Sensor Data Management, Validation, Correction, and Provenance for Building Technologies

Charles Castello, PhD
Oak Ridge National Laboratory
Member ASHRAE

Jibonananda Sanyal, PhD
Oak Ridge National Laboratory
Member ASHRAE

Jeffrey Rossiter
Oak Ridge National Laboratory

Zachary Hensley
Tennessee Tech University

Joshua New, PhD
Oak Ridge National Laboratory
Member ASHRAE

ABSTRACT

Oak Ridge National Laboratory (ORNL) conducts research on technologies that use a wide range of sensors to develop and characterize building energy performance. The management of high-resolution sensor data, analysis, and tracing lineage of such activities is challenging. Missing or corrupt data due to sensor failure, fouling, drifting, calibration error, or data logger failure is another issue. This paper focuses on sensor data management, validation, correction, and provenance to combat these issues, ensuring complete and accurate sensor datasets for building technologies applications and research. The design and development of two integrated software products are discussed: Sensor Data Validation and Correction (SensorDVC) and the Provenance Data Management System (ProvDMS) platform. Both software products are designed to be general purpose with applications to a wide range of building performance management scenarios.

INTRODUCTION

The U.S. building sector accounted for approximately 41% of primary energy consumption in 2010 costing $431 billion in total energy expenditures (DOE. 2010). This consumption produced 584 million metric tons of carbon dioxide emissions (40% of total carbon dioxide emissions in the U.S.) Based on these numbers, the U.S. consumed 18.7% of primary energy in the global buildings sector, second to China, which consumed 20%. However, China's population is significantly larger, making up 20% of the world's population compared to the U.S. which is only 4.6%. This creates a significant opportunity to decrease energy usage and cost in the U.S. through building technologies that promote energy efficiency.

Virtually all buildings currently have utility monitoring devices which collect data for billing purposes. For residential buildings, the bill typically displays monthly data from the past year. In some cases, this data is shared by the utility in an electronic format via a web portal. Some utility providers have deployed area-wide smart meters capable of breaking down utility usage in greater detail. For commercial buildings, hourly data is more common and can involve energy management systems which handle sensor data logging as well as scheduled automation of lights and other building controls. Buildings, systems, components, and plug load devices are expected to continue becoming more intelligent and interconnected.

Increasingly capable sensors continue to become cheaper and non-intrusive load monitoring devices are beginning to enter the market which can perform signature analysis on utility signals to generate dozens of channels of data on individual plug loads from a single physical sensor. The SensorDVC and ProvDMS systems presented in this paper are scalable to the number of channels and sampling resolution of today's and future buildings. The ORNL facilities provided the resources to field test the scalability of the software solutions by applying them to large data sets. In addition, there are several ASHRAE Technical Committees which deal with faulty/corrupt measured data or tracking modifications to data as it proceeds through a series of transformations. The domain-agnostic algorithms used in the software tools are equally applicable in these contexts.

Oak Ridge National Laboratory (ORNL) is heavily engaged in buildings research; exploring innovative technologies to help drive down energy consumption thereby reducing cost and carbon dioxide emissions. An example of this is the ZEBRAlliance research project (ZEBRAlliance. 2012), a collaboration between Schaad Companies, Tennessee Valley Authority (TVA), ORNL, BarberMcMurry Architects, and the Department of Energy (DOE). Four residential homes were built in 2008, located in the Wolf Creek subdivision in Oak Ridge, TN, to integrate experimental energy efficient technologies and determine the performance and cost at the component, system, and whole-building levels. Each home was built with a variety of different energy efficient technologies. Analysis centered on data that was collected from sensors that were integrated in each home. The first and second homes contain 279 sensors each, the third home has 321 sensors, and the fourth home has 339 sensors. Types of sensors include temperature (thermistors, thermocouples, and combo probes), humidity (RH and combo probes), and electrical demand (watt-meters). The majority of sensors have a 15-minute resolution with approximately 80 sensors having a 1-minute resolution. Another example is the Campbell Creek Energy Efficient Homes Project (TVA. 2010), a collaboration between TVA, ORNL, and Electric Power Research Institute (EPRI). Three residential homes were built in 2008, located in the Campbell Creek subdivision in Knoxville, TN. The first home is being used as a control for the project (a.k.a., Building Home), the second home was retrofitted with energy efficient technologies (a.k.a., Retrofit Home), and the third home was built using the latest and most advanced construction technologies to maximize energy efficiency (a.k.a., High Performance Home). Each home has approximately 140 sensors.

ORNL also has research apparatuses called Flexible Research Platforms (FRP), which were built in 2012 and are located on ORNL's main campus. The FRPs represents a novel approach to light commercial building's research by allowing researchers to test different building configurations in an unoccupied setting. This is possible through the FRPs' core components: (1) concrete slabs; (2) metal frames; (3) utilities; and (4) IT infrastructure. Integrated whole-building design comprised of envelopes, heating, ventilation, and air-conditioning (HVAC) equipment, and controls are implemented into the FRPs. The 1-story FRP (FRP1) is 40x60 ft (2,400 ft^2) which is currently intended to improve the energy efficiency of metal buildings and the 2-story FRP (FRP2) is 40x40 ft (3,200 ft^2) which is currently intended to improve the energy efficiency of a wide variety of light commercial buildings. FRP1 and FRP2 have 413 and 617 sensors respectively. Buildings from all three projects are shown in **Figure 1**.

Figure 1 These are photos of ZEBRAlliance House 1 at the left (Miller et al. 2012), a home at Campbell Creek in the middle (TVA. 2010), and the 1-story and 2-story Flexible Research Platforms that represent light commercial buildings at Oak Ridge National Laboratory shown to the right (Hughes. 2012).

The commonality of these projects is the reliance on sensors. Data being collected from sensors are used to analyze the performance of energy efficient technologies on component, system, and whole-building levels. Furthermore, sensor data are used to calibrate building models using software tools such as EnergyPlus (DOE. 2012), OpenStudio (NREL. 2012), and DesignBuilder (DesignBuilder Software Ltd. 2005). Sensor data from buildings are also used to make decisions in building automation systems, fault detection and diagnosis (FDD) for heating, ventilation, and air-conditioning (HVAC) equipment, and much more.

The three projects mentioned above generate over one billion data points per year. The majority of data collection activities for building technologies research at ORNL use Campbell Scientific data loggers that ping sensors for data at a user-defined resolution (typical range from 1 second to 1 hour). Data is forwarded and collected on a connected personal computer (PC) and forwarded again to an ORNL server. Data is stored in a database or as comma-separated values (CSV) files, typically having separate files per building, per month, per data logger, and per resolution.

Whether sensor data is collected for research or real-world applications, comparison and extension of known algorithms are needed for innovative solutions to manage this data in a more efficient manner. As data complexity grows and loosely coupled software systems evolve around the data used repeatedly in ad-hoc processes, the quality assurance of data products and experimental results become important. Complex transformation of the data, for error correction or for validation, becomes important to trace. The definition of data validation in this paper is rooted from computer science, which means to ensure data are clean, correct, and useful. Data acquisition activities at ORNL have shown up to 14.59% of missing data from data logger and sensor failure. This rate doesn't even include data inaccuracy from sensor fouling and calibration error. The knowledge of the provenance, or data *lineage*, help users in quantifying the source and derivations of the data, transformations, repeatability of workflows on similar (newer) data, and in providing holistic meta-information of the scientific process. Data correction and provenance is vital in managing issues arising from missing or corrupt data due to sensor failure, fouling, drifting, calibration error, or data logger failure.

