

Sustainability Data and Analytics in Cloud-Based M2M Systems

Hong-Linh Truong and Schahram Dustdar

Abstract Recently, cloud computing technologies have been employed for large-scale machine-to-machine (M2M) systems, as they could potentially offer better solutions for managing monitoring data of IoTs (Internet of Things) and supporting rich sets of IoT analytics applications for different stakeholders. However, there exist complex relationships between monitored objects, monitoring data, analytics features, and stakeholders that require us to develop efficient ways to handle these complex relationships to support different business and data analytics processes in large-scale M2M systems. In this chapter, we analyze potential stakeholders and their complex relationships to data and analytics applications in M2M systems for sustainability governance. Based on that we present techniques for supporting M2M data and process integration, including linking and managing monitored objects, sustainability monitoring data and analytics applications, for different stakeholders who are interested in dealing with large-scale monitoring data in M2M environments. We present a cloud-based data analytics system for sustainability governance that includes a Platform-as-a-Service and an analytics framework. We also illustrate our prototype based on a real-world cloud system for facility monitoring and analytics.

1 Introduction

Consider complex buildings with thousands of monitoring sensors that monitor hundreds of objects. In such buildings, three main types of data will be collected and integrated: (1) data about building elements (e.g., floors, equipment, and electricity systems), (2) data about sensor configuration and status (e.g., sensor configuration

H.-L. Truong (✉) · S. Dustdar
Distributed Systems Group, Vienna University of Technology, Vienna, Austria
e-mail: truong@dsg.tuwien.ac.at

S. Dustdar
e-mail: dustdar@dsg.tuwien.ac.at

parameters, sensor location and sensor data type), and (3) monitoring and analysis data [e.g., monitoring status of building elements, energy consumption data, and Greenhouse Gas (GHG) data]. These rich types of data have different lifetime, constraints, and usefulness and have different relevancy to different stakeholders. These types of data play a crucial role in understanding, realizing and optimizing business opportunities centering around IoTs (Internet of Things) in smart environments. Therefore, to foster various data analytics for multiple stakeholders, we need to have efficient ways to manage not only data themselves but also complex relationships among them and analytics applications and stakeholders.

Our work is focused on supporting different types of analytics for understanding complex sustainability measurements (e.g., electricity consumption and GHG calculation) and maintaining M2M environments [e.g., monitoring failure of chillers and optimizing the operation of HVAC (heating, ventilation, and air conditioning) systems]. Such data analytics are crucial for different operation and maintenance processes in sustainability governance of buildings that are required by different stakeholders [1], such as building operators, equipment manufacturers, and auditors. Research effort so far has been concentrated on techniques for low level data management, such as sensor integration, data storage, and data query mechanisms, but has neglected the complexity and diversity of stakeholders and their interests to business and data analytics processes. Therefore, sustainability data and corresponding analytics applications are not well managed to support multiple stakeholders. For supporting data analytics required by multiple stakeholders, we cannot simply focus on single types of data but we have to make use of the interconnected relationships among different types of data monitored and gathered from many types of objects in the same environment. This requires us to also address the linked data problems in M2M environments as well as diversity of analytic forms.

In this chapter, we analyze stakeholders and their requirements for data analytics in cloud-based M2M environments with a focus on solutions for Platform-as-a-Service (PaaS) for integrating IoT data and processes in sustainability governance using cloud technologies. Our goal is to develop techniques for linking monitored objects, monitoring data, and applications to stakeholder's needs and to provide a cloud-based data analytics system with a Platform-as-a-Service (PaaS) and supporting tools for stakeholders to access data and perform data analytics of their own interest. This chapter contributes (1) a detailed analysis of stakeholders in cloud-based sustainability governance in smart buildings and their requirements for data analytics, (2) techniques for linking, enriching and managing data for sustainability analysis in M2M environments, and (3) a data analytics system including a PaaS and an analytics framework for managing relationships between stakeholders, monitored objects, monitoring data and applications. Our techniques can be adopted by cloud-based M2M platform providers. To illustrate the usefulness of our work, we present our prototype atop a real-world cloud system for facility monitoring and analytics.

This chapter is an extension of [2]. We have substantially revised and extended stakeholders and requirements analysis, data linked models and platform design as well as introduced new techniques to manage complex data for different types

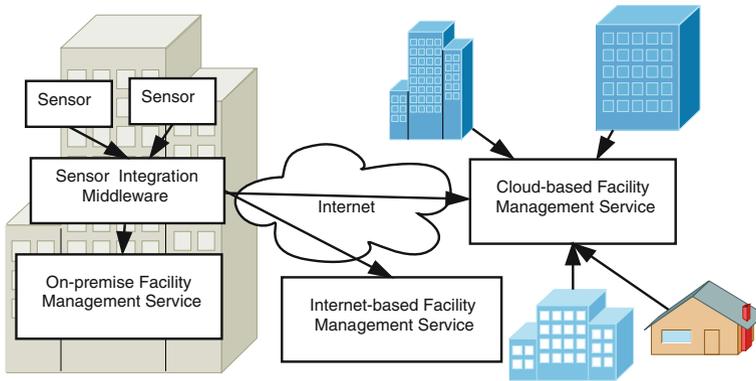


Fig. 1 Typical model of building's IoT monitoring

of analytics and provided detailed techniques inside our system. The rest of this chapter is organized as follows: Sect. 2 presents background, motivation and related work. In Sect. 3 we analyze stakeholders and their requirements for data analytics in cloud-based sustainability governance platforms. We present techniques for enriching, linking and managing M2M data in sustainability governance in Sect. 4. Section 5 describes our cloud-based data analytics system for sustainability governance. Section 6 presents our prototype and experiments. We conclude the chapter and outline our future work in Sect. 7.

2 Background, Motivation, and Related Work

2.1 Background

In an end-to-end M2M system several *monitoring sensors* will perform the monitoring and measurement of facility elements and their surrounding environments, such as equipment, air quality, and meter usages. These elements and environments are *monitored objects* whose status and operations will be monitored, analyzed and controlled to meet stakeholder's business requirements. Depending on different situations and configurations, data captured from sensors can be considered significant for being aggregated/pre-processed in *Sensor Integration Middleware* (also called *M2M gateway*). The significant data is then propagated to storage, analysis, and monitoring services which can be hosted on premise or on cloud-based infrastructure.

Figure 1 describes typical elements in such above-mentioned systems. Depending on systems, data analytics can be performed on premise, over the Internet with or without cloud computing systems. Today, several frameworks and middleware have been provided to support capturing and storing monitoring data and to perform data analytics on-premise or over the cloud [3]. Examples of systems that support

on-premise monitoring and analysis are for homes [4], for the integration of different monitoring sensors to provide data for buildings, houses and transportation vehicles [5–7], and for relaying monitoring data to consumers [8, 9]. Going beyond on-premise monitoring, Internet-based facility monitoring allows to monitor and analyze buildings through the use of enterprise facility information systems, such as shown in [10, 11]. Recently, several cloud-based platforms to support sustainability monitoring and analysis of facilities, such as the AMEE [12], ECView [13] and the Galaxy platform [14]. Our work in this chapter focuses on the cloud-based platforms supporting monitoring, analysis and control of facilities.

