Data Model Needs for Building Device Interoperation

Bruce Nordman, Iris Cheung

Building Technology and Urban Systems Department
Lawrence Berkeley National Laboratory

February 1, 2016

TABLE OF CONTENTS

TABLE of TABLES

Executive Summary

Wide and simple interoperability of building components and devices is necessary to provide high functionality for occupants, and energy efficiency, in a cost-effective way. This minimizes hardware, software, and human time needed to integrate disparate and dynamic devices in buildings as well as human occupancy. The development of a suitable single data model for building interoperability is an important research area; past work has primarily developed data models for particular communications protocols or usage contexts. This paper presents a review of existing taxonomies and data models used in communication between energy-using devices in buildings, and a set of recommended best practices. Research such as this is needed to help move technology standards and devices towards more harmonized information exchange to help accomplish interoperability. Much further work is still required to assess more technology standards, more topic areas, and present the results to the many relevant stakeholders.

The core focus of this report is how numerical information and other data from sensors and devices could be described in a standard way. Examples of such underlying data standards include a list of sensor values and corresponding time stamps. Metadata can identify that the values correspond to a sensor, that the sensor is a power meter, the units are in kW, that it is located in a particular electrical panel, and that it measures lighting loads for a certain floor of a building. We conducted a survey of existing work on data models and schemas in standards bodies, and from community-driven open source development. Overall we reviewed the current status of over twenty organizations on metadata for energy related applications.

There are broad commonalities in the metadata information content across sources, but each organization utilizes somewhat different terminology and data structures to represent and communicate this information. These variations create interoperability challenges as each group forges a path that may be internally consistent to suit their needs, but inconsistent with other groups. The work presented here provides some structure and concepts to use in developing best practices, guidelines, and ultimately a common data model.

1. Introduction and Background

Any system for communicating among sensors and devices requires standards or internally consistent naming conventions of various sorts, to ensure that the data can be organized and understood. This report focuses on the mechanisms by which data are represented and named. This is analogous to facilities in human languages such as dictionaries, conventions for data representation (e.g. street addresses), and structure imposed on the environment (e.g. calendars). For any collection of devices, a local standard is needed. For general interoperability, the ideal is to have a single universal standard. If that is not possible, there should be as few as possible, with clear correspondences between them.

For data communicated within or about buildings, many types of data exist and are used. This report focuses on two specific, and overlapping, themes—data that are needed for sensors and sensing, and those involved in "Energy Reporting" (ER). For the former, communicating entities need to be able to declare what they are, characteristics they have, and have a name. Energy Reporting is the ability of an individual device to report on its own energy use and related data to the local network.

Data standards can be implicit, *ad hoc*, or defined in a technical standard. An implicit standard is one that people use out of habit without being stated. Use of typical measurement units (seconds, kWh, etc.) is an example of an implicit standard. An *ad hoc* standard is one created by an individual or small group of researchers to accomplish interoperability locally. Technical standards, created by recognized standards development organizations, are generally intended to be used widely—and when wildly successful, then everywhere. There are also proprietary standards, but these have limitations on control and usage that make them generally unsuitable for use when wide interoperability is the goal.

Most of the data elements we are seeking to address are covered by two types of entities: sensors and devices. In both cases, there is a mixture of data inherent in what the entity *is* (that can be set at the time of product manufacture), as well as data that are determined locally, at the time of installation or after.

LBNL has engaged in past research of several types that inform this investigation. One is user interface standards for energy-using devices (electronics power control, climate controls, and lighting). These can embody concepts that will be represented in both device data models and what humans experience in controls. It is therefore helpful if these correspond directly to each other, and LBNL sees it preferable to adapt device technology to what works best for humans rather than the other way around. Examples of such concepts are device power states, and representation of climate control setpoints or operational modes. Another area of past work is Energy Reporting (Nordman, 2014), built on the Energy Reporting (ER) Framework, which was developed by LBNL (Nordman, 2013).

The implementation of a common data model (or schema, including taxonomies) then becomes the logical next step, so that device-level data are reported in a uniform way, across all connected platforms, with standard names and fields. A consistent data model not only facilitates the analysis and interpretation of device-level data in the individual building, but also creates a consistent standard across the building sector for measurement and reporting.

2. Methodology

We investigated the needs for data schemas and how they would be used in an example agent interaction. The core purposes of the data schema investigation were to determine:

- Types of information to be represented, in general
- Specific data elements to include
- Names for those data elements
- Data encoding (units, enumerations, etc.)

Our approach was to survey existing standards for how they treat these for relevant information, analyze these for consistency, coverage, and quality, and then make recommendations for best practices and where further research is needed.

The general topics that seemed most in need of standardization are listed in Table 1.

Table 1. Data model topic areas in this report

• Sensors (identification and units)	• Energy Reporting
• Unique Identification	• Timestamps
• General Identification	• Location
• Classification	• Power States
• Local Data	• Static Power Data

3. Data Models

Data models are an essential foundation of any communications protocol in general, and even more so for building devices and systems that have complex characteristics to represent, and data about the physical world that needs to be translated to the information world. Successful interoperability between devices and systems generally requires a consistent data model. The goal is the clear, unambiguous, and bidirectional ability to translate data between two models, but that is challenging in many respects. Data models provide a common language by which to communicate.

We began this research by surveying existing standards with the objectives of first understanding what has already been developed, and then identifying research gaps that need to be addressed before a complete data model can be deployed. We focused our efforts on sensor integration and device energy reporting rather than general management of buildings, because the latter involves a wide scope and extends to other building components, such as security and fire systems, which are beyond the focus of this research.

Below we provide an overview on technical and *ad hoc* standards studied in detail for this report. It describes how the reviewed data models address some parts of the framework but not others. This process highlights the research gaps in this topic area and related ones including sensor information and device location.

Survey of Existing Data Models

The following sources encompass a wide variety, including: network application layer protocols, data model standards, unit representation standards, and research data standards. Their purposes are also diverse: dynamic building operation, energy program evaluation, scientific information exchange, and more. Some are explicitly data models, while others only do so implicitly. Below we briefly describe each source and assign each a short label for easy reference later in the text. The sources are listed alphabetically.

BACnet

BACnet (Building Automation and Control Networks) is a standard that enables communication among building components, principally those that are centrally controlled, and primarily HVAC systems. It is designed for devices or entities that are professionally installed as part of a central building control system, rather than for generic devices that are present in buildings. BACnet has only limited specifications for data modeling; most details in BACnet deployments are developed on a custom basis for individual buildings into which BACnet equipment is installed.

BEDES

BEDES (Building Energy Data Exchange Specification) was created for analyzing whole-building data, but has been extended to subsystems and even individual devices. The specification covers location within a building, basic product identifying data, equipment modes, and measurement units. We reviewed data from the 1.0 version of the BEDES dictionary[1], though recently an update to BEDES has been released.

CTA 2047

The Consumer Technology Association[2] created CTA 2047 to enable residential appliances to report their energy use to the local network. It was specified to be independent of device type or building type. The data elements focus on equipment characteristics and energy use information.

[1] available at: http://bedes.lbl.gov/home
[2] In late 2015, the Consumer Electronics Association (CEA) became the Consumer Technology Association (CTA).

Ecma

The Ecma International[3] standards organization created a standard for “Smart Data Centre” (SDC) operation. It is designed to facilitate interoperability between different types of systems in data centers for both information processing equipment as well as infrastructure for provision of power and cooling.

EMAN

The Internet Engineering Task Force had a working group on Energy Management (IETF/EMAN)[4], which produces several technology standards (called RFCs) that are the most focused on, and most developed for, the Energy Reporting topic. EMAN produced both a data model (in its Framework document), and three MIB (Management Information Base) modules that define how to implement that Framework in the Simple Network Management Protocol (SNMP). EMAN has a highly developed data model—perhaps too detailed—but has some useful features unique to it.

ENERGY STAR

The ENERGY STAR voluntary label covers over 40 product categories for HVAC, lighting, and electronics. ENERGY STAR maintains a Qualified Product List (QPL) for each covered category, with product attributes ranging from brands and models to power and energy consumptions, with products typically having many dozens of fields. Data fields in the QPL vary depending on the reporting requirements in the corresponding product specification.