This paper presents research to meet data management needs, provenance needs, and solve issues related to missing and corrupt data. The cloud-based Provenance Data Management System (ProvDMS) is used to manage data from building technologies research and applications while tracking the provenance of data. The Sensor Data Validation and Correction (SensorDVC) desktop application is used to validate and correct sensor data using statistical and filtering methodologies. These software applications can be used to facilitate building automation and control systems, building monitoring, and building design and modeling activities. These software solutions also ensure data is managed in a robust fashion while guaranteeing the completeness and accuracy of the data.

This paper includes the background of both the SensorDVC and ProvDMS integrated software products highlighting the necessity of these systems. Details of the design and development, and the methodology employed are provided. The paper concludes by discussing planned future work.

BACKGROUND

This section discusses previous research that was used for the SensorDVC and ProvDMS software applications.

Data Correction and Validation

There are many examples of data validation for numerous applications (Ibarguengoytia et al. 2001; Frolik et al. 2001; Uluyol et al. 2006; Postolache. 2005). Two Bayesian networks were used by Ibarguengoytia et al., for the detection of faults in a set of sensors; the first represents the dependencies among all sensors and the second isolates the faulty sensor. Self-validation, fusion, and reconstruction of sensor data was tackled by Frolik et al., by exploring three key steps: (1) employ fuzzy logic rules for self-validation and self-confidence; (2) exploit near-linear relationships between sensors for reconstructing missing or low-confidence data; and (3) fuse this data into a single measurement along with a qualitative indicator for its reliability. A start-up fault detection and diagnosis method was presented for gas turbine engines by Uluyol et al., which consisted of three key techniques: (1) statistics; (2) signal processing; and (3) soft computing. Sensor profiles were generated from good and bad engine start-ups in which a feature vector was calculated and signal processing was used

for feature selection. In the signal-processing step, principal component analysis (PCA) was applied to reduce the samples consisting of sensor profiles into a smaller set. The features obtained from this step were then classified using neural-network-based methods. A Kohonen self-organizing map (K-SOM) was used by Postolache, to perform sensor data validation and reconstruction. Sensor failure and pollution event detections were also studied with the use of this methodology for a water quality sensor network application.

The SensorDVC desktop application is based on research dealing with sensor data validation and correction using statistical (Castello and New. 2012) and filtering (Castello et al. 2013) techniques. Research regarding statistical processing methods (Castello and New. 2012) investigated four techniques: (1) least squares; (2) maximum likelihood estimation; (3) segmentation averaging; and (4) threshold based. Experiments were run using data from the ZEBRAlliance research project (Miller et al. 2012) to test the performance of these techniques in predicting data points that were corrupt and missing. Temperature, humidity, and energy data were investigated. Artificial gaps were introduced by randomly removing portions of existing data for testing the accuracy of the statistical algorithms. This was accomplished by splitting the original dataset into two subsets: training (70%) and testing (30%). Each sensor was treated as an independent variable where predictions were based upon a variable-sized window of observations, w. A prediction model is generated for each window of observations and auto-correction occurs if values are missing or corrupt (far away from the predicted value). The performance metrics used, in order of significance, are absolute error (AE), relative error (RE), and root-mean-square error (RMSE). A summary of testing results (Castello and New. 2012) is shown in **Table 1**. The threshold based technique performed best with temperature (ϵ=2), humidity (ϵ=2), and energy data (ϵ=1) where ϵ is the number of standard deviations used to calculate the threshold in the threshold-based algorithm.

Table 1. Summary of Testing Results for the Statistical Correction Algorithms (Castello and New. 2012)

	Temperature in °F (°C)				Humidity in %RH				Energy in Wh (kJ)			
	w	*AE*	*RE*	*RMSE*	*w*	*AE*	*RE*	*RMSE*	*w*	*AE*	*RE*	*RMSE*
LS	12	4.2%	6.4%	3.44 (-5.87)	24	5.4%	9.5%	6.10	24	12.3%	890.0%	24.66 (88.78)
MLE	12	3.1%	4.6%	2.49 (-6.39)	12	4.8%	8.2%	4.71	96	7.6%	391.3%	10.92 (39.31)
SA	48	12.9%	15.8%	10.25 (-2.08)	6	8.6%	21.2%	8.09	6	7.4%	340.5%	9.93 (35.75)
TB (ϵ=1)	6	2.6%	3.9%	1.94 (-6.70)	6	4.2%	7.3%	3.93	6	7.3%	241.3%	10.19 (36.68)
TB (ϵ=2)	6	2.5%	3.8%	1.92 (-6.71)	6	3.9%	6.7%	3.62	12	7.3%	355.3%	9.99 (35.96)
TB (ϵ=3)	6	2.5%	3.8%	1.93 (-6.71)	6	3.9%	6.7%	3.64	6	7.4%	369.9%	9.83 (35.39)

Pressure and airflow data types are investigated in this paper beyond (Castello and New. 2012) using the four statistical correction algorithms. Testing results are shown in **Table 2**. The AE ranges from 4.5% to 8.7% for pressure data and 0.5% to 1.2% for airflow data. RE ranges from 6.6% to 14.2% for pressure data. However, RE ranges from a low 2.2% to a high 164.2% for airflow data. The RMSE for pressure data ranges from 23.69 psi (163.34 kPa) to 53.25 psi (367.15 kPa) and airflow data ranges from 0.35 ft^3 (0.01 m^3) to 1.49 ft^3 (0.04 m^3). The best performer is the threshold based method, particularly when ϵ=2.

Table 2. Summary Results Using Statistical Correction Algorithms for Pressure and Airflow Type Data

	Pressure in psi (kPa)				Airflow in ft^3 (m^3)			
	w	*AE*	*RE*	*RMSE*	*w*	*AE*	*RE*	*RMSE*

LS	6	6.5%	9.5%	32.34 (222.98)	12	0.9%	78.4%	0.68 (0.02)
MLE	12	7.5%	13.9%	43.77 (301.78)	12	0.7%	70.4%	0.55 (0.02)
SA	12	8.7%	14.2%	53.25 (367.15)	48	1.2%	164.2%	1.49 (0.04)
TB (ϵ=1)	6	4.9%	6.6%	27.98 (192.92)	6	0.5%	2.3%	0.41 (0.01)
TB (ϵ=2)	6	4.5%	7.0%	25.43 (175.33)	6	0.5%	2.2%	0.35 (0.01)
TB (ϵ=3)	6	4.5%	7.5%	23.69 (163.34)	6	0.5%	2.5%	0.40 (0.01)

Two techniques were investigated regarding filtering processing methods (Castello et al. 2013): Kalman and linear predictive coding (LPC) filters. The accuracy of both algorithms was determined using a similar approach to the statistical methods mentioned in the previous paragraph (Castello and New. 2012). Types of data that were investigated include: (1) temperature; (2) humidity; (3) energy usage; (4) pressure; and (5) airflow. Results from this study (**Table 3**) shows the Kalman filter performed best with temperature, humidity, pressure, and airflow data using observation window sizes of w=12 (3 hours), w=96 (24 hours), w=12 (3 hours), and w=4 (1 hour) respectively. The LPC filter performed best with energy usage data using an observation window size of w=4 (1 hour).