2.2 Motivation

Our work is motivated by complex, diverse and mass-customized needs of data analytics for sustainability governance from different stakeholders that a cloud-based M2M provider must support. First, we need to identify main types of stakeholders and which analytics applications, processes and constraints are associated with these stakeholders. Generally, different stakeholders will need different processes, applications and access controls that handle, utilize and apply to different types of data. Currently, it is not very clear how to manage such complex associations in clouds, given the fact that each stakeholder might need only a part of monitoring data of a given monitored object (e.g., a chiller) as well as a stakeholder might need to access the same type of monitoring data across different facilities based on business needs and contracts [3]. This is very different from (open) e-science data or sensor Web platforms in which usually one type of stakeholders (e.g., scientists) tends to access large-scale datasets of similar data types (e.g., satellite images).

Second, we want to support different types of analytics of sustainability measurements that involve multiple types of data. However, data composition for such analyses remains challenging partially due to the lack of techniques for discovering the right data associated with monitored objects. For example, being able to obtain measurements about indoor air quality (IAQ) and costs would allow us to control and adapt the operation of several devices and systems to make sure that the air quality is suitable for specific contexts. Explained in a study of the Environmental Protection Agency (EPA) in [15], several types of building structure data and monitoring data are needed, e.g., floor area, windows, and HVAC monitoring data. But finding all relevant data for such analyses requires us to manage complex, interdependent relationships between monitoring data, monitored objects, applications and stakeholders. This is different from the low-level data management for monitoring data [16] or analytics tasks managements, such as in [17, 18].

Third, data analytics in M2M-based sustainability governance are usually developed and performed by domain experts, who lack IT skills but require different supporting environments for developing and executing data analytics. Their analytics applications are diverse in terms of program execution models (e.g., sequential application, batch jobs, workflows, Hadoop, and automatic lightweighted applications),

execution environments and programming languages (e.g., R, Java, and Python). From the business perspective, stakeholders also require different forms of support, e.g., programmable data analytics APIs for service integrator or platform/application developer, Software-as-a-Service (SaaS) for the end-user, and automatic lightweight applications for OEM (Original Equipment Manufacturer) and equipment operators. Thus, providing PaaS will also hide IT complexity to enable domain experts to develop and test their different applications for specific monitored objects in their domains, fostering different models for data analytics in M2M-based sustainability governance.

2.3 Related Work

Recently, several cloud-based platforms to support the monitoring of energy consumption have been introduced, such as Tendril,¹ AlertMe [19] and xively [20]. Furthermore, there are systems supporting sustainability governance for buildings, such as the Galaxy platform [14]. While these systems can manage different types of data and provide different applications to analyze the data, most of them let the user to manage the complex relationships between monitoring data and monitored objects and do not provide a generic framework for managing these relationships.

Several stakeholder analyses have been studied for sustainability technologies [21], but we are not aware of stakeholder analysis for sustainability governance of facilities in clouds. In [22] the authors discuss about a cloud system for ubiquitous cities. However, stakeholders and application and data controls have not been discussed.

Generic data management techniques for sensor data, such as [23], discuss general security, privacy, and provenance, but not the complex relationships between data, application and stakeholders. Investigation of cloud computing for storing and processing sensor data has been conducted recently, such as techniques to access sensor data stored in their cloud using HBase [24] and using NoSQL [25]. However, they do not deal with complex relationships between monitoring data and sustainability governance features for facilities.

Several general cloud services have been developed, such as CA AppLogic [26], Appistry [27], Google App Engine [28], Microsoft Azure [29] and Amazon DynamoDB [30], and Parabon Frontier [31]. They offer APIs for accessing data but do not provide support for specifying and managing complex relationships between monitoring data and monitored objects. Thus when using them for storing monitoring data, we need to develop models for managing relationships among data, monitored objects, stakeholders and applications. Several works have been focused on data analytics in cloud and grid for e-science data [32] but they do not address facility monitoring data.

¹ <http://www.tendrilinc.com>

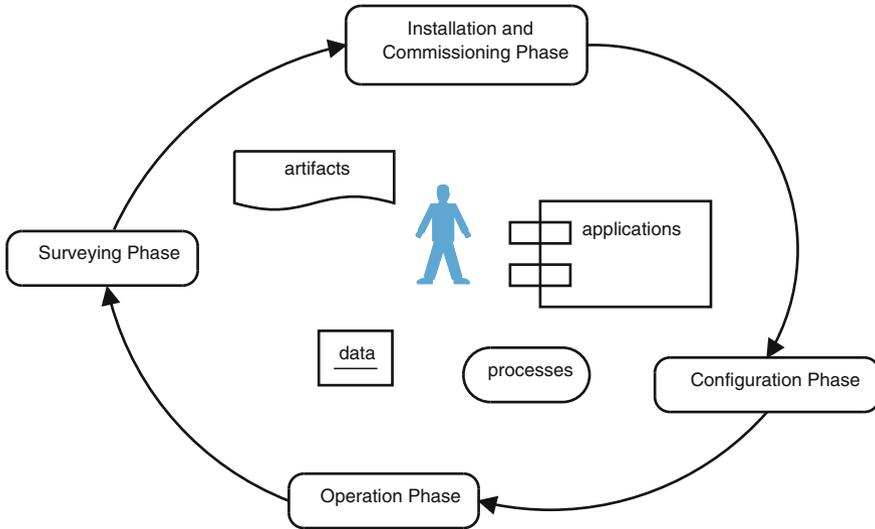


Fig. 2 Phases in establishing and operating facility monitoring and analysis

3 Stakeholders and Requirements of M2M Data Analytics

3.1 M2M-based Sustainability Governance

Several phases exist in the establishment and operation of sustainability monitoring and analysis of a facility, as shown in Fig. 2. In *Surveying Phase*, *Installation and Commissioning Phase* and *Configuration Phase*, few stakeholders are involved, such as the cloud facility management provider and facility owner. However, in the operation phase, several stakeholders involved in different tasks require different combining sets of data, artifacts, applications and processes [3]. Overall, these tasks center around the following types of *monitored objects*:

- *Facilities*: including facility elements, such as buildings, floors, and rooms,
- *Facility equipment and systems*: including equipment and systems used to operate facilities, e.g., electricity systems, freezers, compressors, chillers and fans, and
- *IT monitoring systems*: including sensors, gateways, networks and platforms used to gather monitoring data about facilities, and facility equipment and systems.

There are diverse types of and large amount of monitoring data collected for these monitored objects. Such data can be processed and manipulated by various types of *governance features* implemented as *applications* in a cloud-based sustainability governance platform, as described in the following:

- Near real-time *MonitoringApp*: are used for online monitoring and controlling facilities, such as alarm monitoring and near-real time energy consumption, thus

handling monitoring data on the fly and possibly controlling monitored objects at runtime.

