

M2M Platform-as-a-Service for Sustainability Governance

Hong-Linh Truong and Schahram Dustdar

Distributed Systems Group, Vienna University of Technology, Austria, Email:{truong,dustdar}@dsg.tuwien.ac.at

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 and analytics applications to support the needs of different consumers. However, there exist complex relationships between monitored objects, monitoring data, analysis features, and stakeholders in M2M that require efficient ways to handle these complex relationships. This paper presents techniques for linking and managing monitored objects, sustainability monitoring data and analytics applications for different stakeholders in cloud-based M2M systems. We describe a Platform-as-a-Service for sustainability governance that implements these techniques. We also illustrate our prototype based on a real-world cloud system for facility monitoring.

I. 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: (i) data about building elements (e.g., floors, equipment, and electricity systems), (ii) data about sensor configuration and status (e.g., sensor configuration parameters, sensor location and sensor data type), and (iii) 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. Our work is focused on supporting analytics for understanding complex sustainability measurements (e.g., electricity consumption and GHG calculation). Research effort so far has been concentrated on techniques for low level data management, such as data storage and query mechanisms, but has neglected the complexity and diversity of stakeholders and their interests. Therefore, sustainability data and corresponding analytics applications are not well managed to support multiple stakeholders.

This paper focuses on a Platform-as-a-Service (PaaS) for sustainability governance. Our goal is to develop techniques for linking monitored objects, monitoring data, and applications to stakeholder's needs and provide a PaaS for stakeholders to access data and perform data analytics of their own interest. This paper contributes (i) techniques for linking and enriching data for sustainability analysis, and (ii) a PaaS for managing relationships between stakeholders, monitored objects, monitoring data and applications. To illustrate the

This work is partially funded by the Pacific Controls Cloud Computing Lab. We thank Manu Ravishankar, Saneesh Kumar and Sulaiman Yousuf for providing useful information.

usefulness of our work, we present our prototype atop a real-world cloud system for facility monitoring.

The rest of this paper is organized as follows: Section II presents related work. We present our linking data techniques in Section III. Section IV describes our PaaS for sustainability governance. Section V presents our experiments. We conclude the paper and outline our future work in Section VI.

II. RELATED WORK

Recently, several cloud-based platforms to support the monitoring of energy consumption have been introduced, such as AlertMe [1] and Cosm/Pachube [2]. Furthermore, there are systems supporting sustainability governance for buildings, such as the Galaxy platform [3]. 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. On the other hand, generic data management techniques for sensor data, such as [4], 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 [5]. However, it does not deal with complex relationships between monitoring data and sustainability governance features for facilities.

Several general cloud services have been developed, such as Microsoft Azure [6] and Amazon DynamoDB [7]. 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 [8] but they do not address facility monitoring data.

III. LINKING M2M CLOUD DATA

One of the most important issues in M2M is how to link monitoring data to monitored objects. This is crucial for several analyses, such as data analytics, monitoring data search and composition, and quality of data monitoring. Several specifications have been developed, such as the IFC classes [9], to cover building structure data. From existing specifications, main data concepts that can be used to enrich