Table 3. Summary of Testing Results for the Filtering Correction Algorithms (Castello et al. 2013)

	Kalman				**LPC**			
	w	*AE*	*RE*	*RMSE*	*w*	*AE*	*RE*	*RMSE*
Temperature in °F (°C)	12	3.6%	5.3%	2.75 (-16.25)	96	7.0%	10.0%	11.17 (-11.57)
Humidity in %RH	96	5.2%	8.9%	5.35	96	9.2%	15.1%	14.36
Energy in Wh (kJ)	48	9.7%	468.5%	13.75 (49.50)	4	9.6%	109.1%	12.92 (46.51)
Pressure in psi (kPa)	12	3.5%	74.0%	0.22 (1.52)	96	12.0%	22.5%	91.76 (632.66)
Airflow in ft^3 (m^3)	4	0.6%	0.3%	0.00 (0.00)	48	0.6%	66.0%	0.98 (0.03)

A comparison of results for the statistical and filtering methods is shown in **Table 4**. The threshold based statistical method performed best with temperature, humidity, energy, and airflow data while the Kalman filtering method performed best with pressure data. The best performers for each type of data have acceptable AE, ranging from 0.5% to 9.6%. However, the RE ranges from a low 0.3% to a high 241.3%. The larger REs are associated with the energy data type which comes from the refrigerator of ZEBRAlliance House #2. This could be due to the large energy spikes of the refrigerator due to lights coming on when doors are opened and cycling of the compressor. The results are showing statistical and filtering methods are having a challenge with this type of data compared to the others.

Table 4. Comparison of Results for Statistical and Filtering Correction Algorithms

	Statistical					**Filtering**				
	w	*AE*	*RE*	*RMSE*	*Method*	*w*	*AE*	*RE*	*RMSE*	*Method*
Temperature in °F (°C)	6	2.5%	3.8%	1.92 (-6.71)	TB (ϵ=2)	12	3.6%	5.3%	2.75 (-16.25)	Kalman

Humidity in %RH	6	3.9%	6.7%	3.62	TB (ϵ=2)	96	5.2%	8.9%	5.35	Kalman
Energy in Wh (kJ)	6	7.3%	241.3%	10.19 (36.68)	TB (ϵ=1)	4	9.6%	109.1%	12.92 (46.51)	LPC
Pressure in psi (kPa)	6	4.5%	7.0%	25.43 (175.33)	TB (ϵ=2)	12	3.5%	74.0%	0.22 (1.52)	Kalman
Airflow in ft^3 (m^3)	6	0.5%	2.2%	0.35 (0.01)	TB (ϵ=2)	4	0.6%	0.3%	0.00 (0.00)	Kalman

Research was also performed for sensor data validation and correction using machine learning algorithms (Smith et al. 2013) but currently has not been included in the SensorDVC application.

Data Management and Provenance

Provenance is a term pertaining to the inherent *lineage* of objects as they evolve over time. Provenance has been well researched and different approaches pertinent to application domains have been presented. Simhan, Plale and Gannon (Simmhan et al. 2005) present a high-level taxonomy of data provenance differentiating on the application, subject, representation, storage, and dissemination of the provenance. A number of provenance tools are available for different disciplines. *Chimera* (Foster et al. 2002), a prototype of the *Virtual Data Grid* (Foster et al. 2003) provides a solution for various scientific fields. *Taverna* (Oinn et al. 2004) is a process-oriented workflow environment with provenance support. The *EU Provenance Project* (Vázquez-Salceda et al. 2008) provides an open architecture for provenance projects. The PASS: Provenance Aware Storage System (Muniswamy-Reddy, 2006) builds provenance into the file storage system. Yet another tool, *Karma* (Simmhan et al. 2008), traces the lineage of data in a cyber-infrastructure framework.

For the management of ORNL's sensor data and tracing of their participation (and transformations) in various experiments, a provenance based data management system, ProvDMS, was built that allows researchers to access, share, trace, and query data lineage using a web-based interface. Sensor data from the experimental facilities, often at the sub-minute interval resolution, are constantly collected. The effective management and sharing of this data poses a challenging problem, not just for experimental facilities but more generally, as detailed information is becoming increasingly common. In the typical order of business, the data is appended to CSV files which are located on shared network locations. Researchers access these locations and manually select temporal subsets of the data for their analysis and modeling needs. Sometimes, data sets are shared between researchers via email attachments.

Such methodologies of data management do not scale well in this age of big-data. Data constantly undergoes transformations through various operations on the data such as correction and validation, scaling and transposition, and sometimes deletion. Sometimes data participates in a workflow and undergoes transformation in stages, often with the addition of more knowledge. There is very little metadata that is saved about any such data transformation leading to confusion and loss of productivity.

There are many aspects of provenance design that are important to determine. Primary among them is the required granularity of meta-information that must be stored. A fine granularity is often helpful in understanding finer aspects of the data but adds considerable storage overhead. A coarse granularity often abstracts information to the point that it is not meaningful. Therefore, it is important to determine the granularity of information that must be preserved.

Another aspect is integration of provenance with workflows. Operations on data are usually in some order of application. One of the main reasons why provenance tracking is not so common is because most provenance tracing systems enforce certain restrictions upon their users in terms of choice of tools that they use on their data. This severely limits their flexibility. Additionally, most software tools are not 'provenance-ready'. We have limited our scope and added a dimension of data management to effectively carve out a case where the tracking of provenance is not an imposition but a byproduct of solving researchers' data needs.

METHODOLOGY

This section discusses the methodology, specifically dealing with design and development for the SensorDVC and ProvDMS software applications.

SensorDVC Desktop-Based Application Architecture

The SensorDVC application is implemented with the Qt graphical user interface (GUI) application framework (Qt Project. 2013). Qt was chosen because it is an open-source and cross-platform framework. Qt also has a long history of steady development (since 1991) and excellent documentation. Algorithm implementation for the statistical and filtering techniques was prepared using Mathwork's MATLAB software, a numerical computing environment. In order to use developed MATLAB functions with Qt's C++ coding environment, MATLAB Compiler was used to convert functions to C++ shared libraries. Compiler executes MATLAB code within the MATLAB Compiler Runtime (MCR) which is a standalone set of shared libraries that allows the execution of MATLAB components without the installation of MATLAB.

The Model-View-Controller (MVC) design pattern is used for the SensorDVC application. The MVC pattern simplified data management within the application because there is a single model containing all of the application's data. Initial design separated data between the main window (original data), validation dialog (validation data), and correction dialog (correction data). Under MVC, each dialog has a view element to display the data via the model's interface. The dialogs along with their view elements allow the user to manipulate data through the model's interface. **Figure 2** provides an overview of the structure of the SensorDVC application. MainWindow is the core of the application and provides an interface to its constituent components. MMAValidationDialog (Min/Max All Validation) and PerRowValidationDialog presents data views and controls related to data validation. AllCorrectionDialog and PerChannelCorrectionDialog present data views and controls related to data correction. SDVPlotWidget manipulates SDVGraphicsView and SDVGraphicsAxisItem components to present some basic data visualization. SDVGraphicsAxisItem is used by SDVPlotWidget to display x- and y-axes. SDVGraphicsHRangeItem is used to represent a range of data with a horizontal orientation. The x-axis is composed of SDVGraphicsHRangeItems. SDVGraphicsVRangeItems are used for vertically oriented ranges (the y-axis). AllChannelsTableDialog is composed of two data views; one for original data and one for corrected data. The Scanner component tokenizes data files. The Parser component simply organizes imported data files via tokens received from the Scanner.