- Offline *AnalysisApp*: are used for analyzing sustainability measurements collected via a period of time, such as monthly energy consumption and GHG, for facilities. Typically, this kind of applications will analyze vast data available in the platform. These applications can be utilized by near real-time *MonitoringApp* in order to decide suitable actions at runtime.

Both *MonitoringApp* and *AnalysisApp* can carry out their functionality centralized, e.g., having all their tasks executed in a single server, or distributed, e.g., having multiple monitoring and analysis tasks executed in the cloud servers and M2M gateways. They might implement a simple or a complex process of several tasks. Sustainability governance features used by stakeholders will handle monitoring data under *data constraints*. Such constraints specify different conditions about, e.g., which types of monitoring data, how many data streams, and how many monitored objects. Typically, conditions are associated with the business models agreed between stakeholders and the provider of the cloud-based sustainability governance platform.

3.2 Stakeholders and Sustainability Data

Sustainability governance is not only in the interest of stakeholders that own and/or operate facilities but also other stakeholders who can benefit from data sharing and analytics functions of sustainability governance platforms, such as regulators, community users and developers (see [3] for possible stakeholders). Therefore, understanding stakeholders will enable the modeling and management of complex relationships between stakeholders, monitored objects, monitoring data and applications. For a given type of stakeholders, different roles can exist. For example, the equipment manufacturer and the maintainer can utilize data to perform (1) individual equipment maintenance, (2) electricity system maintenance, or (3) mechanical system maintenance in different ways.

To illustrate stakeholders in sustainability governance, let us consider 5 facilities in Dubai shown in Table 1. Overall, there exist several stakeholders, roles and types of data. Even with only mechanical, electrical, and plumbing (MEP) systems, in these buildings, we have around 60 types of monitored objects and each monitored object has several data streams indicating types of monitoring data. From our analysis of stakeholders and existing platforms in [3], we see the need to enable the customization of sustainability governance feature provisioning and to recommend suitable data streams and applications based on complex relationships between stakeholders, monitored objects, monitoring data and applications. With respect to service provisioning and customization, for a large-scale facility, there could be various stakeholders interested in the same monitored object, but different types of monitoring data, as well as there could be a stakeholder who allows to access to multiple monitored objects. This requires a fine-grained management of applications,

Table 1 Example of facilities and their data streams and stakeholders

Name	Data streams	Nr. stakeholders	Nr. roles	Nr. users
JAFZA	2,656	4	9	15
CARRIER AL FUTTAIM MOSQUE	66	4	9	9
CARRIER BIN HINDI TOWER	98	4	9	10
GREEN BUILDING	657	4	9	17
G2 INTERNATIONAL	155	4	9	15

monitored objects, and monitoring data for each stakeholder. A real-world sustainability governance platform in the cloud will probably serve for hundred thousands of buildings. The number of data streams (for monitoring objects) in each building is large and applications are diverse, making the management of their relationships complex.

3.3 Characteristics of Sustainability Monitoring Data and Analytics Applications

Sustainability monitoring data typically are associated with multiple types of monitored objects which affect each other. A particular interesting characteristic is that these types of data can be monitored and analyzed in isolation or in combination. For example, monitoring data of a chiller can be analyzed alone by the manufacturer or the maintenance operator to understand the operation of the chiller, while analyzed in combined with facility building sensing temperature and outdoor temperature in order to understand how to tune the indoor air quality and comfort. Generally, we cannot assume that different types of monitoring data will always be analyzed together because a specific stakeholder (e.g., chiller manufacturer) and applications for the stakeholder require only specific types of data (e.g., only chiller monitoring data). This requirement is associated with business models (e.g., payment) and compliance issues. Thus, the access to data must be controlled.

Due to very large types of monitoring data and stakeholders, diverse types of analytics exist. First, different monitored objects require different monitoring and analysis applications, and, even for the same category of monitored objects (e.g., chillers) each manufacturer can have a different analysis model for his/her type of objects. Second, diverse types of users exist, including normal users using SaaS and scientists/developers/domain experts writing complex sustainability monitoring and analysis applications/jobs. Third, diverse types of execution platforms exist due to the diversity of experts and analysis models for sustainability governance. Therefore, although different types of monitoring data can be managed by a single platform, analysis applications will not follow a single model. They will be developed

using different languages (e.g., Java, R, Matlab and Python), and execution models (e.g., sequential program and workflows), depending stakeholders and their analytics requirements.

4 M2M Data and Process Integration for Data Analytics

4.1 M2M Data Fragments

One of the most important issues in managing a large number of sensors is how to link them to monitored objects. In many systems, this link is identified only in the deployment and configuration phase, in which sensor identifier is mapped to monitored object identifiers, by the people who install, configure, and manage the sensors. However, in complex facilities, monitored objects are not atomic, they can include other monitored objects, or linked together to provide a virtual monitored object. Furthermore, links from sensor monitoring to monitored objects are not established in a single process but multiple ones, involving multiple stakeholders. This requires us to manage the change of linking data over time.

In our study, several of data collection processes, except in the Operation Phase, are carried out by different teams and currently implemented with software support and manual paper sheets without integration, leading to the lack of links between monitoring data to monitored objects and dependencies among different types of data. To link monitoring data with monitored object information to support data analytics, we consider the following situations:

- **Situation 1**—rich monitored object information is available in a well-defined specification (e.g., newly design building) and can be supplied by the facility owner. In this case, it is possible to develop a correlation between monitoring data and monitored object information. For example, in most of the above-mentioned phases, we just need to capture information about sensors and map sensors to monitored objects using their identifiers.
- **Situation 2**—facility information is not available in a well-defined specification (e.g., old building), facility information is not completed, and/or facility information can only be provided by the end-user at a high level (e.g., in cases of home owners). In this case, it is possible to enrich monitoring data using user-specific facility information, but, in addition to information about sensors, the above-mentioned processes must also capture monitored object structures and dependencies.
- **Situation 3**—facility information is not available (e.g., the facility owner does not want to reveal the facility structure, yet still wants to enjoy complex analysis) but it is possible to annotate sensor data with certain metadata about monitored objects and locations. In this case, enriching monitoring data is performed at the sensor side.

Table 2 Main elements required for sustainability data analytics

Type	Concept	Description
Monitored objects in large-scale facilities	Building	Describe the whole building
	Floor	Represent a floor in a building
	Component	Represent various types of components of a system
	System	Represent different MEP systems inherent in buildings
Information about sensors and monitoring data	Monitoring sensor	Describes sensors used to monitor objects
	Monitoring data type	Describes the type of monitoring data provided by monitoring sensors
	Data stream	Describes a time series of monitoring data entries
	Data item	Describes single monitoring data entries

- *Situation 4*—facility information is not available and sensor monitoring data cannot be annotated with information about facility elements. In this case, automatic correlation techniques could be possible solutions.

