

7.1.4 Subscription of Abstract Resources

Any xA in the ETSI M2M architecture should be able to create a subscription to an abstract resource. The xSC is responsible for managing the subscription. Any xA that subscribes to an attribute value can be notified when the value changes.

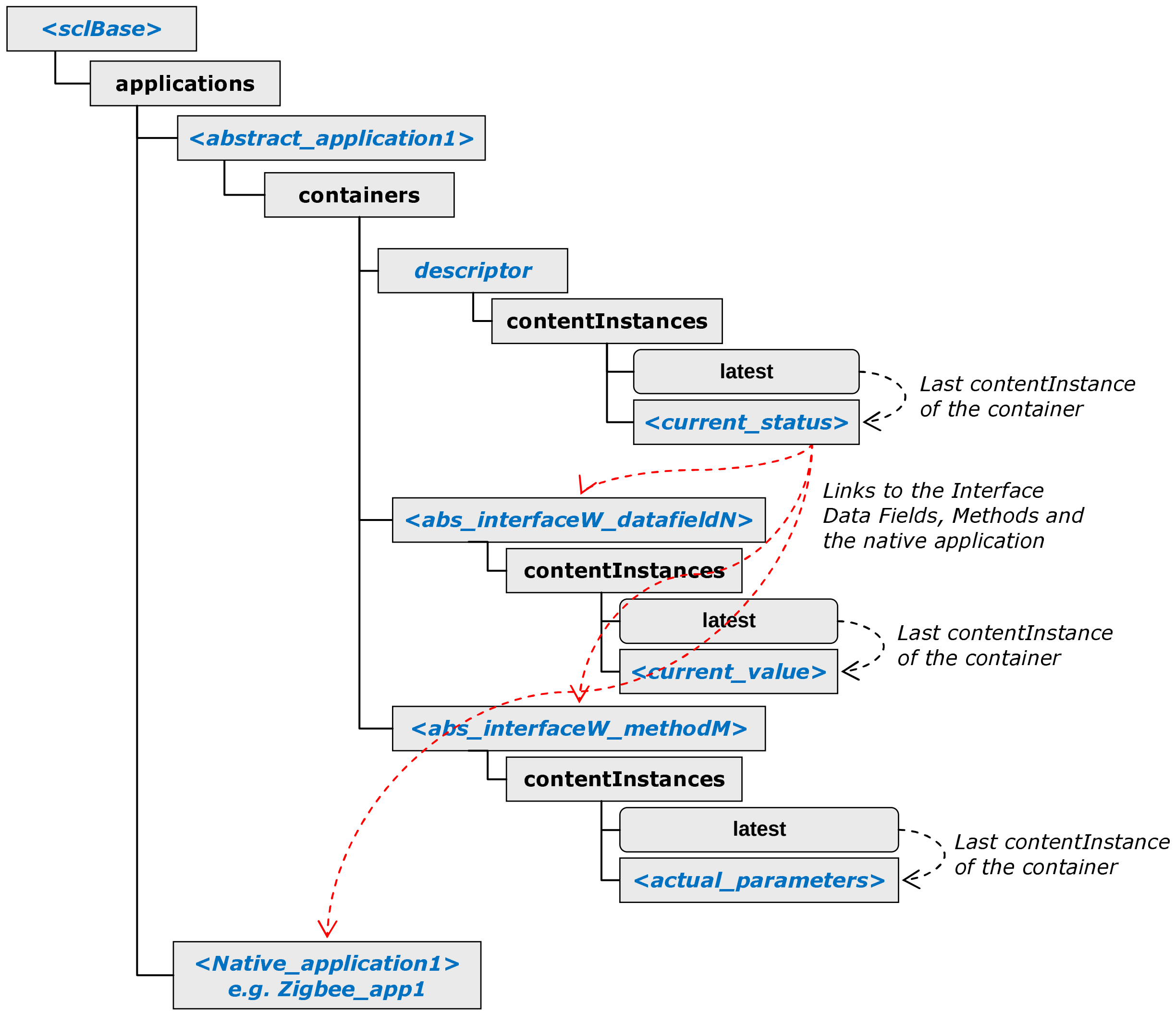

Figure 8: Linking an abstract resource to its native resource based on the ETSI M2M resource archtecture

7.1.5 Mapping Principle

This clause describes the mapping principles that are used to map a generic M2M abstract resource into a native M2M resource.