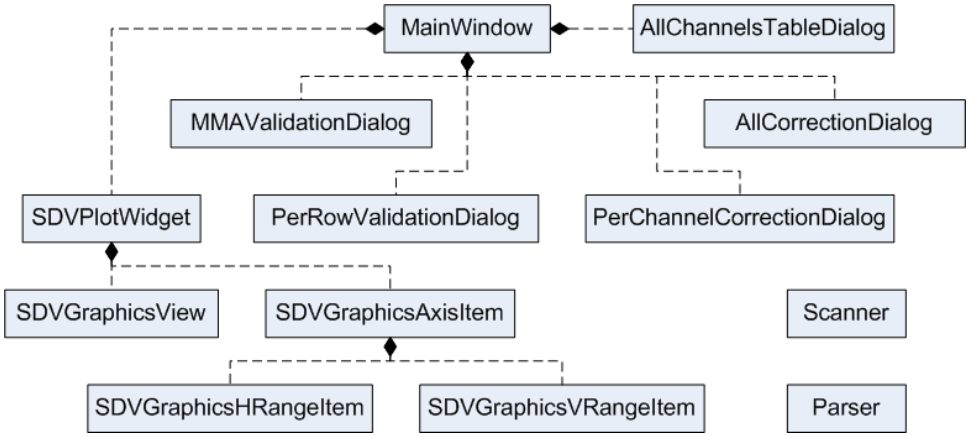


Figure 2 Unified Modeling Language (UML) overview of SensorDVC components.

The SensorDVC application offers basic data visualization functionality through QGraphicsView (Qt's graphics framework). In order to display objects using QGraphicsView, the objects must be added to a QGraphicsScene. The QGraphicsScene is then added to a QGraphicsView object. SDVPlotWidget contains a QGraphicsView object and a QGraphicsScene object. The data being plotted is treated as an image that is scaled to the pixel dimensions of the

SDVPlotWidget using nearest-neighbor interpolation resulting in the production of a bitmap image. The image is then added to the QGraphicsScene. SDVGraphicsAxisItem, SDVGraphicsVRangeItem, and SDVGraphicsHRangeItem are subclasses of QGraphicsObject. An SDVGraphicsAxisItem is added to the QGraphicsScene for each axis.

ProvDMS Cloud-Based Application Architecture

In the design of our provenance system, we focused heavily on researcher requirements, granularity of the provenance, workflow requirements, and object design. Our design principles emphasize the importance of user needs, taking a cohesive but independent stance on the integration of provenance with user tools.

Our system uses the Core Provenance Library (CPL) (Macko and Seltzer, 2012) which is a software library designed to integrate with existing software systems giving users complete choice over which aspects of the system are saved for a provenance trace. CPL uses a versioning system to automatically handle new versions of objects. Ancestry links are created through "data flows" – links that describe data transformations or control actions.

The biggest advantage of using the CPL to build ProvDMS over any other provenance tool is that CPL does not enforce any required levels of granularity. Neither does it force a user to switch to using a specific tool, which makes the integration of the CPL into tools being developed (such as SensorDVC) fairly transparent to the user.

Sensor data is collected from ORNL's FRPs using a number of Campbell Scientific's data loggers. Apart from maintaining multiple backups, one of the final locations of the raw sensor data is a MySQL database populated by the Campbell Scientific's Loggernet Database (LNDB) tool.

ProvDMS connects to this data source and pulls raw data to present to the users of the system. LNDB automatically creates the required schema on the data server. ProvDMS is architected to sense the schema and its logical relationship to the FRP to present a cogent, simplified interface to the users. Illustrated in **Figure 3**, the sensor data is separated into *Stations*, each containing a set of *Data Loggers*. These *Data Loggers* consist of a set of data *Channels*. Physically these *Channels* relate to *Sensors* placed in different locations throughout the test facility. The ProvDMS system itself runs on a different server thereby providing complete separation of the raw data store and trace of the provenance of "experiments".

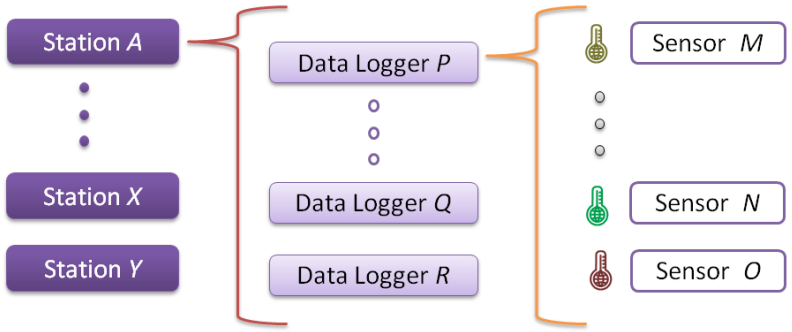


Figure 3 Logical layout of data which are grouped into *Stations*. Each *Station* has a set of *Data-Loggers* which consist of a number of *Sensors*. Each sensor serves a data *Channel* which goes to a MySQL database.

Architecturally, ProvDMS has a layered design and the different components interact cohesively as illustrated in **Figure 4**. The provenance tracking library, CPL, has been designed to be an independent, lightweight solution to provenance. Using CPL allows ProvDMS to act independently of provenance, calling application programming interface (API) hooks when information has to be saved to the provenance database. To interact with CPL, we built an abstraction layer to handle converting user actions to CPL API calls for inserting or querying provenance information. This is encapsulated into a compatibility layer and includes two wrappers: a PHP wrapper and a C++ wrapper. CPL, written in C, already includes some C++ functionality. Our C++ wrapper abstracts the interaction with CPL via a heavily object-

oriented interface.

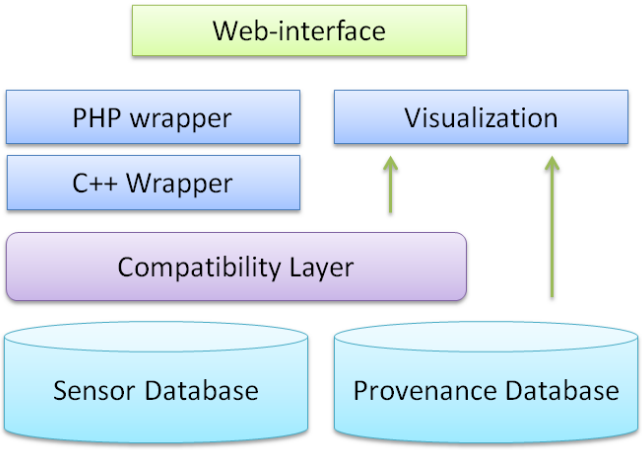


Figure 4 Diagram showing the layers and components of ProvDMS. The Compatibility Layer includes two wrappers: a PHP wrapper and a C++ wrapper that interacts with the PHP wrapper. The C++ wrapper abstracts the provenance back-end through the compatibility layer that interacts with the CPL interface to the provenance store.

TYPICAL WORKFLOW

This section reviews a typical workflow for the ProvDMS and SensorDVC software applications.

Data Collection and Management

ProvDMS (illustrated in **Figure 5**) has been built to be a one-stop access point for data related activities for ORNL's FRPs, including the collection and management of data. Users may browse and select subsets of *Stations*, *Data Loggers*, and *Channels* and define a temporal subset of selected data *Channels* into a new *Experiment*, which is saved on the server (cloud). This *Experiment* (data) can be exported by users at any time in the future. On creation, each *Experiment* is defined as a provenance object by the provenance back-end – creating all finer granularity objects in addition. The *Experiment* data participates in various analysis or modeling needs and sometimes undergoes transformations. Users can upload and define *Experiments* as derivatives of previous *Experiments*. This allows users to save the state of their data including any additional files or information in ProvDMS allowing them to trace the lineage in the future. ProvDMS also provides a data dashboard showing the status of different channels by the use of *sparklines* (Tufte 2004), shown in **Figure 6**. *Sparklines* are small trending plots that have no axes labels and may be embedded in line with text allowing concise visual communication allowing users to pick out trends in the data easily. We make use of *sparklines* to display the status of key channels from different sensors for quick assessment and detection of faults in the system.