In our work, we link monitoring data and sensor management using an external information service. Building information will be provided by the building owner by uploading their building information based on existing specifications. This way is suitable for *Situation 1*. When the building information is not well-defined and user-specific, then the building information service can provide interfaces for the user to specify his/her building information. In this case, suitable for *Situation 2*, standards or specific building information can be used. In overall, building information and monitored objects will be mapped to monitoring sensors using identifiers via integrated processes carried out through different phases.

4.2 *Linked Data Model*

Several specifications have been developed, such as the IFC classes [33], to cover building structure data. From the concept of existing specifications, main data concepts that can be used to enrich monitoring data for analytics are physical containment objects (e.g., building and building floor) and physical MEP (e.g., equipment and components). On the other hand, from the facility monitoring, we have several types of metadata (e.g., sensor configuration and description) besides a large amount of near-realtime monitoring data. Table 2 explains possible elements from building structure and monitoring data that should be linked.

To manage complex relationships among data, we have developed a linked data model, shown in Fig. 3. With this model, before the Operation Phase, most monitored objects and their dependencies, such as *Building*, *System*, *Component*, and *Floor*, and most entities related to sensors and their configurations, such as *MonitoringSensor*, *SensorConfiguration*, and *SensorModel*, are collected. Especially, *MonitoringSensor* will be used to link monitored objects to monitoring data types. Information about sensors and monitored objects can be obtained from the repository of sensors and monitored objects provided to multiple stakeholders by cloud facility management providers. During the Operation Phase, monitoring sensors will be executed, thus data items in sensor data streams will be stored separately from other types of data but they can be linked by using identifiers.

We utilize existing gateways which provide different monitoring data streams. We use data type identifiers, monitored object identifiers as well as data stream identifiers for identifying data. Let *dataURI* be the unique identifier of a type of monitoring data, *dataStreamURI* be the identifier of a data stream in a gateway, *dataTypeURI* be the identifier of data type, and *monitoredObjectURI* be the identifier of a monitored object. Overall, a *dataID* is a combination of *dataStreamURI* and (*dataTypeURI*, *monitoredObjectURI*). Using *dataTypeURI* and *monitoredObjectURI* we will be able to obtain metadata about types of monitoring data and monitored objects, while utilizing *dataStreamURI* we can obtain monitoring data. For example, to indicate the low suction pressure of a chiller, we can use either <http://pcccl/dataStream/stream124> or (<http://pcccl/dataType/LowSuctionPressure>, <http://pcccl/monitoredObject/chiller123>).

4.3 Mixed Data Management

Since we have multiple types of monitored objects and different types of data and analytics, we do not expect that a single data representation, such as a traditional SQL-based model or a NoSQL-based model, would be suitable for all types of monitored objects. Consider that:

1. monitoring data and facility data are mainly gathered, managed and owned by the M2M service provider
2. each analytics stakeholder utilizing only limited, based on contracts, sets of these data for his/her analytics, and
3. the three types of monitoring data, facility data and analytics results have different structures, scales, volumes and access needs.

We utilize a mixed configuration of relational/graph data models and NoSQL data models to store, manage and link these types of data.

Figure 4 presents the conceptual model of our data management.² The data described by the linked data model in Sect. 4.2 have complex but structured

² Detailed design and implementation of these conceptual models are out of the scope of this chapter.

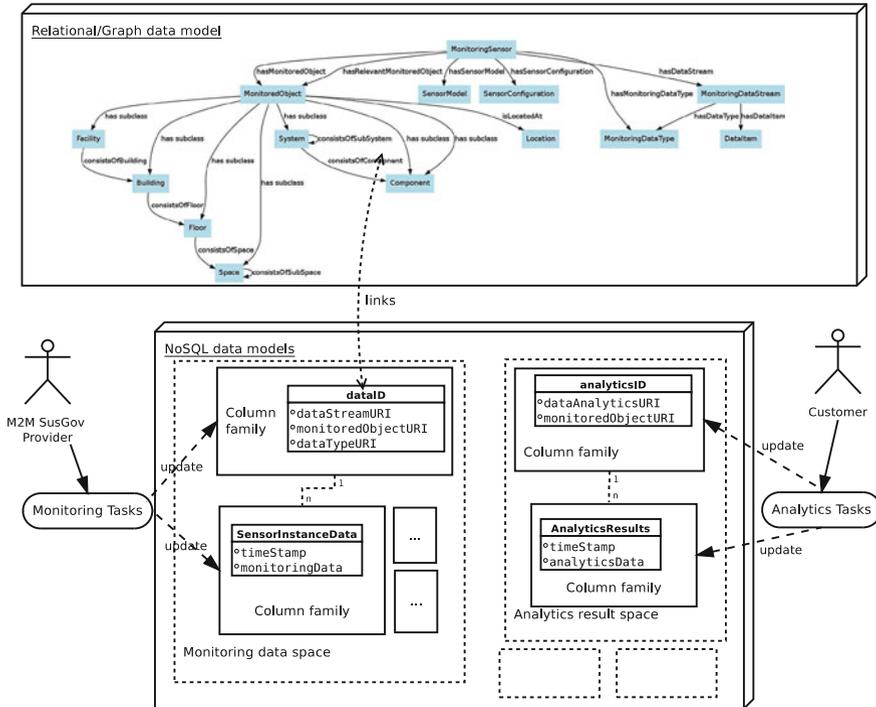


Fig. 4 Combining different types of data models for managing monitoring data, facility data and analytics results

relationships among their elements. Therefore, we utilize a relational/graph data model to manage them. This data model is selected also based on the rationale that the linked data are not changed often and their volume is much smaller than monitoring data which evolve over the time. On the other hand, monitoring data from sensors are voluminous and change rapidly over the time. They are mainly time series data and require elastic, scalable infrastructure to manage. Therefore, we utilize NoSQL data models to manage the monitoring data. The analytics results have different structures, depending on the types of analytics, as well as they are also associated with different unstructured documents about the analytics, e.g., documentation on how the analytics are carried out. Therefore, we also utilize NoSQL models which allow flexible structures of analytics and documents to be managed.

In our model, through *Monitoring Tasks*, the M2M provider manages monitoring data in NoSQL DaaS using different spaces, each is configured for a facility or a group of connected facilities. Within a space for monitoring data, we manage sensor data and link sensor information to facility data using *dataID* column family. Through *Analytics Tasks*, the results of analytics for each stakeholder will be managed in a specific analytics result space for that stakeholder. To support different operations, the NoSQL DaaS will provide different APIs for storing and accessing spaces.