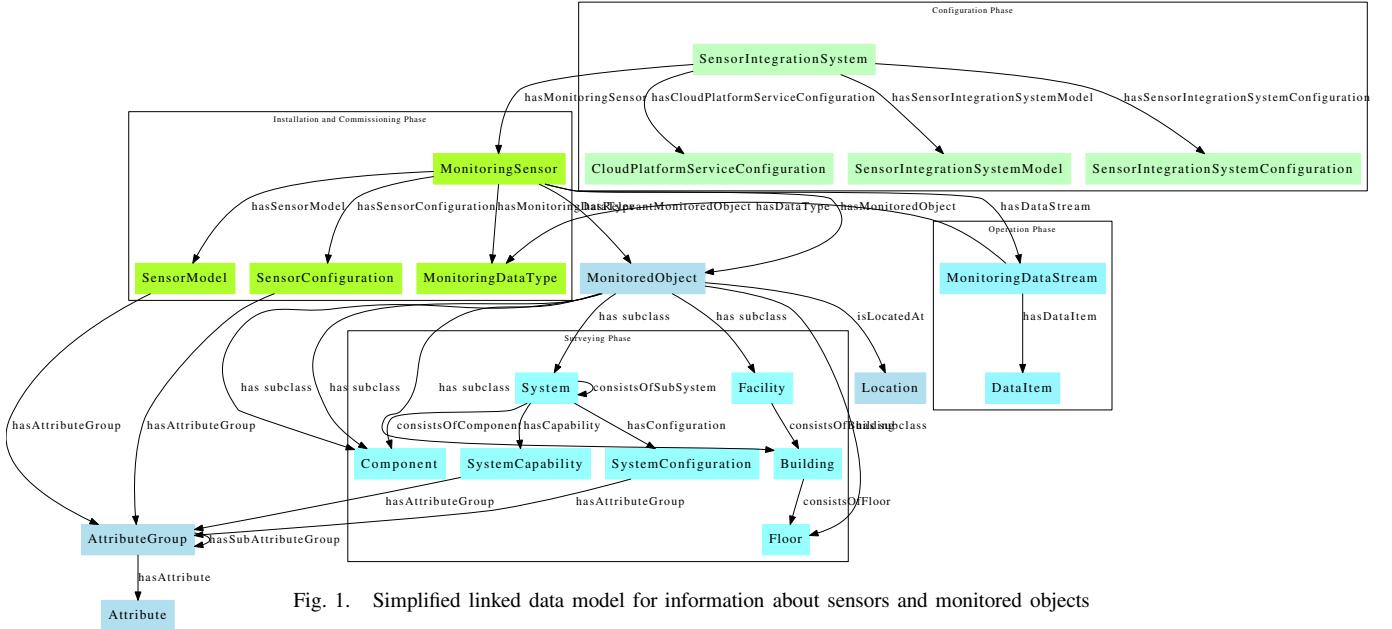


Fig. 1. Simplified linked data model for information about sensors and monitored objects

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. However, currently, we lack techniques to link building information to realtime monitoring information to support advanced analytics.

To manage complex relationships among data, we have developed a linked data model, shown in Figure 1. With this model, before operating buildings, 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, data items from sensor data streams will be stored separately from building structure data but they are linked by using identifiers.

As building-specific gateways 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>).

Based on the linked data model, different processes, used to handle parts of the linked data model, can be integrated. For example, a process may consist of a set of visual analytics allowing stakeholders to retrieve and manipulate data or may include workflow activities analyzing different types of data. All processes will rely on the linked data model and a *Linked Monitoring Data Service* to store and manage information about sensors and building elements, shown in Figure 2.

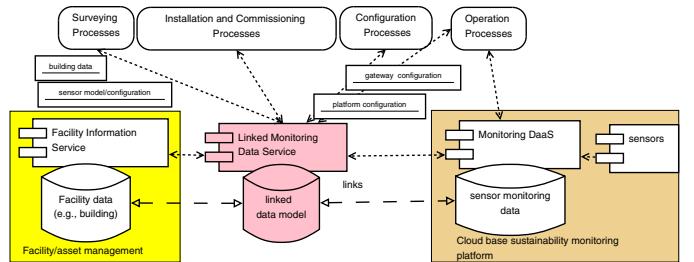


Fig. 2. Integrating processes for managing linked data

IV. CLOUD SUSTAINABILITY GOVERNANCE FRAMEWORK

Figure 3 presents our sustainability governance framework which integrates stakeholder information, monitored objects, monitoring data and applications. The consumers can access governance features via *SusGovSaaS* or *SusGovPaaS*. The first one offers the SaaS model for stakeholders and utilizes the latter, which provides PaaS for sustainability governance. To demonstrate our *SusGovPaaS*, we develop a *SusGov Analysis Framework* aiming at sustainability analytics experts. Automated applications (Bots) can utilize *SusGovPaaS* to execute or download analysis applications. 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 identifications (stored in Facility Information Service). In our prototype, we focus on *SusGovPaaS, Application Store & Stakeholder Profile Service*. We utilize Jersey – and implementation of JAX-RS for RESTful Web Services – and Weblogic 10.3 for developing our SusGovPaaS Web services. *Application Store & Stakeholder Profile Service* are RESTful Web services that store their information under RDF/XML format and use Allegro Graph (<http://www.franz.com/agraph/allegrograph>). In the following, we discuss some features in our SusGov framework.