FSGIM

The Facility Smart Grid Information Model (FSGIM) is a data model. It is oriented to integrating buildings with the grid, rather than our focus, of integrating buildings internally. The FSGIM defines a Facility as any kind of building or collection of buildings. The FSGIM is a project of The Smart Grid Interoperability Panel process (SGIP) and is being created by a committee, Priority Action Plan 17, tasked with the “development of a data model standard to enable energy consuming devices and control systems in the customer premises to manage electrical loads and generation sources in response to communication with the Smart Grid.”[5] This is being carried out by a committee of ASHRAE SPC-201P, which is writing the Facility Smart Grid Information Model (FSGIM)[6].

Haystack

Project Haystack[7] is an open source initiative that aims to define a common vocabulary so that building systems can be interpreted automatically by a variety of software and web-based

[3] Formerly the European Computer Manufacturers Association. As of the 1994 adoption of the name “Ecma International” the name is no longer capitalized and no longer an acronym.
[4] http://datatracker.ietf.org/wg/eman/
[5] http://sgip.org/PAP-17-Facility-Smart-Grid-Information-Standard
[6] http://spc201.ashraepcs.org/
[7] http://project-haystack.org/

applications. It promotes the use of standardized “tags” to uniformly name mechanical components in buildings (e.g. “condenser”, “humidifier”). It is relatively undeveloped in other domains.

HPXML

HPXML[8] is a data specification used by the National Home Performance Council (NHPC) and has some relationship to the Building America Field Data Repository (BAFDR). It is oriented to whole-building data, for planning and evaluation. It includes some measurement units as well as a few device types.

IEEE 1451

IEEE 1451 is a family of standards for representing data about sensors and transducers associated with those sensors, and for specifying how to transmit that data. It includes listings of units and ways to represent arbitrary units. IEEE is the Institute for Electrical and Electronics Engineers.

MODbus

MODbus is a relatively simple and old (but robust) protocol, found mostly in industrial and large commercial applications. MODbus data can be transmitted over a variety of physical media, and can be tunneled over an Internet Protocol network. Beyond a short list of fields, MODbus doesn’t particularly impose semantic meanings on data fields; this is done by device implementers on an *ad hoc* basis.

NILM

Non-Intrusive Load Monitoring (NILM) is a method to estimate device energy use patterns from measurements of entire circuits or entire buildings. Researchers in this area noted that a lack of standards about measurements made for NILM purposes impeded research and so created an informal standard for this purpose (Kelly and Knottenbelt, 2014).

oBIX

oBIX is a standard for building control and automation that is built on web services such as the REST protocol. It is oriented to the needs of and technologies for large commercial buildings. oBIX was created by the OASIS Open Building Information eXchange Technical Committee.

sMAP

The Simple Monitoring and Actuation Profile (sMAP)[9] is a collection of technologies for acquiring, storing, processing, and viewing data about energy use in buildings, as well as enabling better control. sMAP has available a collection of software drivers for interfacing to many existing technologies to enable the data to come into a common, easily-accessible

[8] http://hpxmlonline.com/
[9] https://code.google.com/p/smap-data/ and http://www.cs.berkeley.edu/~stevedh/smap2/

database system. sMAP imposes no requirements on metadata, but has some recommendations, though only a few. The results of this project could be readily incorporated into recommended practice for using sMAP.

TPEx

TPEx, the Technology Performance Exchange, is a database system for data about the energy characteristics and operational specifications for equipment and appliances installed in buildings, principally commercial buildings[10]. It includes data for device identification.

UCUM

The Unified Code for Units of Measure (UCUM) was created to "facilitate unambiguous electronic communication of quantities together with their units". It was created because the existing standards had inherent ambiguities in them, so UCUM was derived from these, and adheres to them as much as possible.

UPnP

UPnP is a set of standards to enable plug-and-play interoperability among Internet Protocol devices. It originated with information technology, but has been extended to include other types of devices. In this review we cover only the Sensor/Internet of Things (IoT) standardization effort currently underway within UPnP.

Volttron

Volttron at its core is a message passing system, with some semantics and applications built on top of that. The Volttron platform was developed primarily by researchers at PNNL, with a community of others also contributing to it. Agents (which can represent devices and services) publish and subscribe to topics on its message bus, which provides a single interface for detail abstraction. The published messages follow a specific format, but there is no standardized naming system or underlying data model to achieve uniformity. We were, however, able to find some relevant naming terms used in the "weather agent" in Volttron, which are discussed below.

XMPP

The Extensible Messaging and Presence Protocol (XMPP) is a network protocol to enable simple and effective presence (discovery) and messaging between entities. The XMPP standards committee is in the process of being extended to include infrastructure for sensors and related Internet of Things entities.

[10] https://performance.nrel.gov/

Zigbee

The term Zigbee refers to two different technologies. One is a physical layer protocol for transmitting data in a wireless mesh network. The second is an application layer protocol for communicating building energy information. This report considers only the latter.

The sources above are those used for analysis in this report. Other activities related to a standard data model are described below. These are university-based research projects, and are more *ad hoc* in nature and less well documented.

UCB

The OpenBAS (Open-Source Building Automation System) project at the University of California, Berkeley is creating a system to manage diverse devices in buildings that use energy or influence its consumption. The OpenBAS researchers have grappled with standard metadata for data streams being pulled into the sMAP system. This activity has focused primarily on the HVAC system, but also includes some lighting. The ideal experience for any building owner would be to have sensors and devices that report data directly in a standard format, "out of the box". However, the UCB researchers often encountered devices reporting in a way that is specific to a particular technology and vendor (as is common in buildings). To work around this they created translators (drivers) to bring the data into a common format. For HVAC devices, they found the system of tags in the Project Haystack model to be a good starting point, but not sufficient. They did create a system of naming of sensor data streams, but caution that such names are inherently *ad hoc* and local, and end up generally just repeating information in the tags (such as type, location, and function). The researchers also identified the issue that uniqueness in names is not guaranteed, and any automated system of creating names could easily create duplicates.

VT

Researchers at Virginia Tech are developing a multi-agent system with components that communicate with each other, using the Volttron platform. The deployed system that consists of three device types (HVAC, lighting, and plug-loads) is complete, and will be expanded to 10 device types in the next demonstration phase. Researchers have shared with us the taxonomy that they have developed for their three-device system, which is discussed below.

While this list of sources for this project is lengthy, there are still many more standards that merit review to more fully cover the topic area. Some of these are listed at the end of this report. These other sources may help confirm the recommendations of this report, may hold better approaches that can be directly referenced or adapted, or may show some commonality across protocols not visible in this projects set of sources.

General Findings on Data Model Content

In this section we report on our findings from reviewing the above sources, to show how the reviewed standards address each of the targeted topics listed in Table 1. By listing and evaluating the relevant data fields found in the standards, we develop:

- a better understanding on what is the state of information available
- where there is significant commonality among standards
- where there is significant diversity among standards
- identification of gaps in establishing a common and standard data model.

Each section below presents the relevant content from the standards, some discussion, and a recommendation for a common approach. The recommendations are mostly about specific data items which have a name, format, and meaning.

As these standards are highly diverse in their purpose and scope, the technical content is quite variable. The level of detail provided ranges dramatically, with some missing key information (e.g. the units of measure for a value) and others having voluminous detail and complexity.

A problem with many of the standards reviewed in this report is that they contain content which is consistent with, and so appears to be derived from, another one, but lack any explicit reference to the second standard. This can lead to differences in detail that are not obvious to the reader, or the derivative standard may omit details included in the source standard. This leaves unanswered the question of whether those details apply. It is always best for standards to identify the source(s) of content and explain how identical or not the two standards are[11]. If the originating standard is then updated, it becomes unclear if the referencing one should then also be updated and versioning better managed.

In previous LBNL research on data model needs (Nordman, 2014), Energy Reporting (ER) was a single topic area. For this report, we split the ER list into several categories, add the sensors topic, and also add location and timestamp as topics.

It is helpful to have standard terminology to describe entities in a building, methods for representing data in general, and syntactical conventions. These go across all topic areas and this report uses and recommends the conventions in the following paragraphs.

Data about building operation may come from or be sent to several types of **entities**. A **device** is an energy-using entity that has an atomic relation to the building—it is attached or detached as a unit, as a device with an electrical plug, or a hardwired interconnection to the building. A **component** is an identifiable part of a device that cannot be operated separately from the device as a whole. A component may be represented in communication protocols; example

[11] This is exacerbated by the fact that very few standards include information that documents the decisions made in the course of developing it. Such content would not be normative, but could be in an informative annex or a companion technical report. Even without this it is possible for standards to reference the source of content that is used directly, or adapted.

components are motors, displays, and sensors. A **sensor** is an entity that reports information about the physical environment. It may be stand-alone or a component of a device.