4.4 *Linked Monitoring Data Service*

Based on the linked data model presented in Sect. 4.2, different processes, used to handle parts of the linked data model in different phases, can be integrated. For example, a process consists of a set of UI forms that allow stakeholders to retrieve and manipulate data in a particular phase. Another example is that a process can be a workflow whose activities span multiple phases. All processes will rely on the linked data model and a *Linked Monitoring Data Service* to store and manage information about sensors, building elements, and their links to monitoring data (see Fig. 5). This service is built atop the two different types of data models—Relational/Graph data and NoSQL data models—mentioned in Sect. 4.3. This service supports the following features:

- *Dependency analysis*: this feature analyzes the dependencies among sensors, monitored objects, data. It aims at providing a unified and integrated view of networks of sensors, monitored objects and existing monitoring data streams and this view can be provided in GUI-based tools for the end-user. This feature can be used in different phases.
- *Monitoring data search and composition*: this feature searches relevant monitoring data based on specification of monitored objects and/or specification of sustainability measurements to be analyzed. Depending on whether monitored objects include sub monitored objects or not, all relevant monitored data can be searched and composed into a new data stream to support complex analytics. For example, if an analysis is applied for a room, then based on the room view, relevant monitoring data will be discovered, whereas if an analysis is applied to energy consumption, then relevant data for energy analysis will be searched.
- *Quality of monitoring data analysis*: this feature utilizes linked data in order to determine the quality of monitoring data and the influence of the quality of monitoring data on the detection of faults, context-aware applications, and sustainability analysis. The quality of monitoring data is strongly dependent on, e.g., sensor model and configuration, monitored object structure, and gateway configuration, thus having all of these data linked will foster the evaluation of quality of monitoring data [34].

5 Cloud Services for Sustainability Data Analytics

5.1 Overview

Figure 6 presents core cloud services for sustainability data analytics which integrate stakeholder information, monitored objects, monitoring data and applications. At the core of the system is the *SusGovPaaS* which is a platform-as-a-service for conducting sustainability analytics. *SusGovPaaS* provides features for analytics and management via service APIs. Based on these features, *SusGovSaaS* will offer governance

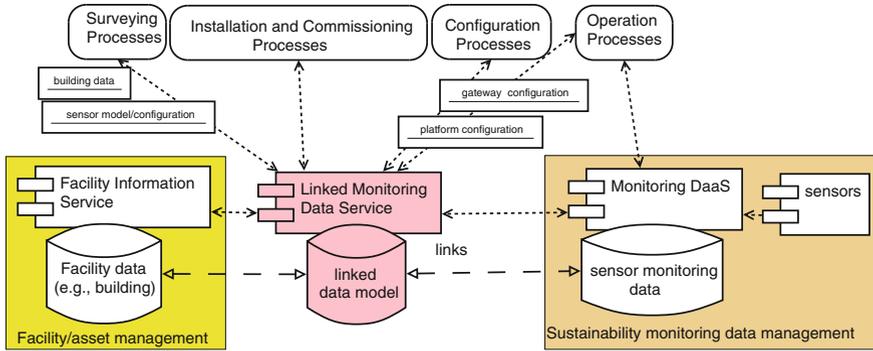


Fig. 5 Integrating processes for managing linked data

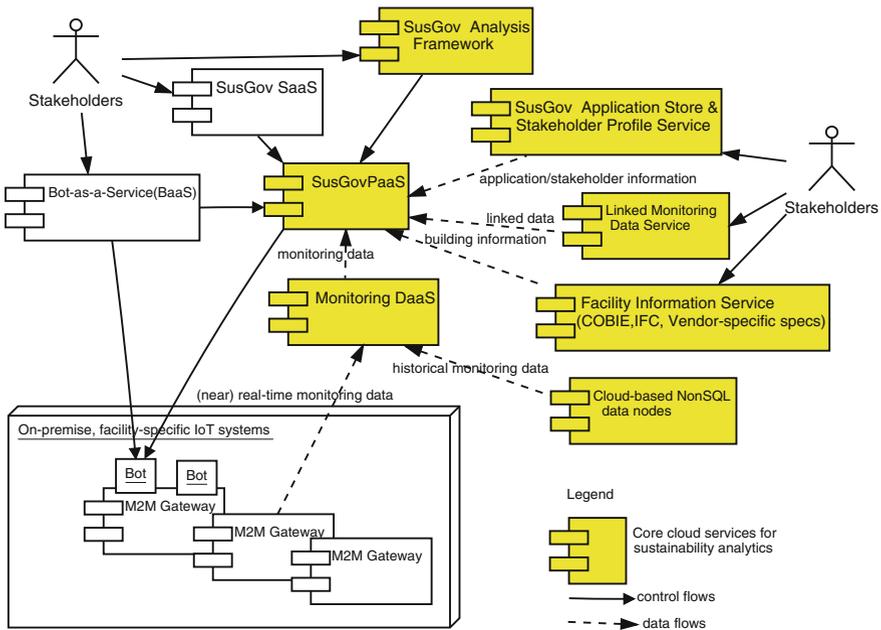


Fig. 6 Conceptual architecture of the cloud-based data analytics system for sustainability governance

analysis for stakeholders using the SaaS model. A *SusGov Analysis Framework* aiming at sustainability analysis expert that exploits *SusGov PaaS* is provided. Furthermore, to enable the development and provisioning of automated, light-weighted monitoring applications (called Bots) [35], *Bot-as-a-Service (BaaS)* which supports the development and management of Bots is also integrated. *BaaS* could deploy its Bots in *SusGov PaaS* and *M2M Gateways* and these Bots can utilize

PaaS Application Discovery API		PaaS Application Execution API		PaaS Data Discovery API		PaaS Data Retrieval API	
Application Management and Execution	Batch Application	Workflow Application	Streaming Application	Bot		DaaS Connectors	
	C/C++ Env.	Python/R/Matlab Env.	Java Env.				
	IaaS		IaaS	Bot Hosting Environment		IaaS	

Fig. 7 Diverse forms of applications and execution environments

SusGovPaaS to carry out certain analytics features in the cloud as well as utilize *M2M Gateways* to perform analytics and monitoring activities in specific gateways.

Stakeholder profiles and applications, managed by *SusGov Application Store & Stakeholder Profile Service* will be utilized by *SusGovSaaS* and *SusGovPaaS* to support service provisioning, customization, and execution. *SusGovPaaS* utilizes stakeholder information and application information to control data access to monitoring data (stored in *MonitoringDaaS*) and monitored object information (stored in *Facility Information Service*). In the following, we discuss some features in these services, in particular in *SusGovPaaS* and *Application Store & Stakeholder Profile Service*.

5.2 *SusGov Platform-as-a-Service*

Figure 7 outlines main building blocks in our sustainability governance analytics PaaS (*SusGovPaaS*) that aims at dealing with dynamic properties of data and applications and supporting analytics for multiple stakeholders. *Monitoring DaaS* is responsible for monitoring data storage and access which can rely on low level data management techniques. The access to data can be performed via data connectors (such as SQL- and REST-based connectors). To enable different types of analytics application models, such as batch, workflow and stream applications and intelligent bots, written in different languages, several execution environments, such as based on Java, R, and Matlab, are provided atop different Infrastructure-as-a-Service (IaaS) and specific bot hosting environments. For bots, we integrate the Bot-as-a-Platform [35]. To expose capabilities of applications and monitoring data for different stakeholders, several APIs are provided for discovering, retrieving and accessing/executing data and applications.