Each abstract application belonging to a Device (N.B.: they are not ETSI M2M Applications) is modelled with an ETSI M2M <abstract-application> resource. The URI used to access this <abstract-application> resource has the following format:

```
<sclBase>/applications/<networkX_deviceY_abstract-applicationZ>
```

The <abstract-application> resource contains an ETSI M2M <container> sub resource. The URI used to access this <container> resource has the following format:

```
<sclBase>/applications/<networkX_deviceY_abstract-applicationZ>/containers/descriptor
```

The <container> resource contains one or more <contentInstance> sub resource. The "content" attribute of this sub resource contains the representation of the Application. In particular, since an Application can implement several Interfaces, each of them modelled with ETSI M2M resources (see next bullet for description), the "content" attribute of the <contentInstance> resource may contain the URIs of the ETSI M2M resources representing these Interfaces. The URI used to access the <contentInstance> resource containing the current representation of the Application has the following format:

```
<sclBase>/applications/<networkX_deviceY_abstract-
applicationZ>/containers/descriptor/contentInstances/latest
```

The <contentInstance> resource pointed by the "latest" attribute of the contentInstances resource contains always the current representation of the Device.

Each Data Field and each Method belonging to an Abs_Interface is generalized from their corresponding native Data Field and method. Same as to the native one, they can be mirrored or retargeted.

If the Data Field or the Method is mirrored the ETSI M2M <abstract_application> resource modelling the Application contains an ETSI M2M <container> sub resource for each interface element mirrored (either Data Field or Method). The URI used to access this <container> resource has the following format:

```
<sclBase>/applications/<networkX_deviceY_abstract-
applicationZ>/containers/<abs_interfaceW_datafieldN>
```

or

```
<sclBase>/applications/<networkX_deviceY_abstract-
applicationZ>/containers/<abs_interfaceW_methodM>
```

The <container> resource contains one or more <contentInstance> sub resource. The "content" attribute of this sub resource contains the representation of the Data Field or the Method; for the Data Field it is its value, for the Method it is the actual parameters used for a Method invocation or the result of a Method invocation. The URI used to access the <contentInstance> resource containing the current representation of the Data Field or the Method has the following format:

```
<sclBase>/applications/<networkX_deviceY_abstract-
applicationZ>/containers/<abs_interfaceW_datafieldN>/contentInstances/latest
```

or

```
<sclBase>/applications/<networkX_deviceY_abstract-
applicationZ>/containers/<abs_interfaceW_methodM>/contentInstances/latest
```

The ETSI M2M <abstract_application> also has a link to its native <application>. The URI used to access the <native_application> resource containing the native representation of the resource has the following format:

```
<sclBase>/applications/<networkX_deviceY_native-applicationZ>
```

7.1.6 Abstract Resource Management (Add/Remove/Replacement)

Adding an abstract resource to an existing network is performed through the GRA capability. When a new native device is detected and added by the GIP capability, the GRA capability communicates with the GIP and initiates device abstraction to add a corresponding abstract device to the GSCL as well as NSCL.

In the ETSI M2M architecture, the replacement of a device from an existing network can be achieved through executing the device remove procedure followed by adding a new device to the network. However, removing and adding a device from/to the network cause the update of the all scls in the associated entities.

Replacing the original resource, for example due to maintenance or changing home automation system, does not remove the abstracted resource. Since the abstracted resource only contains access technology agnostic information, the SCL where the original resource is located does not issue any changes of the abstracted resource. After replacing the original resource, the SCL issues the update of the abstracted resources to update any changes.

Removing an abstracted resource, for example due to deregistration of an SCL (which correspond to a removal of the parent SC L resource), does remove the original resource and all the associated resource, when the original resource is removed, it is the responsibility of the original SCL, where the original resource is hosted, to remove the abstracted resource. If this is not done (e.g. because the original SCL is offline), the abstracted resource will be removed when it expires.