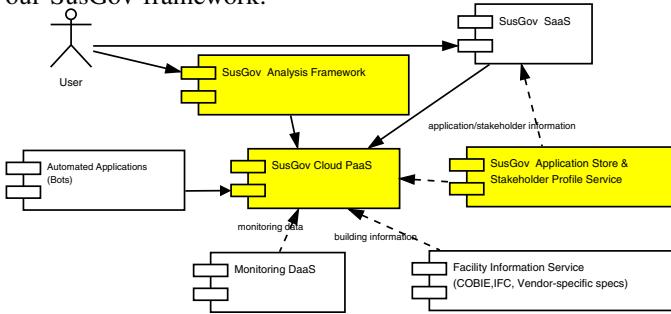


Fig. 3. Conceptual architecture of our prototype

PaaS Application Discovery API		PaaS Application Execution API		PaaS Data Discovery API		PaaS Data Retrieve API	
Application Management and Execution		Batch app	Workflow app	Stream app	Bot	Monitoring DaaS	Facility Information Service
		C/C++ Env.	Python/R/Matlab Env.	Java Env.			
App/Bot Store		IaaS	IaaS	Bot Hosting Environment			

Fig. 4. Diverse forms of applications and execution environments

1) *SusGov Platform-as-a-Service*: Figure 4 outlines main building blocks in our sustainability governance 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. Currently we use monitoring data from the Galaxy platform [3]. 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 [10]. To expose capabilities of applications and monitoring data for different stakeholders, several APIs are provided for discovering, retrieving and accessing/executing data and applications.

2) *Describing SusGov applications*: Sustainability applications in sustainability governance are very different than 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 sim-

ple 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. In our work, applications are encapsulated in the so-called *sustainability governance applications* (*SusGovApp*). Overall, information about a *SusGovApp* will include (i) category of the application (*SusGovAppCategory*), (ii) metadata about the application (*SusGovAppDescription*), (iii) concrete monitoring data (*MonitoringDataIdentification*), (iv) abstract supporting monitoring data types, (*TypeOfMonitoringData*), and (v) 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.

In our model, we consider applications as black boxes and these applications can be exposed via a generic interface. For example, instead of defining detail logic of a complex analysis application as a workflow and managing the workflow, the workflow can be considered as an input parameter of an application which is implemented as a batch job by running a workflow engine with the input parameter. Depending the implementation, 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.

3) *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. 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 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.