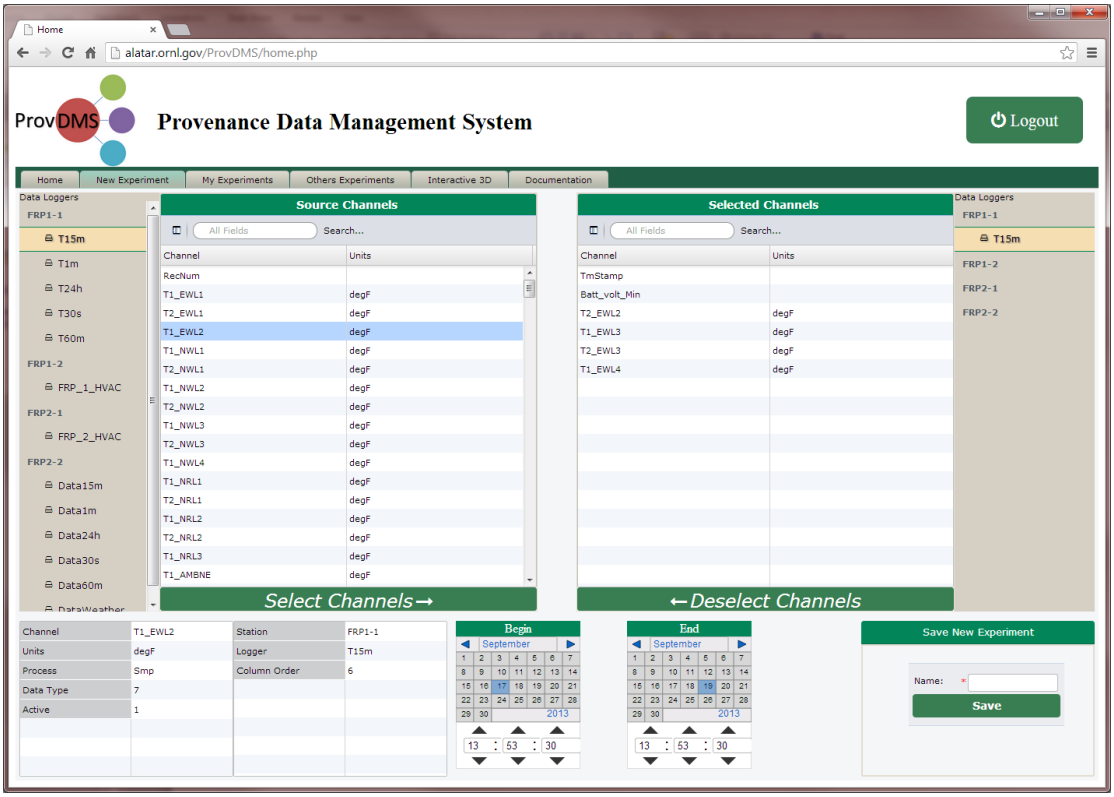


Figure 5 User interface of ProvDMS illustrating the data access/experiment creation interface. Users can select temporal subsets of data from *Channels* spanning different *Stations* and *Data Loggers*.

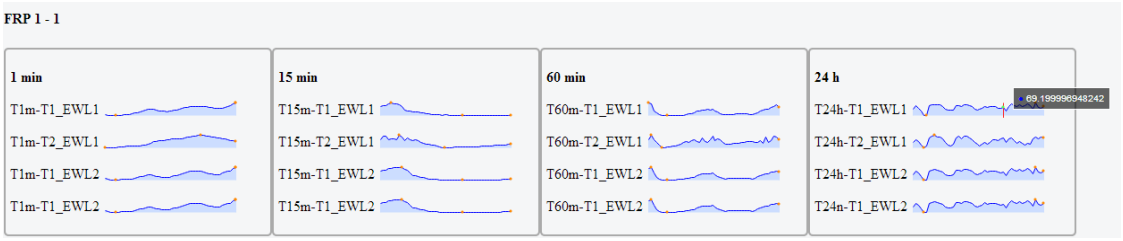


Figure 6 *Sparklines* are short graphs without axes labels that can be used to concisely present data trends. We use *sparklines* in our dashboard for visualizing the status of sensors.

Data Validation and Correction

Once a subset of data (i.e., *Experiment*) is extracted from ProvDMS, the SensorDVC application can be used to validate and correct the data. The SensorDVC application is currently composed of a main window (**Figure 7**), three validation dialogs, two correction dialogs, an original/corrected data view, original/corrected data visualizations, an actions log, a summary of dataset characteristics, and a dialog to view all original/corrected data side-by-side. The User starts using the SensorDVC application by importing the data set from ProvDMS. During the import, the parser component performs some simple corrections as issues are encountered. There are two cases that the parser is able to handle. One is null values in the data file. The other is gaps in the data (missing entire lines).

The specific action taken for null values depends on where in the file the null values appear. If the null values appear on the first line of data, then a single row of data points is inserted. The timestamp for that row is determined by subtracting time (depending on the timescale described in the file header section) from the next timestamp in the file. If null

values appear at the end of the file then a single row is inserted but the timestamp is determined by adding time to the last timestamp found. For null values found in the middle of the file, the last and next timestamps are used along with the timescale to determine the number of rows to insert. All inserted rows are populated with NaNs for their data points. There is often data present at the end of rows containing null values. The parser replaces inserted NaN values by working backwards from the end of each row that contains null values. Missing lines of data are detected by examining the last and next timestamps that were found. If the difference between them is greater than the current timescale, then an appropriate number of rows are inserted and populated with NaNs.

After importing data, validation is needed. This is accomplished through validation dialogs that validate data in different ways. The first validation dialog (**Figure 8**) asks the user for minimum and maximum values which are then used as validation limits for all data channels. The second validation dialog (**Figure 9**) allows the user to choose 1, 2, 3, 4, or 5 standard deviations which is used to determine minimum and maximum values for validation of all data channels. The third validation dialog allows the user to choose min/max or standard deviation for individual data channels.

Once the data has been validated, correction is needed for data points that have been flagged. This occurs through the correction dialogs. The first correction dialog (**Figure 10**) allows the user to choose a correction technique to be applied to all channels. The other correction dialog allows for the correction technique to be set for individual data channels. Data correction techniques include statistical (e.g., least squares, maximum likelihood estimation, segmentation averaging, and threshold based) and filtering (e.g., Kalman and LPC). Once correction is completed, a number of error metrics (relative, absolute, and root-mean-square error) are presented in the correction dialog's data view. The user can evaluate the efficacy of each correction technique by examining the error metrics. The lower the error, the better the prediction model and hence more accurate corrected data. The original/corrected data view (**Figure 7**) under the "Data" heading allows the user to select a channel from a drop-down box. The selected channel is then displayed in a table with original data next to corrected data. Data points which have been flagged by validation are displayed with a red font in the original data column. Corrected values are similarly shown in the corrected data column. The original and corrected data visualizations (**Figure 7**) under the "Data Plots" heading provide a total of 3 different viewing modes. The original data visualization offers flagged and range modes. Flagged mode displays invalid data points (based on validation) as red rectangles and valid data points as green rectangles. Range mode chooses each data point's color from a gradient (very light green to very dark green) based on the range of the channel. The colors for the range gradient were chosen based on (Borland and Taylor, 2007) and ColorBrewer (Brewer, 2009). The corrected plot offers corrected and range modes. Corrected mode is simply range mode but with the corrected data. NaN values are displayed as red by all modes. Zooming in on the visualizations is possible by selecting a range from the x- or y-axis. Zooming out is accomplished by clicking an appropriate zoom-out button. The actions log simply records the user's actions. The dialog for viewing all original/corrected data side-by-side (**Figure 7**) provides a somewhat spreadsheet-like view of the data. Additional functionality includes save/load session, write actions log to file, and write corrected data to file. Once the corrected data has been written to a CSV file, the file is uploaded to the ProvDMS application.