8 Conclusions

For ETSI M2M Release 1 no semantic support is planned.

Annex A: Application design support for semantic M2M data WI

A.1 References

ETSI TC M2M technical specifications, Release 1.

A.2 Application design

The development of business applications able to deal with ETSI M2M SCLs service platforms, will take into account some constraints defined in that standard and the derived degree of freedom that can be translated into flexibility for application design.

The following clauses describe some ETSI M2M standard items that can be seen as *tools* for application business logic design, providing some support for the study on semantic M2M data Work Item.

A.2.1 SCL base

The *<sclBase>* resource defines the name of an SCL entity operating in a service platform.

An ETSI M2M compliant service provider may sell standardized M2M services in more than one <sclBase> resource tree.

The application will be able to discover the SCL trees of a service provider and register itself to the wanted ones to obtain vertical services through ETSI M2M access interfaces.

Figure A.1 shows three SCL trees of the same service provider, addressable with the following URIs:

- http://www.m2mserviceprovider.com/sclUtilities/...
- http://www.m2mserviceprovider.com/sclEhealth/...
- http://www.m2mserviceprovider.com/sclAutomotive/...

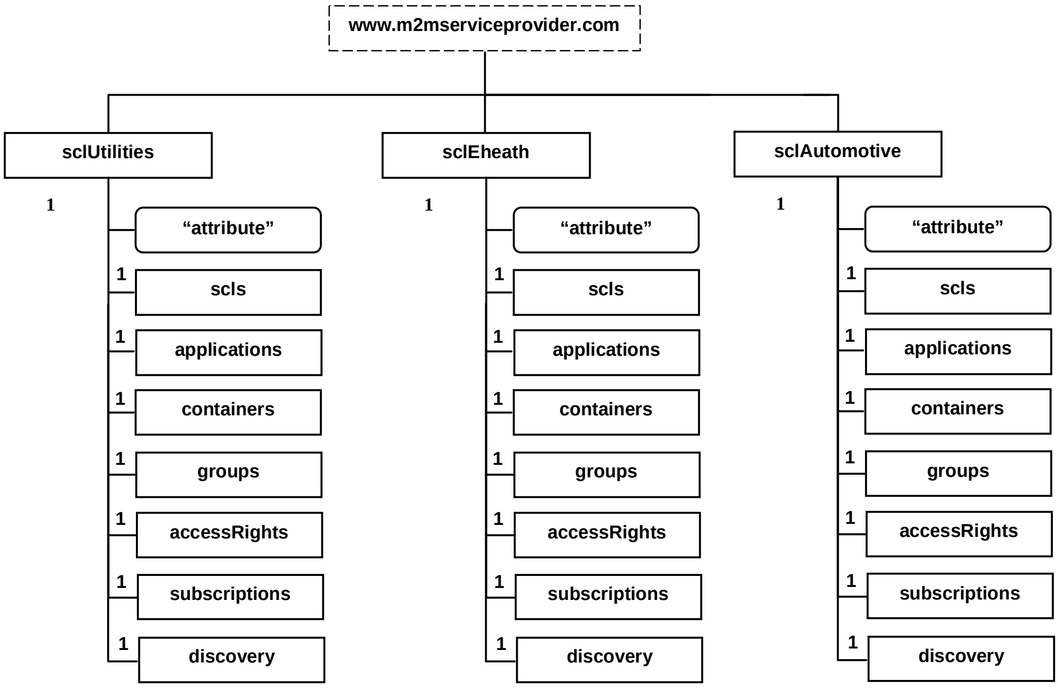

Figure A.1: SCLs of the same service provider

An SCL provider can organize the service platform architecture introducing different SCL resource trees then populate with the resources created by the applications.

It is important to underline that even for SCLs co-located in the same physical infrastructure, the N-to-N communications in-between is not foreseen by the ETSI M2M Release 1 technical specifications.