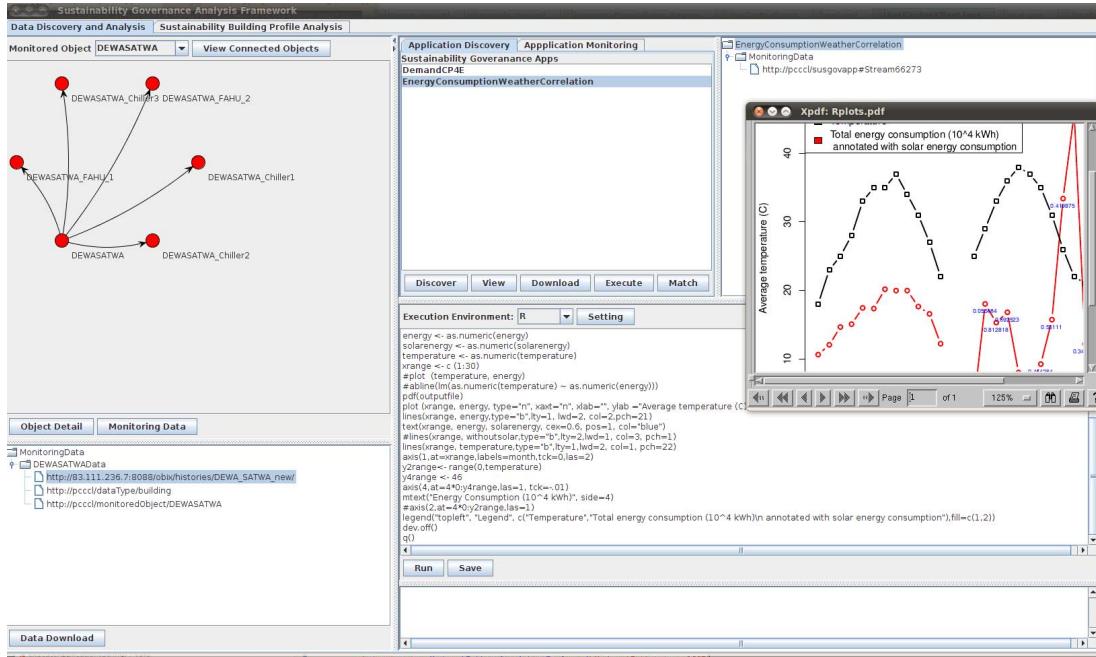


Fig. 5. Example of SusGov Analysis Framework

V. ILLUSTRATING EXPERIMENTS

In our prototype, near-realtime monitoring data are obtained from building-specifics gateways, which are based on Niagara AX (<http://www.niagaraax.com/>) as part of the Galaxy platform [3]. 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 and described in our linked data model, specifies dependencies among different monitored objects, e.g., which sensors are associated with which chillers in which buildings.

Figure 5 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 via a dependency graph (in the top-left window of Figure 5). This graph allows the user to examine the influence among monitored objects, e.g., which monitored objects are relevant to the analysis of a room. For each monitored object, the user can also see which types of monitoring data she can access (in the bottom-left window of Figure 5). 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 Figure 5). Based on applications 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. For example, the chart in Figure 5 shows an analysis result from a R-based application. By downloading monitoring data and applications, the user can also execute the

applications in her local execution environment (e.g., in the window – Execution Environment – in Figure 5).

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we analyze relationships among sustainability monitoring data, analysis applications and monitored objects. Based on that we design our SusGovPaaS for supporting data and application discovery, and analysis application management and execution that can be used by various stakeholders and automated applications. 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 visual data analytics techniques for sustainable building profile analysis.

REFERENCES

- [1] “Alertme,” <http://www.alertme.com>. Last access: 17 Feb 2012.
- [2] “Cosm,” (Pachube), <http://cosm.com/>, Last access: 22 May 2012.
- [3] “Pacific controls galaxy,” 2011, <http://pacificcontrols.net/products/galaxy.html>.
- [4] M. Balazinska, A. Deshpande, M. J. Franklin, P. B. Gibbons, J. Gray, M. Hansen, M. Liebhold, S. Nath, A. Szalay, and V. Tao, “Data management in the worldwide sensor web,” *IEEE Pervasive Computing*, vol. 6, pp. 30–40, April 2007.
- [5] I. Rolewicz, M. Catasta, H. Jeung, Z. Miklós, and K. Aberer, “Building a front end for a sensor data cloud,” in *Proceedings of the 2011 international conference on Computational science and its applications - Volume Part III*, ser. ICCSA’11. Springer-Verlag, 2011, pp. 566–581.
- [6] Microsoft Azure Services Platform, 2009, <http://www.microsoft.com/azure/default.mspx>. Last access: 8 July 2009.
- [7] “Amazon DynamoDB,” <http://aws.amazon.com/dynamodb>.
- [8] R. L. Grossman, Y. Gu, J. Mambretti, M. Sabala, A. Szalay, and K. White, “An overview of the open science data cloud,” in *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, ser. HPDC ’10. New York, NY, USA: ACM, 2010, pp. 377–384.
- [9] “Ifc,” http://en.wikipedia.org/wiki/Industry_Foundation_Classes.
- [10] H. L. Truong, P. H. Phung, and S. Dustdar, “Governing bot-as-a-service in sustainability platforms - issues and approaches,” *Procedia CS*, vol. 10, pp. 561–568, 2012.