Any device or independent sensor could, in principle, communicate directly to the local building network such as the general Internet Protocol network installed in the building, and not require communication with entities outside the building[12].

Data can be represented in a number of different ways. Boolean, integer, floating point, and text string are the basic representations. In cases where integer values have an assigned correspondence to particular meanings, this is called an enumeration. A common example is the ASCII character set wherein 65 is "A", 66 is "B", etc. Enumerations have many advantages over the text strings they usually replace. They are more compact; they are unambiguous; they lack issues such as interpretations of capitalization, length, and spaces; and they are international in that the meaning can be readily translated into any language.

A generic issue is the choice of syntactical conventions. For example, some standards include spaces in data item names; others don't use spaces (because that would not work in their contexts) and instead use underlined spaces. Other standards just run words together, with capitalization to signal word beginnings. Finally, others use all lowercase or uppercase. The words-together approach with capitalization seems the best for general interoperability and clarity of reading.

Sensors

A sensor is "a device that detects events or changes in quantities and provides a corresponding output, …"[13]. For data models, the key issue is how a sensor appears in a modern data communications network, principally those that uses the Internet Protocol. To the degree that transducers or gateways exist to move the data into the network context, that is outside the scope of this discussion.

Several distinct aspects of what a sensor is or does are commonly combined and conflated:

- Underlying phenomenon being revealed (e.g. air temperature, electrical voltage)
- Unit of measure (e.g. degrees Celsius, kWh)
- Method of measurement (e.g. current transformer, hot-wire anemometer)
- Functional purpose of measurement (e.g. return air temperature, hot water supply temperature)
- Location of measurement (e.g. second floor hall temperature, exterior electricity use)

[12] In practice, some sensors and devices communicate through proxies or gateways, but that distinction is considered unimportant for this purpose.

[13] http://en.wikipedia.org/wiki/List_of_sensors – This is a remarkably extensive list ; it has 14 major categories that cover over 300 different types of sensors.

It is desirable to separate these aspects as much as possible. Some of these data are static and determined at time of manufacture (and knowledge of the brand/model of a sensor will determine some of these), whereas others are locally determined and potentially dynamic (e.g. location). Other values that can be attached to a sensor are static ones such as nominal accuracy or range of operation, or dynamic ones such as last date of calibration (all these not a focus of this report). Finally, there are the actual time-series values the sensor produces.

Sensors operate in diverse environments and applications such as vehicle components, scientific research, industrial equipment, portable electronics, as well as in ordinary buildings. The sensors installed in buildings will only rarely come into contact with sensors from other domains (weather data are a principal example of where this happens). For this reason, for the goal of interoperability within the buildings domain should be specified to cover residential and commercial buildings (not necessarily industrial, but not precluding that), for any country[14].

Sensors classically cover physical phenomena. However, some protocols include within the sensing regime, reporting of data about technological items, such as equipment status, data metrics, or human occupancy. The units or other representation of these other items are not standard in the way that units are, so should be covered by separate listing of names and enumerations.

Volttron Central (formerly OpenEIS) is a collection of code and documentation to enhance the interoperability of Volttron applications. Volttron Central defines conventions for sensor data, derived originally from Project Haystack. Volttron Central (OpenEIS) defines 46 sensor types. These cover temperatures, pressures, humidity levels, electrical power levels, gas consumption, and equipment status (including position and status). Also included are setpoints for temperature and pressure.

The OpenEIS github code omits significant information to know how to interpret and use these points. Temperature could be expressed in several different types of units (and Project Haystack notes that either C or F can be used). There is a separate document, “User Guide: OpenEIS Reference Code” that does mention that Fahrenheit is used (and for power, kW). Also missing are units for gas, humidity, and pressure. Boolean values are used to indicate equipment status (and so presumably on/off) and Occupancy (someone is present or not). It is unclear what the specific meaning is for some floating point values: valve positions, OutdoorDamperSignal, CoolingCall, TerminalBoxDamperCommand, SupplyFanSpeed, PumpVFDCommand, CompressorCommand, HeatingCall. Boolean values for first and second stage heating and cooling could indicate that these are active, or are enabled.

In the OpenEIS Application Documentation, there are several references to temperature being in Fahrenheit; one references area being in ft^2; the benchmarking application has **power** data in kWh, **gas** in kBtu, and **area** in ft^2; and the sensor suitcase application has **HVACstat** as HVAC:

[14] Interoperability with technologies and systems beyond buildings is possible, helpful, and helps reduce costs.

0/off, 1/fan, and 2/compressor. Conventions on metadata are scattered throughout the documentation.

The OpenEIS documentation lists for each sensor type: **name**, **data type** (boolean, integer, float), and **unit type** (dimensionless, unitless, temperature, energy, power, and pressure). It is unclear what distinguishes dimensionless from unitless.

Many standards reference units for measurement; this is not surprising as units are so basic to many technology activities. Sensors need to indicate the both their value and the units they report in, but units are also used in many contexts other than sensors. As power and energy are of special interest in buildings, discussion of their units appears in both this section and in the Energy Reporting section below.

BACnet has an enumerated list, **BACnetEngineeringUnits**, of nearly 200 different unit types[15], including volts, amps, watts, and watt-hours. It has separate entries in the enumeration for scaling prefixes such as kilo-. For non-power units it includes both SI and non-SI units.

DMTF has a list of over 50 unit types, **BaseUnits**, including metric and non-metric measurement units, as well as IT units like bits and bytes. They are in an enumeration and include Degrees C, Volts, Amps, Watts, and others useful for energy measurement. The enumeration appears to be specific to DMTF. A second enumeration is for **SensorType** and has only 20 entries, including Temperature, Voltage, Current, Power Consumption, and Power Production.

Ecma specifies SDC:**Temperature** and SDC:**Humidity**; the units for both can be any in a list of about a dozen which includes "C", "degree Celcius" (sic), "%", or "percent". It also includes terms such as Supply air temperature, Return air temperature, Set temperature, Outdoor air temperature, Supply air humidity, Return air humidity, and Set humidity. Ecma also has two relevant enumerations: **SensorType** including 13 (Power Consumption) and 2 (Temperature); and **BaseUnits** which can be 7 (Watts), 2 (Degrees C), 3 (Degrees F) or 4 (Degrees K).

As an information model, FSGIM is built up in a formal structure, for example, with a particular measure of the temperature of air having a name and being a member of the class **AirTemperature**. AirTemperature in turn references **Temperature**, and Temperature is part of the class of a **Measurement**.

For power and energy measurements, FSGIM specifies the units and scaling from tables as enumerations. In **UnitSymbolKind** it includes basic measures such as voltage, current, and power, as well as those for apparent and reactive energy. While any unit can be used, the basic energy unit in the FSGIM is Wh, and the basic unit of power is W. It includes features for static **PowerAttributes** such as nominal frequency, voltage, and whether AC or DC.

FSGIM references over 100 units, based on NIST Special Publication 811. The FSGIM documentation notes (6.5.1.4):

[15] as of 2004; see p. 411 of the BACnet documentation.

Only VA, W, var, VAh, Wh, varh, Btu, J, therm, BtuPerh, tonne, and none are normative for the FSGIM standard. The enumerated values of VA, W, var, VAh, Wh, and varh were changed from their original NAESB values of vA, w, vAr, vAH, wH, and vArH to make them consistent with the standard SI abbreviations for these units. The enumerated values of Btu and BtuPerh were changed from their original NAESB values of btu and btuPerH to make them consistent with the standard abbreviations for British thermal units and hours.

FSGIM also has an enumeration of **MeasurementKind** distinct from the units, though there is an obvious close correspondence between them. This includes values such as energy and power. In any case, since these are all enumerations, the exact spelling of the units is not critical as it would be if text values were used.

IETF RFC 3433 (2002) is titled "Entity Sensor Management Information Base". It defines a dozen units in an enumeration, **PhySensorType** (which is of data type **SensorDataType**), which include voltsAC, voltsDC, amperes, watts, and celsius. The data value is **PhySensorValue**. It also provides for describing the precision, a "textual description of the data units"[16], and a scaling factor. More recently, draft-jennings-core-senml-00 (November, 2014) proposes that an IANA registry be created for unit representation in ASCII and corresponding description. It lists about 50 in this version. It draws on SI unit representation (including from NIST 811 which specifies how to write units), and from UCUM (described later in this report).