While the execution environment platform is not in the focus of this chapter, we should note that sustainability applications in sustainability governance are very different that in contemporary data analytics platforms, such as in e-science, due to the fact that applications are needed and required for different stakeholders. Applications can be simple sequential programs that examine only a single type of chiller data or can be complex workflows that optimize air quality based on, e.g., temperature, chiller, and electricity consumption information.

5.3 Describing and Managing SusGov Applications

In our work, the types of applications mentioned in Sect. 3.1 are encapsulated in the so-called *sustainability governance applications* (*SusGovApp*). Managing *SusGovApps* is a complex task due to (1) a large number of types and instances of applications exist, and (2) control parameters and input data for applications are complex. Thus, an efficient management of these applications is crucial for complex sustainability governance analysis and service provisioning and configuration recommendation.

We develop a model that can be used to describe *SusGovApps*. Figure 8 presents a simplified conceptual model for *SusGovApps*. Overall, information about a *SusGovApp* will include (1) category of the application (*SusGovAppCategory*), (2) metadata about the application (*SusGovAppDescription*), (3) concrete monitoring data (*MonitoringDataIdentification*), (4) abstract supporting monitoring data types, (*TypeOfMonitoringData*), and (5) output presentation (*OutputPresentation*). Detailed information about execution environments can be specified in application description, whereas with concrete monitoring data types, we can constrain types of data accessed by the application. Many *SusGovApps* can be managed by an application store (*SusGovAppStore*). This model can manage applications which are executed at gateways.

Listing 1 presents a simplified example of the description for an application analysis named *Demand_CP4E* which is used to analyze energy consumption.

Listing 1 Example of application descriptions

```
<owl:NamedIndividual rdf:about="http://pcccl/susgovapp#
  DemandCP4E">
  <rdf:type rdf:resource="http://pcccl/susgovapp#
    SusGovApp"/>
  <rdfs:label xml:lang="en">DemandCP4E</rdfs:label>
  <appCategory rdf:resource="http://pcccl/susgovapp#
    AnalysisApp"/>
  <appDescription rdf:resource="http://pcccl/susgovapp#
    DemandCP4EDescription"/>
  <monitoringData rdf:resource="http://pcccl/susgovapp#
    Stream66273"/>
</owl:NamedIndividual>
```

To implement *SusGovApp*, we consider two possibilities: *SusGovApp* will access data from gateways in buildings and *SusGovApp* access data directly from *MonitoringDaaS* in data centers. In both cases, *SusGovApp* will use certain data connectors, however, we consider these connectors are internal part. In our model, we consider applications as black boxes and these applications can be exposed via a generic interface. For example, instead of defining a detailed logic of a complex analysis application as a workflow and managing the workflow, the workflow can be considered as an input parameter of the application which is implemented as a batch job by invoking a workflow engine to execute the workflow. Depending the implementa-

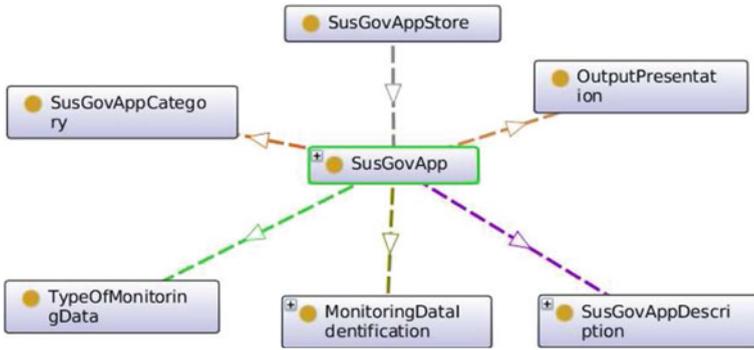


Fig. 8 Description of SusGovApp

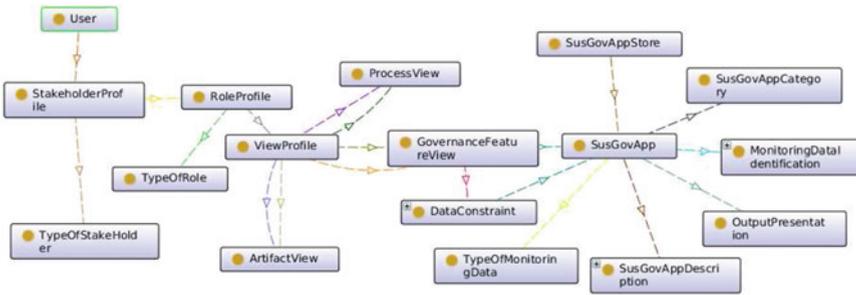


Fig. 9 Linking stakeholders, monitored objects, monitoring data and governance features

tion, the internal logic of a *SusGovApp* can be simple java implementation, complex workflow (for extracting, transforming and loading data) or complex event processing (e.g., for monitoring). In our work, we consider *SusGovApp* can be developed using different methods and frameworks, such as the R framework, Java, Matlab, and complex scientific workflows, due to the diversity and complexity of sustainability measurements. Therefore, a *TypeOfApp* is used to support the identification of possible execution environments.

5.4 Linking Types of Data, Applications and Stakeholders

To link data, applications and stakeholders together, we have developed a conceptual model for managing stakeholders and their views on sustainability governance features and data. Figure 9 describes our model for managing stakeholder in sustainability governance. A stakeholder is described by *StakeholderProfile*. Each stakeholder will be associated with different roles, described by *RoleProfile*. *ViewProfile*, used to specify what a role can be done, will be associated with one

Table 3 Examples of SusGovPaaS APIs

APIs	Description
<code>discover/monitoredobjects/</code>	Discover all monitored objects
<code>discover/monitoredobjects/{monitoredobjecturi}/...</code>	Discover a monitored object based on its uid using <code>monitoredobjecturi</code> and other information
<code>discover/apps/</code>	Discover all available applications based on some metadata
<code>discover/apps/{appid}/...</code>	Discover a specific application indicated by <code>appid</code> and the application's metadata
<code>execute/apps/{appid}/...</code>	Execute applications
<code>discover/data/{monitoredobjecturi}/{uid}</code>	Discover all data related to the monitored object (indicated by <code>monitoredobject</code>) and/or its data indicated by <code>uid</code>
<code>download/{category}/{uid}...</code>	Download monitoring data (<code>category=data</code>), applications (<code>category=apps</code>) and analytics reports (<code>category=reports</code>) by using their <code>uid</code>

or multiple roles. `ViewProfile` will identify which processes should be used for a particular role, using `ProcessView`, and which governance features are allowed, specified by `GovernanceFeatureView`. Governance features will be characterized via possible applications to be used, described by `SusGovApp` and possible constraints on data, described by `DataConstraints`. Both `SusGovApp` and `DataConstraint` are linked to monitoring data they apply, specified by `MonitoringDataIdentification`. The data that the stakeholder can use will be identified via identifiers of monitoring data and data constraints. Each stakeholder can have different permissions for different types of applications and corresponding monitoring data. The above-mentioned model is described in RDF and used to provision sustainability governance analytics features.