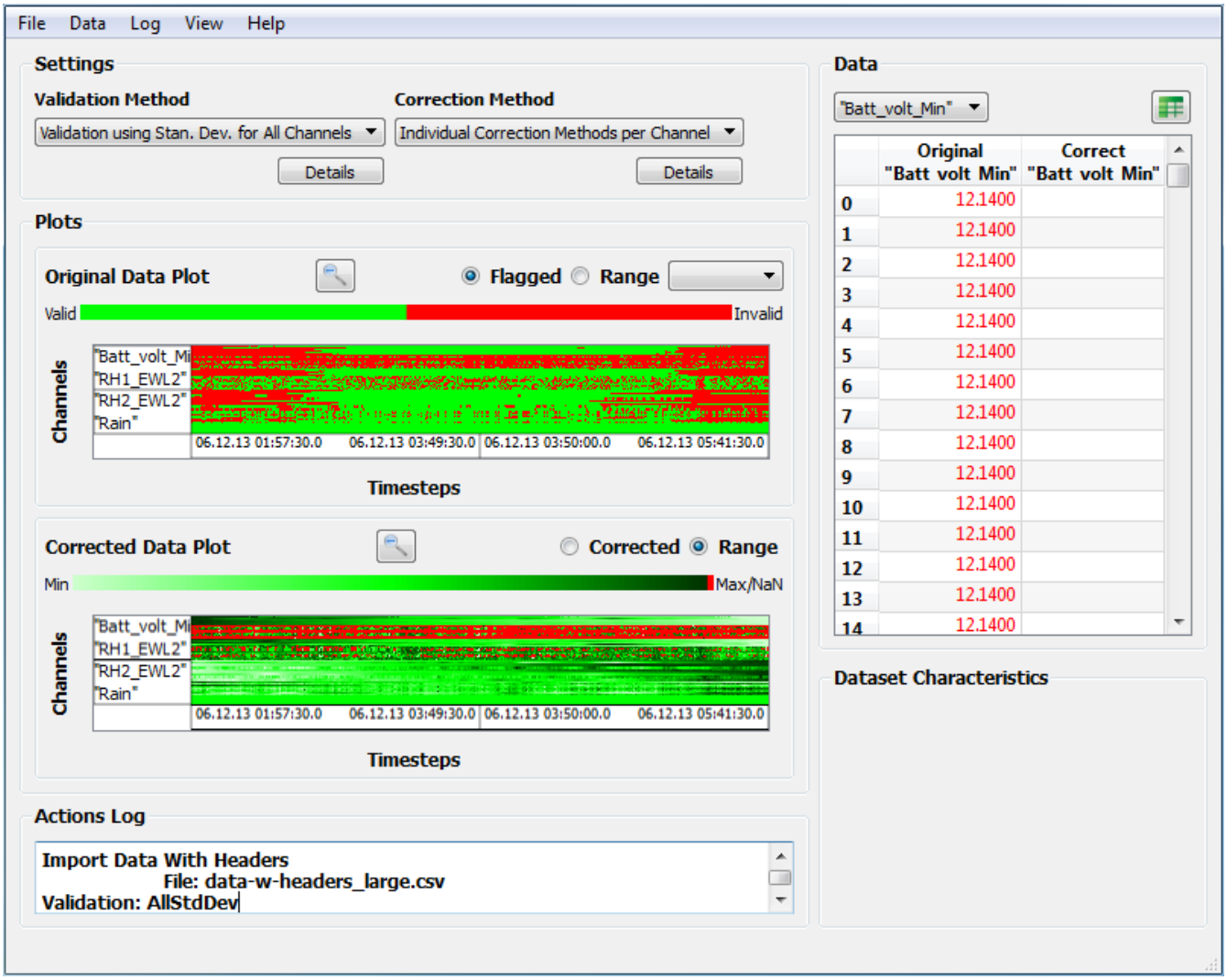
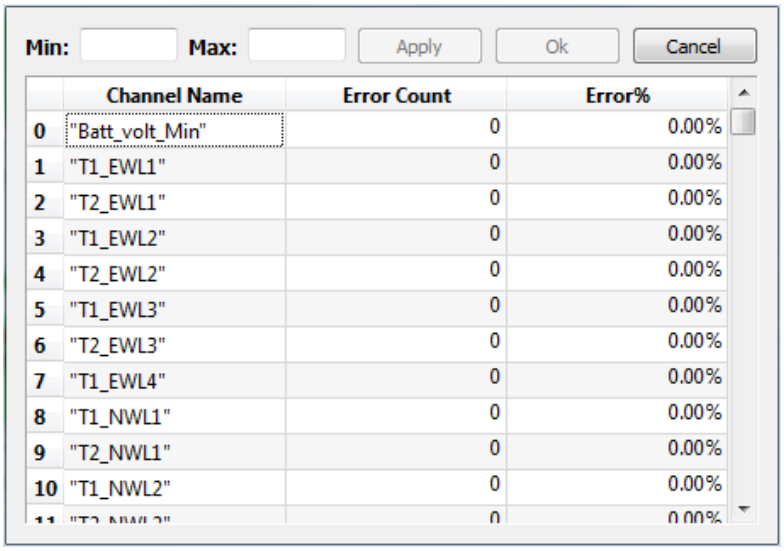
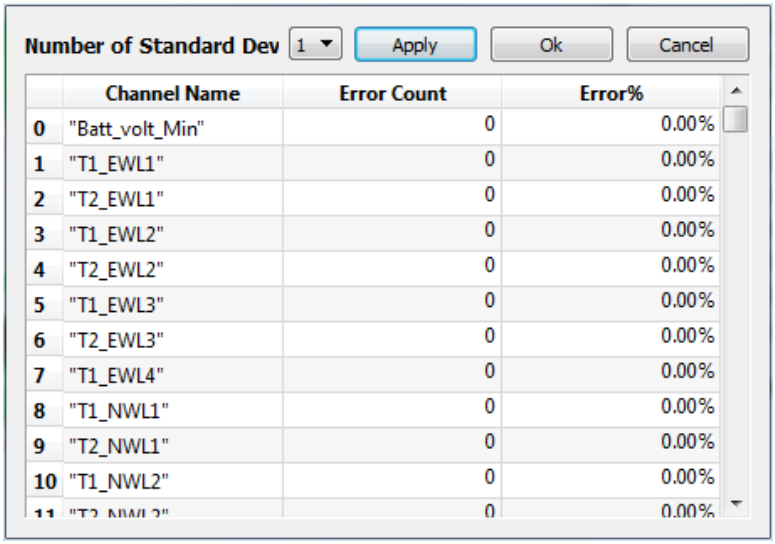


Figure 7 The SensorDVC application's main window.