OBIX has a sophisticated system for describing any combination of the seven basic SI units to any exponent, plus or minus. It further defines how to specify shorthand text descriptions, such as "kilowatt" or "celsius".

Haystack defines types of sensors, and separately indicates the units, since for some, more than one unit is possible. The units for electric power and energy are kW and kWh, as well as volts and amps. Originally, Haystack defined HVAC-specific measures (e.g. **zoneTempCoolSp** but these have since been replaced by use of a tag system. A separate set of sensors informs weather conditions, with the tags: **air temp** (dry bulb outside temperature in °C or °F); **wetBulb temp** (web bulb outside temperature in °C or °F); and **humidity** (percent humidity). Other sensor types include: **lightLevel** (in "lux" or "lumen"); **co2** (carbon dioxide in ppm); and **pressure** (static air pressure in H_2O or kPa).

sMAP metadata has a field **Properties/UnitsOfMeasure** that is a string to indicate the unit, as defined by a table of over 100 defined units[17].

HPXML uses **energyUnitType** for energy units as a string with a limited (about 20) set of possibilities, including kWh, MWh, Btu, kBtu, MBtu, and therms. **Temperature** and **Power** are floating point numbers (units are not specified).

[16] This apparently only for the convenience of human users, not for computational use.
[17] http://ar1.openbms.org:8079/api/query/Properties__UnitofMeasure

UPnP defines sensors according to a **DataItem** which has a field for a **Name** which is one of several dozen predefined types. In general, SI units are used. Of interest here **Current** – Ampere [A], **Humidity** – percent, **Power** – watts [W], **PowerSwitch** – one of on/off/sleep, **Temperature** – Celcius[18] [C], and **Voltage** – Volts [v] (plus a separate Voltage_dc). It does not include a unit for energy (perhaps because it is not a conventional instantaneous measure).

For lighting, VT specifies a lighting **brightness level** (0-100%) and RGB color. **Room temperature** and **temperature set point** are listed but without units specified.

XMPP includes several dozen types of sensors, and for each, a field **Unit** for the units it is encoded in, including metric and non-metric units. The units are specified as a text field.

Zigbee[19] includes **UomType** which is an enumeration of several dozen units which includes all the core energy/power measures as well as temperature in Kelvin (Temperature) and Degrees Celsius (Relative Temperature). It also has an object called **UnitType** for "end device control target reductions" which has about a dozen enumerated entries of which includes kWh, kW, and Watts (but not volts or amps).

The UCUM standard provides standard ways to textually represent units. Table 2 below shows ones that are, or might be, important for buildings and energy (The full units of measure documentation makes clear that watt is "W".). The descriptions are from UCUM "where they exist".

Several systems allow units to be scaled by factors of 10, including DMTF, EMAN, and FSGIM, and IETF/RFC 3433. Almost all of these range from -24 to 24; that is, from 10^{-24} to 10^{24}.

Over all these standards, units are variously defined by enumerations, text strings, or implicit in the data item being represented.

Sensors are a key element of data for building systems. As such, the data model for sensors should be derived from a conventional standards development organization (SDO) that has wide visibility and recognition. The approach should be well-grounded and minimize the potential for ambiguity.

Project Haystack is not a standards organization. The tag approach from Haystack can be useful in those fields that have key/value pairs (some use of key/value pairs seems essential, given the diversity of needs of different applications). The Volttron Central naming system mixes two distinct concepts; the underlying sensor type, and how it is used (e.g. ReturnAirTemperature implies the type of sensor, units, and location). At its time of manufacture, a sensor can be programmed with information on what it *is* and what it *does* (e.g. measure air temperature), and so always be able report that information. Where the sensor is located (e.g. in return, supply, mixed ducts) is only determined later. It would be better to separate these two aspects, by location or function; this would also bring together different types of sensors (e.g.

[18] Celsius misspelled again!
[19] ZigBee Alliance, Smart Energ Profile 2 Application Protocol Standard, 2013.

temperature, humidity, air speed) that are in the same place. Project Haystack itself separates these into different “tags”.

Table 2. Common UCUM Units (excerpted from Common UCUM Units[20])

UCUM_CODE	Description of the Unit	UCUM_CODE	Description of the Unit
A	ampere	t	metric ton
atm	atmosphere	us	microsecond
{clock_time}	clock time e.g 12:30PM	mA	milliampere
{count}	count	mbar	millibar
{CPM}	counts per minute	min	minute
m3/s	cubic meter per second	mo	month
d	day	{mm/dd/yyyy}	month-day-year
Cel	degree Celsius	{#}	number
[degF]	degree Fahrenheit	Ohm	Ohm
{fraction}	fraction	Pa	Pascal
Hz	Hertz	/h	per hour
h	hour	/s	per second
{index_val}	index value	%	percent
J	joule	[psi]	pound per square inch
L	liter	{ratio}	ratio
L/h	liter per hour	s	second
lm	lumen	V	volt
lm.m2	lumen square meter	wk	week
m/s	meter per second	a	year

There are other efforts to organize and categorize sensors that have had more time and attention than Project Haystack and likely more adoption. Some of these are described below.

The IEEE 1451 set of standards is designed to address this very topic of sensor metadata standardization. It is generic, not specific to the needs of buildings and energy. For sensor units, it includes an enumeration, which for the 2010 version of 1451.7 had 28 entries. IEEE 1451 includes entries for current, voltage, frequency, relative humidity, power, temperature, and time. Values beyond 28 are reserved for future use, so it could be extended.

IEEE 1451 uses SI units (with some exceptions such as an option to use days instead of seconds for time, and oddly a battery status value where “0=OK” and “1=low”). It is missing a value for energy, presumably since energy occurs over time, so a simple sensor would not report it directly. A later table lists the Joule as the standard unit for Energy. IEEE 1451 does allow either Celsius or Kelvin for temperature.

[20] https://loinc.org/usage/units

For energy, kWh is the most common unit used in buildings and electricity; however, the time period of an hour is not an SI unit, so neither is kWh. That said, IEEE 1451 does include C for temperature, for convenience, and that is a fixed numeric calculation from K. kWh is also a fixed numeric calculation from J, with a 3.6 million multiplier.

UCUM (Unified Code for Units of Measure) was created to "facilitate unambiguous electronic communication of quantities together with their units". It was created because the existing standards had inherent ambiguities in them, so UCUM was derived from these, and adheres to them as much as possible. It is based on NIST Special Publication 811, the NIST Guide to the SI.

UCUM includes only SI units in its core; it does include non-SI units in a separate and distinct format. This raises the question of whether non-SI units should be used at all in a standard data model. If only SI units were to be used, that would require entities that today only report non-SI units to have their data converted before being brought into the new system. Legacy devices will always require some sort of driver or gateway for protocol conversion, so this would be simply an additional function of the gateway/driver. Another issue is that many units that appear to be SI are not. For example, the base unit of time in SI is the second, so that the unit of energy is the Joule, not, as commonly used for electricity, the kWh. A kJ is about a third of a kWh so a suitable

For values that have a number of discrete possibilities, standard enumerations should be used, not text. Enumerations are beneficial as they are self-documenting and for otherwise boolean values allow for "missing" values. Since any enumeration could have data missing, that should be a consistent value, and as missing data may well end up being zero by virtue of never being filled in, zero is a logical choice for missing[21]. For enumerations which have some logical ordering, then higher values should be "higher" in the ordering. For example, "on" is considered to higher than off, so should be a higher value, as having off be encoded as 1 and on as 2. Occupancy could be considered binary but since we may eventually have occupancy sensors that include counts rather than just binary indications, a better term for this would be **Occupied**.

Other enumerations known to exist are equipment status, damper position, and heating/cooling called for (and by stage). Volttron Central also has items for OutdoorDamperSignal, TerminalBoxDamperCommand, CompressorCommand, PumpVFDCommand that need clear encoding of meaning. Specific standard enumerations should be found or developed for these.

Recommendation

Any data item should be one of four types, to be called **ItemType**[22]: **Sensor**, **Enumeration**, **Text**, **Float**.

[21] And, for example, zero is used for missing in: https://datatracker.ietf.org/doc/draft-jennings-core-senml/

[22] There is nothing special about this choice of term name, but **DataType** seemed likely to be already used in many contexts.