5.5 PaaS APIs and Analytics Framework

To foster the utilization and integration of features of SusGov PaaS, we provide different APIs for different consumers to control, execute and manage analytics. Table 3 presents some examples of (simplified) APIs in our PaaS. In general, we provide APIs for (1) discovering monitored objects, analytics and monitoring applications, and data, (2) executing analytics applications, and (3) managing and downloading applications and analytics reports.

To support advanced users to develop analytics applications and conduct sustainability analysis, we develop a *SusGov Analysis Framework* offering rich GUIs and exploiting RESTful APIs for *SusGovPaaS* (e.g., shown in Table 3) to discover and access data and applications as well as to execute applications in our SusGovPaaS.

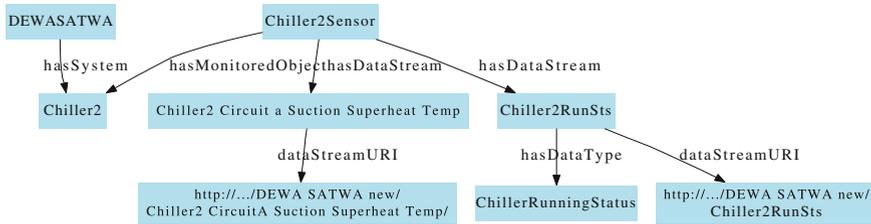


Fig. 10 An exemplified graph for DEWASATWA building and one chiller

6 Illustrating Experiments

In the prototype, (time series) monitoring data are obtained from gateways at the building site; gateways are based on Niagara AX³ as part of the Galaxy platform [14]. The data is in XML but it does not include metadata about monitored objects which are known by engineers who perform the setup and configuration of sensors and gateways. Therefore, *dataTypeURI*, *monitoredObjectURI* and *dataStreamURI* are constructed by extracting metadata about sensors in combination with configuration information. In particular, the configuration information, manually collected via different processes and described in our linked data model, allows the specification of dependencies among different monitored objects. For example, configuration can specify which sensors are associated with which chillers in which buildings. Figure 10 presents an example of a graph describing how monitoring data is linked to sensors and buildings using RDF. Based on dependencies, we could define data constraints for specific stakeholders as well as reason relevant types of monitoring data for specific analyses of monitored objects.

We utilize Jersey—and implementation of JAX-RS for RESTful Web Services—and Weblogic 10.3 for developing our SusGovPaaS. *Application Store & Stakeholder Profile Service* are RESTful Web services that store their information under RDF/XML format and use Allegro Graph.⁴ We use Cassandra as NoSQL data nodes which are controlled and accessed by/via the *MonitoringDaaS* as a REST-based Web service.

Figure 11 shows a snapshot of our *SusGov Analysis Framework*. First of all utilizing monitored objects/data discovery APIs, we can show the relationships among monitored objects that a user can access via a dependency graph (in the top-left window of Fig. 11). This graph allows the user to examine the influence among monitored objects, e.g., which monitored objects can contribute to the analysis of a room. For each monitored object, the user can also see which types of monitoring data he/she can access (in the bottom-left window of Fig. 11). Such access controls are controlled by the cloud providers or stakeholders who have the right to control the monitored object. Similarly, the user can also discover existing applications (in the top-middle window—*Application Discovery*—of Fig. 11). Based on

³ <http://www.niagaraax.com/>

⁴ <http://www.franz.com/agraph/allegrograph>

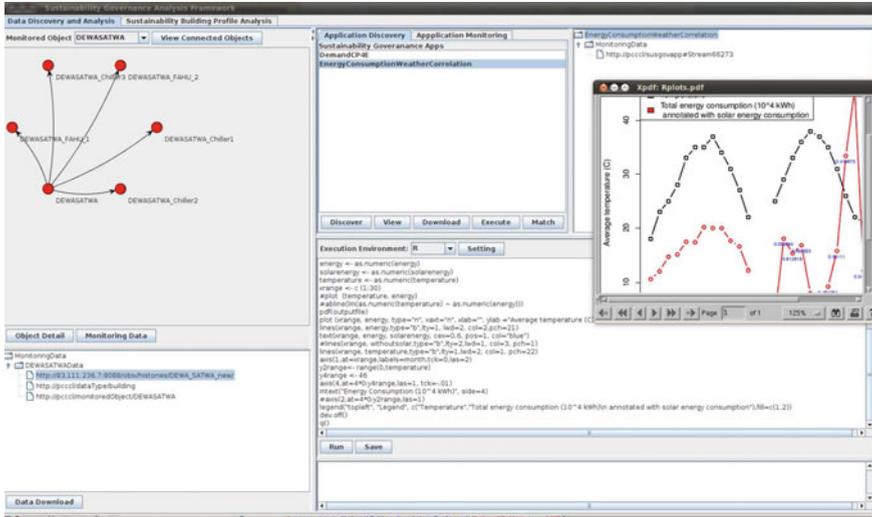


Fig. 11 Example of SusGov analysis framework

application description and monitoring data discovered, the user can select applications and ask our SusGovPaaS to execute them. Monitoring data, applications and reports from analyses can be downloaded back to the user working place, subject to the access control. For example, the chart in Fig. 11 shows an analysis result from a R-based application. By downloading monitoring data and applications, the user can also execute the applications in his/her local execution environment (e.g., shown in the right-middle window—Execution Environment—in Fig. 11).

7 Conclusions and Future Work

In this chapter, we analyze stakeholders, their relationships to sustainability monitoring data, analysis applications and monitored objects. We have developed models for managing these relationships. Based on that we design our SusGovPaaS with APIs for supporting data and application discovery, and analysis application management and execution that can be used by various stakeholders and automated applications. We have illustrated that by linking monitoring data streams, monitored objects, applications and stakeholders, we can manage their complex relationships, thus facilitating complex sustainability governance monitoring and analysis features within cloud-based M2M systems. This also hides complexity by not showing low-level data streams and supports customized features in the SaaS model for sustainability governance. Overall, our techniques aim at supporting cloud-based M2M platform providers to solve complex issues within their cloud systems.

Currently, we test our prototype with monitoring data from the Galaxy platform with a small number of buildings. Thus, we need to move to large-scale testing. We are currently focusing on data analytics for sustainable building profile analysis based on data mining techniques. Furthermore, we are developing recommendation solutions for sustainability governance by analyzing data associated with multiple facilities.

Acknowledgments This chapter is an extended version of [2]. This work is partially funded by the Pacific Control Cloud Computing Lab (PC3L—`pc3l.infosys.tuwien.ac.at`). We thank Manu Ravishankar, Saneesh Kumar, Sulaiman Yousuf and Terry Casey for providing useful information and datasets of facilities. We thank our colleagues in PC3L for fruitful discussion on cloud platforms and analytics.