A.2.2 Resource name

The application developer can choose the name of any resource to be created in the addressed SCL resource tree. The choice of the resource name is very important for the overall organization of resources in particular for resource addressing, due to the presence of the name token into the URI.

A resource name could be:

- any kind of alphanumeric identifier defined according to some business logic rule, e.g. *6fdb17ddbd40f38*;

  http://www.m2mserviceprovider.com/sclEhealth/scls/6fdb17ddbd40f38

- a human readable identifier, e.g. *myApplication*:

  http://www.m2mserviceprovider.com/sclEhealth/applications/myApplication

- an SCL generated identifier when the name is not provided by the creator, the generation is implementation dependent, e.g. a content instance is automatically named by a fixed token part followed by a progressive numeric part, e.g. *ci_42*:

  http://www.m2mserviceprovider.com/sclEhealth/containers/sensor/contentInstances/ci_42

A.2.3 Containers

The SCL resource tree design will take care of the positioning of container resources into a tree, in fact they can be created in the following positions:

1) Under <*sclBase*> resource: http://www.m2mserviceprovider.com/sclEhealth/containers/container.

2) Under an <*application*> resource: http://www.m2mserviceprovider.com/sclEhealth/applications/myApplication/containers/container.

3) Under an <*scl*> resource: http://www.m2mserviceprovider.com/sclEhealth/scls/device/containers/container.

Positioning 1 is suitable for data that can be considered as common to many applications or SCLs.

Positioning 2 and 3 is suggested when containers need to be linked with an application or an SCL, creating them under these resources the link is self-provided.

A.2.4 Access rights

The access to resources belonging to an SCL resource tree can be **limited** by the adoption of access rights resources created by the application developer that contains flags enabling the following operations:

- "**READ**" : read enabled
- "**WRITE**" : write enabled
- "**DELETE**" : deletion enabled
- "**CREATE**" : create child enabled
- "**DISCOVER**" : discovery enabled

If a resource has not a link to an access right one, then the default access rights are applied.

To improve the overall resource access performance it is recommended to create access right resources in the **same** SCL tree.

A.2.5 Search strings

When a resource needs to be found using the specified filter criteria, one or more than one search string can be associated to a resource, enabling a matching criteria method for retrieving resources having common search string tokens.

Search strings can be assigned at resource creation and updating procedures.

Application developers should apply a search string definition strategy similar to one adopted for classic databases design, choosing appropriate search keys.

A.2.6 Content types

Content types are special search strings that can be defined for content instance resources, helping cataloguing and filtering of application's data, enabling huge data mining operations.

A.2.7 Partial addressing

ETSI M2M standard offers the possibility to address single attributes of a resource. This is called *partial addressing* that allows CRUD operations on attributes and the use of filter criteria to them.

It is a powerful tool for very selective searches.

A.2.8 Expiration time

The lifecycle of a resource can be controlled by the application provider or the service provider optionally setting the expiration time attribute that defines its deletion time.

The use of this resource life timer can be dangerous when the time-out value is not well configured and the resource is prematurely deleted with an unrecoverable loss of its content.

It is safe not to set an expiration time if not explicitly required by the business logic of an application because the deletion of an expired resource cannot be protected by access rights.

A.2.9 Maximum URI length

ETSI M2M resources are addressed by URIs created by the issuer application or SCL with the options described in clauses A.2.1, A.2.2 and A.2.3.

The length of provided URIs will not overcome the maximum value suitable for all the ETSI M2M entities involved in business service provisioning.

If an entity is unable to manage URIs with lengths exceeding some value, some resources become **unreachable** by these entities.

A.3 Impacts on semantic M2M data

The present document aims to study a new Service Capability devoted to discover, interpret and use the M2M data from different sources, without any kind of prior knowledge of that. This is essential to offer high-level M2M horizontal services and to develop open markets for M2M data.

The study wants to know the **amount of M2M data** of an M2M system to be considered as a kind of data discovery environment. The M2M data is mainly **application data** stored into **container resources** of a resource tree that is usually managed with the tools suggested in clause A.2.

History

Document history		
V2.1.1	December 2013	Publication