Use a text field, **Units**, with one of the following (third only if first two do not apply):

- A text entry from UCUM
- An enumeration value from Table A.1 of IEEE 1451.7-2010,
- A descriptive text string enclosed in brackets or braces per UCUM as appropriate.

This is simple and unambiguous.

Identification, unique

Electronic systems need methods (identifiers) to unambiguously differentiate between and identify physical world entities such as the devices and sensors in our scope. These IDs might be generated by the entity, by a local server that manages IDs (and may assure uniqueness), or created manually by a person. However, once determined, the identifier should be used for all purposes. The UUID[23] (RFC 4122; Leach et al., 2005) is used in sMAP, Volttron, and EMAN (which calls it **identifier**). Haystack specifies only that an **id** must be locally unique. CTA 2047 includes **UID** which is said to be unique but not necessarily a UUID (though as a string of arbitrary length, it could be a UUID). XMPP has **Unique Device ID**. IEEE 1451.3-2003 has a UUID, but it is different from the IETF UUID. VT has a field called **uuid** but it appears to be the device MAC address.

BACnet uses an **Object_Identifier** of type **BACnetObjectIdentifier** that "shall be unique internetwork-wide" so is essentially a UUID, but no external standard is reference.

Some standards use a network address in addition to or instead of a UUID; common examples are device Media Access Control (MAC) addresses (**MgmtMacAddress** in EMAN; **device address** in VT) and Internet Protocol (IP) addresses (**MgmtAddress** in EMAN; **device IP address** in VT). VT also covers **legacy device address** for BACnet, Modbus, etc. EMAN also has **MgmtDNSName** for DNS name. XMPP has **GSM Adapter ID fields**, **IPv4**, **IPv6**, **Node ID**, and **RFID**.

A device serial number by itself is not unique as it may be replicated with a different model or different manufacturer; however, in combination with the brand and model fields, it should be unique. It is included in: **Instrument/Serial number** (sMAP), **SerialNumber** (HPXML), and **Serial Number** (BEDES).

DMTF defines a **DeviceID** as "An address or other identifying information to uniquely name the LogicalDevice." Thus, it could be any method for unique identification described in this section, or others.

A building may contain multiple instances of the same type (brand/model) of device (perhaps many, as in the case of light sources), so individual identification is definitely needed. It is also

[23] A UUID is a Universally Unique Identifier. This is a system of locally creating very long numbers that are highly improbable to be created elsewhere. This ensures that IT systems know when two sets of data refer to a single entity, or to different entities.

necessary to have a single identifier for each device so that if information about the same device from two different sources is available, they can be properly matched up. This is why many standards use the UUID-Universally Unique IDentifier, a 16-byte value. Note that this is for the device as a whole (or sensor if stand-alone). A component (which could be a sensor internal to a device) may have its own UUID.

Recommendation

Identifiers other than UUID are commonly used for unique identification, such as serial numbers or asset numbers. As there are many potential types of such variations, we propose a text element that can hold a series of keyword/value pairs. Common keywords: SN for Serial Number; Asset for Asset Number. Locally-determined text names are covered below under Local Data.

For fields use:

- **UUID** – RFC 4122 – 16 bytes – unique numeric identifier
- **LocalIdentity** – text – keyword/value pair for additional identification data

Identification, general

Energy data are much more useful when accompanied by information identifying the corresponding device in a *general* (not individual) sense. Some of the identification information can be set at time of manufacture and be fixed, such as the device's brand and model. Other data, such as a local name and other local attributes (e.g., location) are not known before installation and can change over time.

Many standards have free-form text for the name of the company that created the product: **vendor-identifier** (a 2-byte numeric value) and **vendor-name** (BACnet; 120, 121); **Vendor** (FSGIM); **VendorName** (MODbus); **Instrument/Manufacturer** (sMAP); **Vendor name** (VT); **ENERGY STAR Manufacturing Partner** and **Brand Name** (ENERGY STAR); **Manufacturer** and **Make**[24] (BEDES); **Manufacturer** (HPXML); **Manufacturer** and **Brand** (NILM); **Manufacturer** (XMPP; to include the name and a URL for the company); **Manufacturer** (DMTF); **deviceManufacturer** and **deviceVendor** (Haystack; second one "in case vendor and manufacturer are different"); **MakeModel** (CTA 2047, though it also can include the serial number, which is individual, not general, identification; "brand" is said to be a synonym for "make"). In BACnet, vendor_id is a universal Organizationally Unique Identifier, in this case handed out by ASHRAE. OUIs deserve further inquiry; IEEE does this most extensively.

Next is information about the model of the product: **model-name** (BACnet, 70); **Model** (FSGIM), **ModelName** and **ProductCode** (MODbus); **Device model number** (VT); **Instrument/Model** (sMAP); **Model Name** and **Model Number** (ENERGY STAR); **Model** (DMF and VT); **ModelNumber** (HPXML); **Model** (NILM); **BRAND** and **PRODUCT LINE / FAMILY NAME** (TPEx; in a

[24] This defined as 'Equipment identification indicating manufacturer and or high-level category of equipment'.

different place it has **MANUFACTURER**[25]). XMPP has **Name** but it is unclear if this is a model number or a local name.

Energy Star also has a field, **Additional Identifying Information**, for "SKUs, UPC codes, retail numbers, and/or descriptions…". NILM has **gtin** for Global Trade Item Number and **version**. n a developer section, TPEx has **UPC** (for Universal Product Code). DMTF has **Part Number**, **SKU**, **Tag**, and **Version**.

HPXML includes **ModelYear**. BEDES includes **Year Installed** which can be important for models that have the same number over time but with different characteristics (though note this is date installed, not manufactured). NILM has **year_of_manufacture**, **year_of_purchase**, and **dates_active** (this as NILM is for time-series data and devices may get replaced periodically).

ENERGY STAR uses a **PD_ID** field, which is a unique number assigned to each product entry.

XMPP includes a URL, but it is for the manufacturer as a whole, not the particular model of product.

Several standards provide for pointers to manufacturer-provided data about the product; a web page is a natural entity to point to. CTA 2047 has URI for this very purpose[26]. None of these standards specify the nature of the data found.

Devices and stand-alone sensors should be able to report information about their manufacturer and model number, so that external information about it can be brought to bear, and to aid in the physical identification and management of devices. Components, including sensors, should have the option to report this, but not be required to do so. All such entities should have a manufacturer and model number listed. One problem is that there are commonly many different ways to write each of these. For the manufacturer, suffixes such as "Inc.", "Corp.", or "Ltd." should be omitted when not needed. For model numbers, spaces can be included for clarity of reading, but are not used for computational purposes[27].

Some products have a brand distinct from the manufacturer that may be needed for proper identification, e.g. HP LaserJet 3xi, or IBM Thinkpad X10. The last example is informative in that brands are sometimes sold from one company to another. Model numbers are presumably unique within a brand, but might not be within a manufacturer generally.

Recommendation

Some products have additional information to identify them, such as a second model number, more for internal purposes, or a version number for a product, or model year. For any of these,

[25] From data online, it appears that the company name might appear in either of these TPEx fields; those who submit data are not consistent.

[26] CTA 2047 does state: "URI is typically maintained by the manufacturer, vendor, or a service provider. Note: The URI can contain information necessary for communications between the device's native messaging and protocols such as OpenADR or SEP 2 through the use of, for example, an XML schema. It can provide additional information relating to the functions the device supports, including vendor specific functionality. "

[27] This is similar to what gmail does in ignoring periods in email addresses for the @gmail.com domain.

the hardware design may vary according to this value, even if the consumer-facing model number does not change. As there are many potential types of such variations, and many products lack them, we propose a text element that can hold a series of keyword/value pairs.

The manufacturer of any product could publish data about their technical specifications that were in a standard format, readable by both people and machines, and updateable by the manufacturer as new data became available. Such a resource could be simple but widely useful. A method to accomplish this would be for manufacturers to provide a web page for each product model, and have every product be able to report the URL for the appropriate page. This information would be available to anyone, whether they own the product or not, and readily displayable on web browsers. The page could have XML tags for machine readable data (the format for this data not yet specified).