References

1. Murguzur, A., Truong, H.L., Dustdar, S.: Multi-perspective process variability: A case for smart green buildings (short paper). In: 6th IEEE International Conference on Service-Oriented Computing and Applications (SOCA 2013) (2013)
2. Truong, H.L., Dustdar, S.: M2M platform-as-a-service for sustainability governance. In: IEEE SOCA, pp. 1–4 (2012)
3. Truong, H.L., Dustdar, S.: A survey on cloud-based sustainability governance systems. *Int. J. Web Inf. Syst.* **8**(3), 278–295 (2012)
4. Acker, R., Massoth, M.: Secure ubiquitous house and facility control solution. In: Proceedings of the 5th International Conference on Internet and Web Applications and Services, ICIW '10, Washington, DC, USA, pp. 262–267. IEEE Computer Society (2010)
5. Tompros, S., Mouratidis, N., Draaijer, M., Foglar, A., Hrasnica, H.: Enabling applicability of energy saving applications on the appliances of the home environment. *Netw. Mag. Glob. Internetworkg.* **23**, 8–16 (2009)
6. Krishnamurthy, S., Anson, O., Sapir, L., Glezer, C., Rois, M., Shub, I., Schloeder, K.: Automation of facility management processes using machine-to-machine technologies. In: Proceedings of the 1st International Conference on the Internet of Things, IOT'08, pp. 68–86. Springer, Berlin, Heidelberg (2008)
7. Choi, J., Shin, D., Shin, D.: Research and implementation of the context-aware middleware for controlling home appliances. In: International Conference on Consumer Electronics, ICCE 2005, pp. 161–162. Digest of Technical Papers (2005)
8. Broering, A., Foerster, T., Jirka, S., Priess, C.: Sensor bus: An intermediary layer for linking geosensors and the sensor web. In: Proceedings of the 1st International Conference and Exhibition on Computing for Geospatial Research & #38; Application, COM.Geo '10, New York, pp. 12:1–12:8. ACM (2010)
9. Consortium, B.: Deliverable d2.2: End-to-end platform specification beywatch data model (annex). http://www.beywatch.eu/docs/D22_Annex.pdf. June 2010
10. Granderson, J., Piette, M., Ghatikar, G.: Building energy information systems: User case studies. *Energ. Eff.* **4**, 17–30 (2011). doi:[10.1007/s12053-010-9084-4](https://doi.org/10.1007/s12053-010-9084-4)
11. Swords, B., Coyle, E., Norton, B.: An enterprise energy-information system. *Appl. Energ.* **85**(1), 61–69 (2008)
12. AMEE: <http://www.amee.com/>. Last Accessed 7 Feb 2012
13. Thyagarajan, G., Sarangan, V., Sivasubramaniam, A., Suriyanarayanan, R., Chitra, P.: Managing carbon footprint of buildings with ecview. *Computer* **99**(PrePrints) (2010)
14. Pacific Controls: Galaxy. <http://pacificcontrols.net/products/galaxy.html> (2011). Last Accessed 7 Feb 2012

15. EPA: Energy cost and iaq performance of ventilation systems and controls—executive summary. Technical Report EPA-4-2-S-01-001, United States Environmental Protection Agency (2000)
16. Bulut, A., Singh, A.: A unified framework for monitoring data streams in real time. In: Proceedings of 21st International Conference on Data Engineering, ICDE 2005, pp. 44–55, Apr 2005
17. Hauder, M., Gil, Y., Liu, Y.: A framework for efficient data analytics through automatic configuration and customization of scientific workflows. In: eScience, pp. 379–386. IEEE Computer Society (2011)
18. Ekanayake, J., Pallickara, S., Fox, G.: Mapreduce for data intensive scientific analyses. In: Proceedings of the 4th IEEE International Conference on eScience, Washington, DC, USA, pp. 277–284. IEEE Computer Society (2008)
19. AlertMe: <http://www.alertme.com>. Last Accessed 17 Feb 2012
20. xively: <https://xively.com/>. Last Accessed 27 Aug 2013
21. Matsuura, M., Suzuki, T., Shiroyama, H.: Stakeholder assessment for the introduction of sustainable energy and environmental technologies in Japan. In: IEEE International Symposium on Technology and Society, ISTAS '09, pp. 1–9, May 2009
22. Yun, C.H., Han, H., Jung, H.S., Yeom, H.Y., Lee, Y.W.: Intelligent management of remote facilities through a ubiquitous cloud middleware. In: IEEE International Conference on Cloud Computing, CLOUD '09, pp. 65–71, Sept 2009
23. Balazinska, M., Deshpande, A., Franklin, M.J., Gibbons, P.B., Gray, J., Hansen, M., Liebhold, M., Nath, S., Szalay, A., Tao, V.: Data management in the worldwide sensor web. *IEEE Pervasive Comput.* **6**, 30–40 (2007)
24. Rolewicz, I., Catasta, M., Jeung, H., Miklós, Z., Aberer, K.: Building a front end for a sensor data cloud. In: Proceedings of the International Conference on Computational Science and its Applications—Volume Part III, ICCSA'11, pp. 566–581. Springer, Berlin, Heidelberg (2011)
25. Li, T., Liu, Y., Tian, Y., Shen, S., Mao, W.: A storage solution for massive iot data based on nosql. In: GreenCom, pp. 50–57. IEEE (2012)
26. CA AppLogic: <http://www.ca.com/us/cloud-platform.aspx> (2013). Last Accessed 23 Dec 2013
27. Appistry Platform: <http://www.appistry.com/products> (2013). Last Accessed 23 Dec 2013
28. Google App Engine: <http://code.google.com/appengine/> (2013). Last Accessed 23 Dec 2013
29. Microsoft Azure Services Platform: <http://www.microsoft.com/azure/default.msp> (2013). Last Accessed 23 Dec 2013
30. Amazon: Amazon DynamoDB. <http://aws.amazon.com/dynamodb/>. Last Accessed 17 Feb 2012
31. Parabon Frontier: <http://www.parabon.com/> (2013). Last Accessed 23 Dec 2013
32. Grossman, R.L., Gu, Y., Mambretti, J., Sabala, M., Szalay, A., White, K.: An overview of the open science data cloud. In: Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing, HPDC '10, New York, NY, USA, pp. 377–384. ACM (2010)
33. BuildingSMART International Limited: Ifc4. <http://www.buildingsmart-tech.org/specifications/ifc-releases/ifc4-release/ifc4-release-summary>. Last Accessed 15 Nov 2013
34. Truong, H.L., Dustdar, S.: On evaluating and publishing data concerns for data as a service. In: APSCC, pp. 363–370. IEEE Computer Society (2010)
35. Truong, H.L., Phung, P.H., Dustdar, S.: Governing bot-as-a-service in sustainability platforms—issues and approaches. *Proc. CS* **10**, 561–568 (2012)