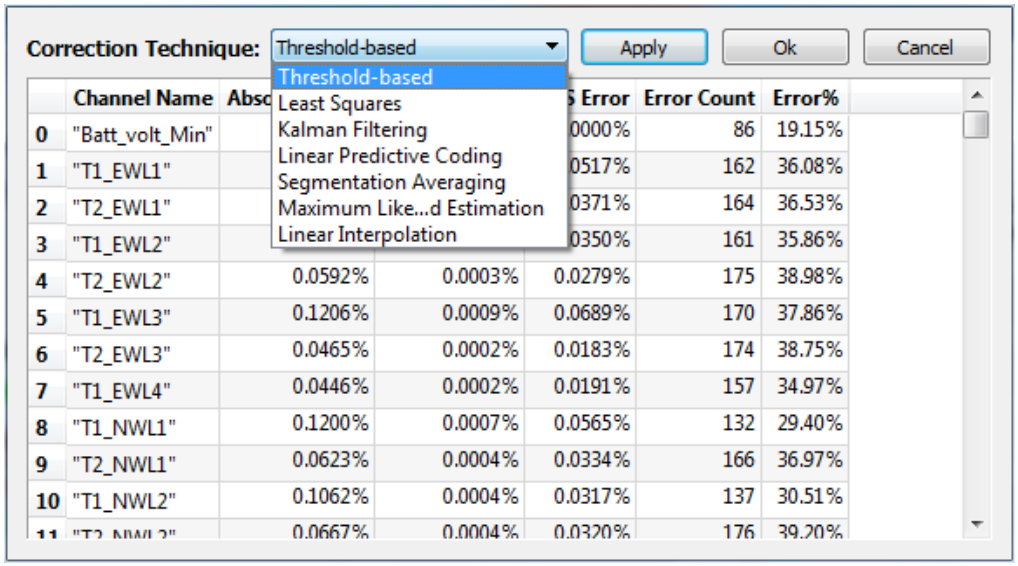
	Channel Name	Error Count	Error%
0	"Batt_volt_Min"	0	0.00%
1	"T1_EWL1"	0	0.00%
2	"T2_EWL1"	0	0.00%
3	"T1_EWL2"	0	0.00%
4	"T2_EWL2"	0	0.00%
5	"T1_EWL3"	0	0.00%
6	"T2_EWL3"	0	0.00%
7	"T1_EWL4"	0	0.00%
8	"T1_NWL1"	0	0.00%
9	"T2_NWL1"	0	0.00%
10	"T1_NWL2"	0	0.00%
11	"T2_NWL2"	0	0.00%

Figure 8 The validation dialog for minimum/maximum limits for all channels.



	Channel Name	Error Count	Error%
0	"Batt_volt_Min"	0	0.00%
1	"T1_EWL1"	0	0.00%
2	"T2_EWL1"	0	0.00%
3	"T1_EWL2"	0	0.00%
4	"T2_EWL2"	0	0.00%
5	"T1_EWL3"	0	0.00%
6	"T2_EWL3"	0	0.00%
7	"T1_EWL4"	0	0.00%
8	"T1_NWL1"	0	0.00%
9	"T2_NWL1"	0	0.00%
10	"T1_NWL2"	0	0.00%
11	"T2_NWL2"	0	0.00%

Figure 9 The validation dialog for standard deviation limits for all channels.



	Channel Name	Absc		S Error	Error Count	Error%
0	"Batt_volt_Min"			0000%	86	19.15%
1	"T1_EWL1"			0517%	162	36.08%
2	"T2_EWL1"			0371%	164	36.53%
3	"T1_EWL2"			0350%	161	35.86%
4	"T2_EWL2"	0.0592%	0.0003%	0.0279%	175	38.98%
5	"T1_EWL3"	0.1206%	0.0009%	0.0689%	170	37.86%
6	"T2_EWL3"	0.0465%	0.0002%	0.0183%	174	38.75%
7	"T1_EWL4"	0.0446%	0.0002%	0.0191%	157	34.97%
8	"T1_NWL1"	0.1200%	0.0007%	0.0565%	132	29.40%
9	"T2_NWL1"	0.0623%	0.0004%	0.0334%	166	36.97%
10	"T1_NWL2"	0.1062%	0.0004%	0.0317%	137	30.51%
11	"T2_NWL2"	0.0667%	0.0004%	0.0320%	176	39.20%

Figure 10 The correction dialog for all channels.

Provenance of the Data

The provenance of the raw data and validated/corrected data is tracked through the ProvDMS application which provides visualization capabilities to allow users to explore their data's lineage. A contextual Node-Link tree is used to visualize the provenance in a hierarchical fashion. User interaction is used to highlight and expand different levels of the tree as well as providing additional contextual information if it exists. The visualization is illustrated in **Figure 11**.

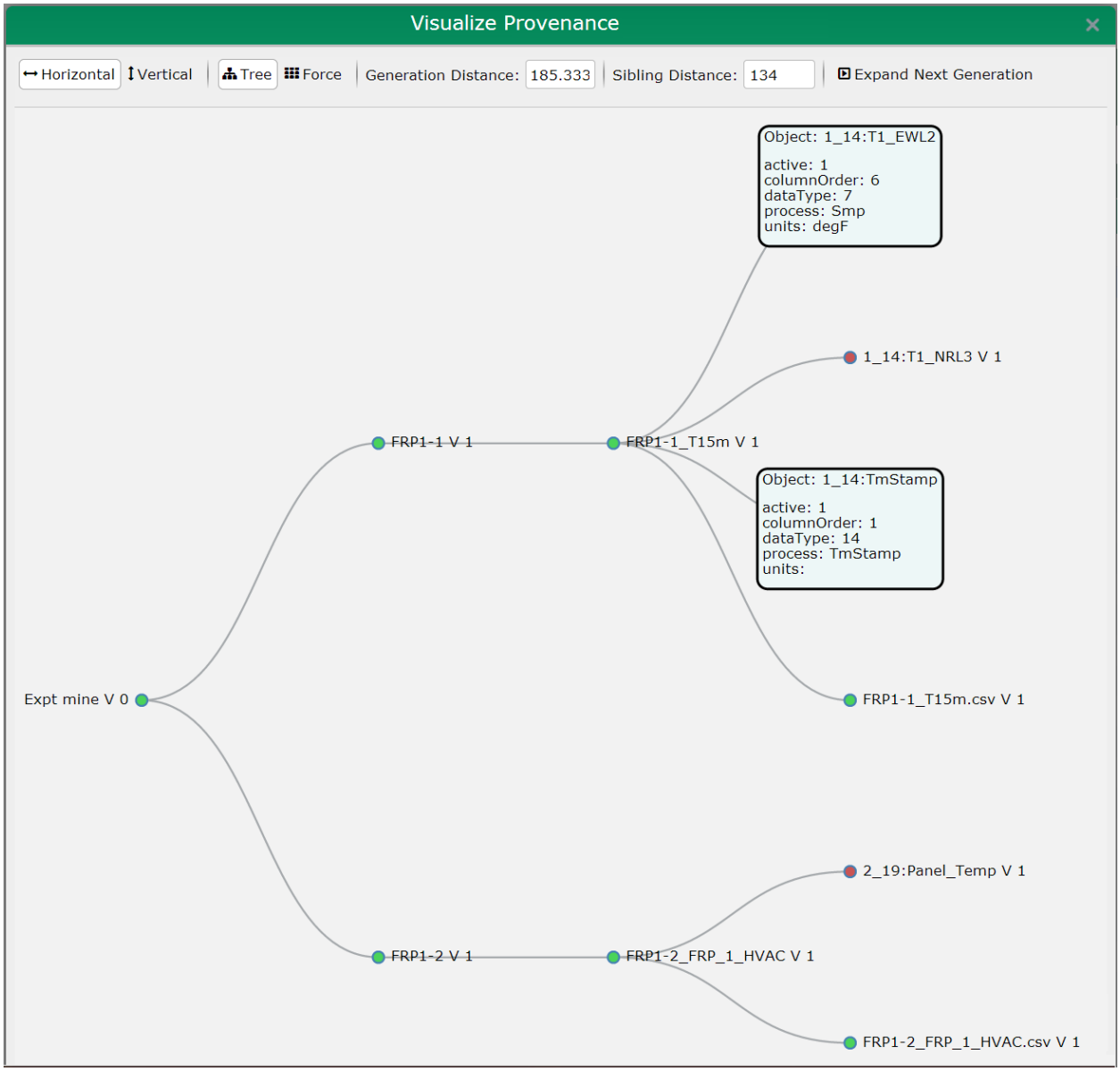


Figure 11 Contextual Node-Link Trees are used to visualize the provenance of different objects. User interaction is used to show contextual information and allow interactive exploration of the tree.