As identification can apply to a device, a sensor, or a component, the word Entity is used in the term names. For fields use:

- **EntityManufacturer** – text – name of Manufacturer, generally without suffix (e.g. Inc.)
- **EntityBrand** – text – name of Brand if different from manufacturer, otherwise empty
- **EntityModel** – text – model number/name
- **EntityIdentityGeneral** – text – keyword/value pair
- **EntityURL** – text – URL of additional data about device from manufacturer

Classification

Many standards have small, anecdotal, or application-specific sets of device types within them, but do not seek to be comprehensive, or harmonized with other standards. One example is BEDES which has **Electronic Equipment Type** which has just two entries, though elsewhere characteristics of half a dozen types can be included, with sourcing to Energy Star. Energy Star itself lists the **Product Type** in its data sets, but this is limited to the types of products currently covered by Energy Star specifications. Some standards provide text fields without indicating standard content for them: **device-type** (BACnet, 31); **Device Type** (FSGIM and VT). **Technology Category** in TPEx has about 20 entries for product types it covers.

NILM provides four different classification mechanisms: **traditional** (similar to end use), **size** (just small or large), **electrical** (based on nature of signal seen on wire), and **google_shopping** (a hierarchical list from Google with thousands of entries).

HPXML can identify an **endUseType** that is more generic than a device type (eight entries in this case). BEDES can reference data that is submetered to heat, cool, plug, or DHW, which map to end uses. FSGIM has a field **EFacilitySystem** with over 30 types of building systems, including the traditional end uses, but also many specific to particular commercial building types.

Any device should be able to report what it is, in a fundamental sense. Based on extensive research, LBNL created an enumerated list of device types in 2013, in response to the apparent lack of any existing list of device types that is both simple, and universal (UDC) (Nordman and Cheung, 2014)[28]. The list has 92 entries, to be numbered consecutively starting at 1 (zero meaning missing). UPnP uses this list[29], though extends it with some device types that are referenced in other UPnP specifications when they are more specific than the UDC list.

Using numbers rather than text names avoids the possibility of misspellings and makes the list language-independent. For example, item 24 on the list is “Refrigerator”, and a later one is “Sensor”; the list does not address components.

Many devices will want to report additional data about themselves, and the existence of UDC does not impair doing that. HVAC is an example in which more detail is often desired (the UDC lists 12 HVAC device types), but that complexity is HVAC-specific, not universal.

Recommendation

We propose to use:

- **DeviceType** – enumeration – from Nordman and Cheung, 2014.

Local Data

Local data refers to informal data that is created when a device is installed. This can include fields for device identification, location, and other data. Examples may include “alan-pc”, “2nd-floor”, or “division_name”. VT uses the fields **Nickname** and **Device ID (unique)** for this category of data. CTA 2047 includes **Name** which is to be set by the manufacturer but can be changed by the user (example they give is “TV” as shipped, changed to “Bedroom TV” locally). DMTF has **Name**; EMAN has name which is “An [RFC6933] entPhysicalName”.

EMAN also provides for a **RoleDescription** text string, and set of text strings of **Keywords**, both with no defined structure or meaning to standardize around. sMAP also includes a namespace for **Extra** tags that “may have any value”.

XMPP has **Name** but it is unclear if this is a model number or a local name.

DMTF has **IdentifyingDescriptions** and **OtherIdentifyingInfo** which together form a set of keyword/value pairs for any other useful information, including “… the Operating System's user friendly name for the Device …”.

FSGIM has **EndDeviceAsset** which has a single field **name** which is “any free human readable and possibly non unique text naming the object.”

Other more general identification fields offered by the reviewed data models include: **Object_Name** (BACnet; 77), e.g. “Floor3ExhaustFan”; **description** (BACnet, 28), e.g. “Third floor

[28] The report on this project is available at http://nordman.lbl.gov.
[29] The LBNL report does not include the numeric associations; the version in the UPnP standard does. The ordering of the items is the same in both lists.

bathroom exhaust fan"; **MajorMinorRevision** (MODbus). NILM has **original_name** but does not indicate what this refers to.

Recommendation

Each device in a building should have a text name used to identify it colloquially. Example uses of this data element, LocalName, could be "Joe's Computer", "2nd Floor Refrigerator", and "Basement Light". Some buildings have other local data elements, but as these are diverse, are put into a keyword/value text string.

- **LocalName** – text – arbitrary string for convenience and use by building occupants and/or managers
- **LocalOtherInfo** – text – keyword/value pair for additional data locally determined

Location

Location has two fundamental aspects:

- location on the planet of the building in question
- location of a device or sensor within a building

Some standards cover one or the other of these; others cover both. In the context of this project, all entities are in the same building so that location of the building as a whole is not of interest.

The IETF defines a system for representing relative locations (RFC 7035), but it is really designed for use within a city or region, not within a building. RFC 4589 defines a list of location types; these are oriented to building types, though some could be interpreted as a type of room (e.g. office). RFC 4119 defines locations on the planet using addresses ("civicloc"), but adds a few more detailed elements applicable here, as summarized in the table below (excerpted from RFC 4119).

Label	**Description**	**Example**
LOC	Additional location information	Room 543
FLR	Floor	5
NAM	Name (residence, business or office occupant)	Joe's Barbershop

Some standards specify room types, though in all cases they seem anecdotal and not comprehensive. BEDES includes infrastructure-type room types (e.g. "crawlspace") along with HVAC locations (e.g. "supply chamber"). FSGIM specifies a mixture of locations with some being building types and some room types. Zigbee has about 20 room types oriented to office buildings in its Building Automation Standard. XMPP includes **Apartment** as part of **Address**.

NILM includes fields for **room** and **floor**, but does not specify their content other than that they are strings. For Location, sMAP also includes **Floor** and **Room**, as well as **Text** and **Description**, in addition to geographical location entries. BACnet has a freeform string for **Location**; this is in the context of the individual device, not the building, so is presumably location within the building.

Some standards use general text strings to store data for location within a building. DMTF uses **Name** and **PhysicalPosition** for this. Other standards have text strings for general purpose use that could include location information; EMAN is an example of this.

Recommendation

There does not yet seem to be an obvious standard for representing location within a building to adopt, so for the time being, this is specified as a text field of keyword/value pairs. Further guidance should be developed for standard keywords to use and their meaning, with obvious candidates being floor levels, room numbers, HVAC zones, and x-y (or x-y-z) coordinates.

We propose **LocationLocal** to distinguish it from identification of the building itself. This to be a text of keyword/value pairs, with **room** and **floor** recommended for use within this if applicable. Use:

- **LocationLocal** – text – keyword/value pairs

Timestamps

Any power or energy data needs a timestamp to locate the data in time. There are a huge number of formats for dates and times in common use, as well as complexities introduced by time zones, daylight saving time, and the like. It is possible to translate between formats, but that is burdensome and prone to introduce errors.

There are two methods for specifying absolute time that have the widest use in standards today, and seem suitable for future use. One is "unix time" and the other specified by the IETF in 2002. Unix time is "defined as the number of seconds that have elapsed since 00:00:00 Coordinated Universal Time (UTC) , Thursday, 1 January 1970"[30]. The value of this is currently about 1.4 billion. The IETF standard (RFC 3339) is based on ISO 8601[31]. Times of this form look like, for example, "1985-04-12T23:20:50.52Z". One difference between the two is that unix time does not account for leap seconds while RFC 3339 time does. The IETF standard directly provides for representing time zone details, which unix time does not, but the IETF format is directly readable by humans, whereas unix time always requires translation before that can occur, which could include presenting in a local time.

OBIX uses time consistent with RFC 3339, calling it **abstime**, though does explicitly reference the source. A sMAP repository has timestamps for each data entry. These are Unix millisecond timestamps. In sMAP this is called **ReadingTime**.

[30] http://en.wikipedia.org/wiki/Unix_time

[31] International Organization for Standardization, Data elements and interchange formats – Information interchange -- Representation of dates and times", ISO 8601:1988(E), June, 1988. (updated in 2000).

CTA 2047 uses the form “dddd-hh-mm-ss” for relative times. In the IETF context, **Timestamp** in SNMP v2 (RFC 2579) uses relative time based on system uptime, in the form of time ticks (in hundredths of a second). It also specifies **DateAndTime** in a format which looks like “1992-5-26,13:30:15.0,-4:0”. This was superseded by RFC 3339. One point this raises is that **Timestamp** as a term is used for both absolute and relative time, though the dictionary definition would seem to support absolute time.

BACnet uses **start-time** (142), **stop-time** (143), **local-date** (56), and **local-time** (57). Elsewhere in BACnet, the **Date** and **Time** are each encoded into four bytes (including for date the day of week). Volttron has individual fields for elements of date and time, rather than single composite fields, and does not appear to follow any particular standard.

An issue raised by time measurement is the potential need for very high resolution data, and hence data types which can accommodate this. Some studies of power quality for AC systems have found that timestamps measured in nanoseconds are required, which requires careful attention to the precision and size of the underlying numeric representation to assure that the needed granularity can be achieved.

Recommendation

TimeStamp is an obvious term to use, though in many cases the field name will be to identify the event the timestamp is referring to, so will be different. Either unix time or the RFC 3339 time format seem suitable for use. Unix time is more suited to arithmetic, and is much more compact. RFC 3339 time is human-readable. Conversion between the two is not difficult, and standard code is available to do this for many languages.

- **TimeStamp** – either text per RFC 3339, or number in “unix time”.

Power States

Several standards include fields for the status of equipment. This is most highly formalized in EMAN which includes a facility for representing a variety of series of **power states** in an IANA registry. A device can include zero, one, or many power state series in its own reporting. The registry is initially populated with three: IEEE 1621, ACPI, and one called **EMAN** but derivative of a Cisco scheme[32]. BEDES includes **Equipment Operational Mode** which is a mixture of power states and operational states. DMTF has a list of 19 **PowerStates**, though many are either commands to change states or transitional states. ECMA has **Power status** which references **PowerManagementService.PowerState**. Energy Star refers to several possible **Low-power Modes.** VT has **Device status** - e.g. ON, OFF (VT); Energy Star has “Off Mode”, “Sleep Mode”, and “Idle Mode”.

The basic power states for an electronic device are on, sleep, or off (IEEE 1621). For other devices, active, ready, and off are common. There are standards which specify more fine-

[32] This scheme mixes power state and operational states in an awkward way.

grained states, though it is important to distinguish power states from functional states. For example, a washing machine has rinse, spin, and other cycles, but these functional states are all part of a single power state (active).

For power state, EMAN provides a flexible mechanism for reporting the power state among multiple standard series of states, the first of which is On/Sleep/Off.

For functional states: **mode** and **present-value** (BACnet; 160, 85); **Device Status** (i.e. ok, warning, alarm) (FSGIM); **hisStatus, curStatus** - historical and current (ok; fault; down; disabled; unknown; pending; syncing) (Haystack).

Recommendation

Many power or functional state listings are specific to particular types of devices. For broad use, there are three basic states used for electronics, and for appliances and other devices, off, ready, and active are most applicable.

- **PowerState** – enumeration – IEEE 1621 states (on, sleep, off) plus ready and active (and of course, missing)

Energy Reporting

The core measures for Energy Reporting are power and energy; these are covered in this section. Voltage and current are considered in the sensor section of this report, as are many data about power and energy.

The current power level of a device is in theory an instantaneous snapshot, but in practice is typically an average over a period of time, even if that time is short (e.g. under a second). For AC power, average power needs to be calculated over at least one AC cycle to be useful. Regardless, the distinction is rarely if ever important for ER purposes. A power snapshot is different from the average power level over a time period. The current power level is necessarily associated with a specific time.

Fields named are **CurrentPower** (CTA 2047), **PowerMeasurement** (EMAN), **Input power** (ECMA), **power** (NILM, under Measurements / physical_quantity), and **Measured Power** (Energy Star). All of these use watts as the unit of measurement, though EMAN uses kWh. Some standards provide for prefixed quantities such as "MWh".

BACnet uses **present-value** (85) for power and energy data, in conjunction with a timestamp. Haystackuses **power** (in kW for "electric meters") or **curVal** (current real-time value of a point). Volttron references OpenEIS (now Volttron Central), and uses **power** such as "watt" ("W"), "kilowatt" ("kW").

The core of energy reporting is the type of data provided by a utility meter—accumulated energy use along with a time stamp. If a series of meter readings are available, energy use over time durations can be calculated by subtracting adjacent readings. It is then trivial to calculate the average power level over each period, as power is energy over time, and both are known. Conventionally, utility meters were read on monthly or similar intervals, but with the advent of communicating ("smart") meters, the granularity of the data can be greatly increased. When devices are queried over a network, the time period between readings could easily be anything from seconds to years, based on the building manager's needs. Electricity energy values would be most commonly reported as kWh or Wh.

Energy is power over time, so needs to be accompanied by a time stamp (often the current time) and either a beginning time or time period (and each of these can be calculated from the other). There are two primary approaches to reporting energy that derive from this difference. Reporting or storing the incremental energy used since the previous report, or the total energy use since measurement began. The latter has an advantage in that if any report "goes missing", then no energy use is lost; only the time resolution is reduced.

BACnet has the **present-value** (85), which could present power, energy, or a device's status in the system (e.g. "TRUE", or "2" representing "ON"). Project Haystack has an **energy** tag with unit of kWh for electric meters in the system. ENERGY STAR has fields that represent a device's energy consumption such as "TEC (Typical Electricity Consumption)", measured in kWh per year based on established test procedures.

Mechanisms which use the cumulative consumption approach include **TotalEnergy** (CTA 2047, in Wh), **energy** and **cumulative energy** (NILM in kWh); and **EnergyMeasurement** (EMAN, oddly kWh as its base unit, even though it includes a scaling ability).

Volttron references OpenEIS (now Volttron Central), and includes measures of **energy** as "kilowatt_hour" ("kWh"), and "watt_hour" ("Wh").

EMAN and several other standards allow a measurement to be scaled by a factor of 10 multiplier up or down by 10^{24}.

CTA 2047 enables an end-use device to track energy use over defined periods of time and report out time series data. CTA 2047 also can report the time since a device was last turned on and since last turned off. This is particularly useful for (and the reason for including in CTA 2047) for devices that can allow or cut power flowing to a second device, but which don't measure the energy flowing to the second device.

Recommendation

Electrical power should be expressed in W (including fractions thereof if applicable). Electrical energy should be expressed in Wh. While energy can be the total for a specified period, often it is advantageous for data to be expressed as a meter reading. Use:

- **PowerLevel** – float – current electrical power in W
- **CumulativeEnergy** – float – accumulated energy use in Wh

Static Power Data

Static refers to the power properties and accuracy reported by the device that are set at time of manufacture and do not change. Frequency is not expected to vary and is always reported in Hz: **freq** (Haystack). Voltage in the static sense is the nominal voltage at hand, not a measurement of the actual voltage occurring at a particular point in time. "Volt" in Haystack refers to the measured voltage for an electrical meter.

Static data about electrical specifications includes a field for whether the device uses AC or DC, and for AC, the frequency and number of phases. Also needed is the nominal voltage range, and an indication of accuracy. Since accuracy can be expressed in many ways, the format of the accuracy is presently left as an open-ended text field. NILM specifies **Alternative Current (AC) Type** to be "one of {'reactive', 'active', 'apparent'}".

sMAP includes **Min value**, **Max value**, **Resolution**, **Precision**, **Accuracy**, and **Sampling period**.

DMTF and EMAN both reference **Accuracy** and both express it in units of hundredths of a percent.

Accuracy is a quantification of how accurate the measurement is guaranteed to be, either as an absolute value, or a percentage.

Recommendation

For fields to use we propose **CurrentACDC**, **ACPhases**, **ACFrequency**, **NominalVoltageMinimum**, **NominalVoltageMaximum**, **PowerAccuracy**. However, as this topic is not central to dynamic building operation, we have left it off of Table 2.

Summary of Data Model Needs

Table 3 lists the data elements we have identified as most important to standardize.

Table 3. Summary of Data Model Needs

Item	Data Type	Comment
Sensors		
ItemType	enumeration	One of: Sensor, Enumeration, Text, Float.
Units	Text	UCUM or IEEE 1451
Identification, Unique		
UUID	uuid	128 bits (16 bytes)
LocalIdentity	Text	list of “keyword=value;”
Identification, General		
EntityManufacturer	Text	name of Manufacturer, generally without suffix (e.g. Inc.)
EntityBrand	Text	name of Brand if different from manufacturer, otherwise empty
EntityModel	Text	model number/name
EntityIdentityGeneral	Text	list of “keyword=value;”
EntityURL	Text	
Classification		
DeviceType	Enumeration (0..92)	Universal Device Classification, B. Nordman and H.Y. Cheung, 2013.
Local Data		
LocalName	Text	Locally-determined name
LocalOtherInfo	Text	list of “keyword=value;”
Location		
LocationLocal	Text	list of “keyword=value;”
Power State		
PowerState	Enumeration (0..5)	
Timestamp		
TimeStamp	Float or text	Unix time or RFC 3339 time
Energy Reporting		
PowerLevel	Float	current electrical power in W
CumulativeEnergy	Float	accumulated energy use in Wh

4. Future Review Work Needed

We reviewed over 20 existing standards that are relevant to building interoperability, including several existing standards. We then summarized the findings and produced recommendations according to these topics.