CONCLUSIONS AND FUTURE WORK

Data management, provenance, completeness, and accuracy are areas that must be addressed due to the large amounts of data being generated from sensors, especially in the realm of building technologies research. This paper targets these areas of research by producing the cloud-based Provenance Data Management System (ProvDMS) for data management and provenance and the Sensor Data Validation and Correction (SensorDVC) desktop application for data completeness and accuracy. These two software packages allow users to effectively collect and manage data while ensuring the data's accuracy and completeness from missing or corrupt data due to sensor failure, fouling, drifting, calibration error, or data logger failure. Furthermore, the knowledge of the provenance, or data *lineage*, is recorded to help users in quantifying the source and derivations of the data, transformations, repeatability of workflows on similar (newer) data, and in providing

holistic meta-information of the scientific process.

Future work regarding ProvDMS and SensorDVC includes the possible integration of both software applications via web-services and the compiled Core Provenance Library (CPL), which makes it easy to integrate into applications. This would allow users to directly access data in the SensorDVC application from ProvDMS, validate and correct the data, and upload to ProvDMS. The modified dataset will be added to lineage of the original dataset. Real-time visualization of the data can also be added to give users a better understanding of where the data originated and how it has changed over time. An example of is shown in **Figure 12**. The ProvDMS and SensorDVC software applications can also be modified to meet data management, quality assurance, and provenance needs in other applications besides building technologies.

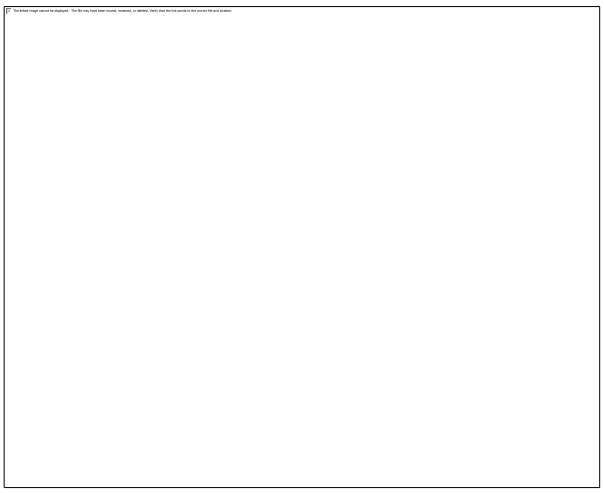


Figure 12 Visualization of a model of the 'medium-office' Flexible Research Platform (FRP) in Google Earth. Potential future work includes illustration of real-time data on interactive web-based 3D models of the FRPs.

ACKNOWLEDGMENTS

This work was funded by fieldwork proposal RAEB006 under the Department of Energy Building Technology Activity Number EB3603000. We would like to thank Edward Vineyard for his support and review of this project. Oak Ridge National Laboratory is managed by UT-Battelle, LLC, for the U.S. Department of Energy under contract DE-AC05-00OR22725. This manuscript has been authored by UT-Battelle, LLC, under Contract Number DEAC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

REFERENCES

Borland, D., and R.M. Taylor. 2007. Rainbow color map (still) considered harmful. *IEEE Computer Graphics and Applications* 27(2):14-17.

Brewer, C.A. 2009. ColorBrewer, Version 2.0. http://colorbrewer2.org/.

Castello, C.C. and J.R. New. 2012. Autonomous correction of sensor data applied to building technologies utilizing statistical processing methods. *Energy Informatics*, Atlanta, Georgia.

Castello, C.C., J.R. New, and M.K. Smith. 2013. Autonomous correction of sensor data applied to building technologies using filtering methods. submitted to *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Austin, Texas.

DOE. 2010. *Building Energy Data Book*. http://buildingsdatabook.eren.doe.gov.
DOE. 2012. EnergyPlus Energy Simulation Software. http://energyplus.gov.
DesignBuilder Software Ltd. 2005. DesignBuilder. http://www.designbuilder.co.uk/.
Foster, I. T.. 2003. The virtual data grid: A new model and architecture for data-intensive collaboration. *IEEE Scientific and Statistical Database Management* 3:11-11.
Foster, I., J. Vockler, M. Wilde, and Y. Zhao. 2002. Chimera: A virtual data system for representing, querying, and automating data derivation. *IEEE 14th International Conference on Scientific and Statistical Database Management*, pp. 37-46.
Frolik, J., M. Abdelrahman, and P. Kandasamy. 2001. A confidence-based approach to the self-validation, fusion and reconstruction of quasi-redundant sensor data. *IEEE Transactions on Instrumentation and Measurement*. 50(6): 1761-1769.
Hughes, P. Light commercial building flexible research platforms. 2012. Oak Ridge National Laboratory report ORNL/TM-2012/143.
Ibarguengoytia, P.H., L.E. Sucar, and S. Vadera. 2001. Real time intelligent sensor validation. *IEEE Transactions on Power Systems*. 16(4): 770-775.Macko, P. and M. Seltzer. 2012. A general-purpose provenance library. *4th USENIX Workshop on the Theory and Practice of Provenance*.
ZEBRAlliance. 2012. ZEBRAlliance: An alliance maximizing cost-effective energy efficiency in buildings. http://www.zebralliance.org.
Muniswamy-Reddy, K.K., D.A. Holland, U. Braun, and M.I. Seltzer, 2006. Provenance-aware storage systems. *USENIX Annual Technical Conference*, pp. 43-56.
NREL. 2012. OpenStudio: Commercial buildings research and software development. http://openstudio.nrel.gov.
Oinn, T., M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver, K. Glover, M.R. Pocock, A. Wipat, and P. Li. 2004. Taverna: A tool for the composition and enactment of bioinformatics workflows. *Bioinformatics* 20(17):3045-3054.
Postolache, O.A., P.M.B.S. Girao, J.M.D. Pereira, and H.M.G. Ramos. 2005. Self-organizing maps application in a remote water quality monitoring system. *IEEE Transactions on Instrumentation and Measurement*. 54(1):322-329.
Qt Project. 2013. Qt, Version 5.1. http://www.qt-project.org/.
Simmhan, Y. L., B. Plale, and D. Gannon. 2005. A survey of data provenance techniques. Computer Science Department, Indiana University, Bloomington IN, Technical Report IUB-CS-TR618.
Simmhan, Y. L., B. Plale, and D. Gannon. 2008. Karma2: Provenance management for data-driven workflows. *International Journal of Web Services Research (IJWSR)* 5(2):1-22.
Smith, M.K., C.C. Castello, and J.R. New. 2013. Sensor validation with machine learning. *IEEE International Conference on Machine Learning and Applications (ICMLA'13)*, Miami, Florida.
TVA. 2010. Campbell Creek energy efficient homes project. http://www.tva.gov/campbellcreekresearchhomes/.
Tufte, E. 2004. Sparklines: Theory and practice.Edward Tufte forum. http://www.edwardtufte.com/.
Uluyol, O., K. Kim, and E.O. Nwadiogbu. 2006. Synergistic use of soft computing technologies for fault detection in gas turbine engines. *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*. 36(4):476-484.
Vázquez-Salceda, J., S. Alvarez, T. Kifor, L.Z. Varga, S. Miles, L. Moreau, and S. Willmott. 2008. EU provenance project: An open provenance architecture for distributed applications. *Agent Technology and e-Health*, pp. 45-63.