Earlier, we identified ten topics that need attention to ensure successful standardization and interoperability (see Table 1).

This review showed that categories such as General and Unique Identification are found in many schemas and represented in very different ways. On the other hand, there are fewer variations in the Power, Energy, and Static Power categories, and standardization would be more straightforward. Finally, in the reviewed schemas, Power State is either omitted, or represented as device operational status (e.g. ok, fault, warning), rather than in the form that is relevant to power and energy reporting - e.g. on, off, and standby.

Future data model review needs

In summary, the data model review strengthened LBNL's understanding on how energy reporting fields are addressed in existing standards. Although we examined more than 20 standards here, the reviewed standards are only a portion of the existing standards relevant to these topics. Further research should include the review of additional standards to further understand data schema needs and opportunities. A list of possible standards for review is listed in Table 4.

Table 4. Promising sources for future data model review (no order implied)

• LONtalk • EU Ontology research • Modelica • EnergyPlus • Industry Foundation Classes (IFCs) • gbXML • COBIE • IOT-DB • MESA • ISO/IEC 18012 • OData • WirelessHART • W3C	• The Common Industrial Protocol (CIP) • KNX (building control protocol widely used in Europe) • LonWorks (local operating network) • OpenADR (a protocol for Automated Demand Response) • OpenGeospatial • CTA 709 • Echonet • Niagara Framework • *Standards other that already reviewed:* – Universal Plug and Play (UPnP) – ZigBee

Beyond addressing other standards, other topics also merit attention, as follows:

Power Interfaces

A facility only present in one standard is Power Interfaces (in EMAN), and even there it is not fully developed. A description of the purpose and implementation of Power Interfaces can be found in draft-nordman-eman-er-framework-02 (Nordman, 2013), and is based on draft-quittek-eman-reference-model-03 (Quittek and Nordman, 2011).

Power Interface (PI) is an interface on a device through which power can flow into a device (an inlet) or out of it (an outlet). Some PIs change over time from being an inlet to being an outlet and vice versa, however most PIs never change. Most devices have a single inlet. Devices with multiple inlets often have them connected to separate power distribution trees. Most devices have no outlets, but those that do often have many. The only distinction between an inlet and outlet is the sign of the power value: positive for an inlet and negative for an outlet. A PI can indicate whether it is capable of being only an inlet, only an outlet, or can switch between the two. PIs are part of a device; a PI is never within a component, and a PI cannot contain anything within it. A PI consumes no power itself. It is always on the border of a device, never internal.

The flow of electricity within a building is determined by how power interfaces are connected to each other — the wiring topology. Thus, a key PI attribute is a list of the other interfaces to which the PI is connected. The power interface term is not new; the Power over Ethernet (PoE) standards describe a power interface as the interface between a device and the Ethernet transmission medium.

Location

The solution for Location described in this report is weak. A better solution should be found, or developed.

Advanced Energy Reporting Data

The LBNL report on Energy Reporting LBNL report (Nordman 2013) also identified data needs for advanced energy reporting categories. EMAN covers all of these, to some degree. Examples include ways to address and report 3-phase power, and battery as a special type of component in a device. Identifying solutions for these topics is beyond the scope of this report, but should be further investigated in future research. Below are the advanced categories.

- Identification, Advanced
- Power, Advanced
- Battery
- Energy, Advanced
- Time-series data
- Reporting, advanced
- Proxy control

OUIs

An OUI is an Organizationally Unique Identifier. It exists to unambiguously identify a company as the producer of a product, as part of identifying a particular device. There are several forms of these, most of which are standardized and registered by IEEE. OUIs could serve a useful purpose in General and Unique Identification as in this report, and so deserve future investigation for their merits and problems. There are also issues with their length and standard representation.

Classification

Device classification is covered in this report, and while the enumeration we recommend seems satisfactory, it needs to be adopted by a suitable organization set up to maintain it. IANA is an obvious example, but not the only one. Effort should be directed towards making this happen.

Standard URL

This report recommends that devices report a standard URL as part of a device's standard reporting. This would be a web page maintained by the manufacturer with diverse information about the device's capabilities, energy test procedure results, and other data. It should have both a human readable form, and one designed to be machine readable. There should be a process to create a standard for what types of data could go on such a page and how to format it. This would be useful for many purposes.

Acknowledgments

We would like to thank the many people that contributed to this, including Rich Brown, Stephen Czarnecki, Steven Lanzisera, Jessica Granderson, Mary Ann Piette, Janie Page, Marina Sofos, Jereme Haack, Gabe Fierro, Arka Bhattacharya, Steve Ray, Ken Wacks, Bill Rose, Stephen Dawson-Haggerty, William Miller, Wouter van der Beek, Norm Bourassa, and Alan Meier.

References

ASHRAE, BACnet: A Data Communication Protocol for Building Automation and Control Networks, ANSI/ASHRAE Standard 135-2004, 2004.

ASHRAE, Facility Smart Grid Information Model, BSR/ASHRAE/NEMA Standard 201P, Review Draft, 2012.

Bierman, A., D. Romascanu, and K.C. Norseth, RFC 3433 - Entity Sensor Management Information Base, IETF, 2002.

Consumer Electronics Association, CTA 2047 - Consumer Electronics - Energy Usage Information (CE-EUI), 2014.

Ecma International, Smart Data Centre Resource Monitoring and Control, ECMA-400, 2nd Edition, June 2013. www.ecma-international.org/publications/files/ECMA-ST/ECMA-400.pdf

Kelly, Jack, and William Knottenbelt, Metadata for Energy Disaggregation, 2nd IEEE International Workshop on Consumer Devices and Systems (CDS 2014) 2014.

IETF, RFC 4589: Location Types Registry, H. Schulzrinne, H. Tschofenig, July 2006.

Modbus, Modbus Application Protocol Specification V1.1b3, 2012.

Nordman, Bruce, Basic Device Classification, report to the Northwest Energy Efficiency Alliance, draft, 2014.

Nordman, Bruce and Iris Cheung, Basic Device Classification, report to the Northwest Energy Efficiency Alliance, draft, 2014.

Nordman, Bruce, Energy Reporting Framework, draft-nordman-eman-er-framework-02, October 21, 2013, http://datatracker.ietf.org/doc/draft-nordman-eman-er-framework/

Nordman, Bruce, Anna Liao, Iris Cheung, Stephen Czarnecki, Steven Lanzisera, Jessica Granderson and Mary Ann Piette, End of FY2014 Draft Report on Self-Measuring and Self-Reporting Sensors and Systems, prepared for DOE Building Technologies Office, September 30, 2014.

Quittek, J., Nordman, B., and R. Winter, "Reference Model for Energy Management", draft-quittek-eman-reference-model-03 (expired), October 2011.

Quittek, J., Chandramouli, M. Winter, R., Dietz, T., Claise, B., and M. Chandramouli, "Requirements for Energy Management ", RFC 6988, September 2013.

Quittek, J., Winter, R., and T. Dietz, "Definition of Managed Objects for Battery Monitoring", RFC 7577, July 2015.

Quittek, J., Chandramouli, M., Winter, R., Dietz, T., and B. Claise, "Requirements for Energy Management", RFC 6988, September 2013.

Parello, J., Claise, B., Schoening, B., and J. Quittek, "Energy Management Framework", RFC 7326, September 2014.

Schoening, B., Chandramouli, M., and B. Nordman, "Energy Management (EMAN) Applicability Statement", RFC 7603, August 2015.

P. Leach, M. Mealling, R. Salz, RFC 4122, A Universally Unique IDentifier (UUID) URN Namespace, Internet Engineering Task Force, July 2005.

Weng, T., Nwokafor, A., and Agarwal, Y. An Integrated Management System for Building Analysis and Control, BuildSys 2013, November, 2013.