

Sensor lifecycle management using scientific workflows

Derik Barseghian¹, Daniel Crawl², Matthew B. Jones¹, Ilkay Altintas², Jing Tao¹, and Sean Riddle³

¹ National Center for Ecological Analysis and Synthesis, University of California Santa Barbara

² San Diego Supercomputer Center, University of California San Diego

³ University of California Davis

{barseghian, jones, tao}@nceas.ucsb.edu, {crawl, altintas}@sdsc.edu, swriddle@ucdavis.edu

Abstract—Sensor networks are increasingly being deployed to create field-based environmental observatories. As the size and complexity of these networks increase, many challenges arise including monitoring and controlling sensor devices, archiving large volumes of continuously generated data, and the management of heterogeneous hardware devices. This paper presents the Kepler Sensor Platform, an open-source, vendor-neutral extension to a scientific workflow system for full-lifecycle management of sensor networks. This extension addresses many of the challenges that arise from sensor site management by providing a suite of tools for monitoring and controlling deployed sensors, as well as for sensor data analysis, modeling, visualization, documentation, archival, and retrieval. An integrated scheduler interface has been developed allowing users to schedule workflows for periodic execution on remote servers. We discuss and evaluate the scalability of periodically executed sensor archiving workflows that automatically download, document, and archive data from a sensor site. We conclude by discussing and comparing the Kepler Sensor Platform to related software.

Keywords—sensor network; scientific workflow; data discovery; data preservation; data analysis; quality assurance

I. INTRODUCTION

Automated sensing is increasingly used within field-based environmental sciences that traditionally used much more labor-intensive processes to collect data. In addition to the well-known, large-scale observatory programs (e.g., the National Ecological Observatory Network), individual graduate students, technicians, postdoctoral fellows, and faculty are increasingly specifying, deploying, maintaining, and managing sensor networks consisting of tens to thousands of sensors. These individual researchers face all of the management burdens that these complex, technological systems engender, but have few open software choices available to use in facing these burdens.

Some of the challenges that arise include: 1) the need to manage large volumes of data on a continuous basis; 2) quality assurance analysis for these data streams; 3) archival of both the raw data streams and quality-corrected derived data products; 4) visualization of the data; 5) monitoring of large

collections of sensors spanning multiple vendors, each with their own vendor-specific control software; and, 6) control and configuration of these sensors that span vendors. For typical scientific users that have minimal background in technology and programming, these challenges impede their ability to deploy and utilize small to large-scale sensor networks, and therefore limit the effectiveness of these systems for environmental science.

Vendor-neutral tools that assist the user-scientist throughout the lifecycle of sensor data are needed for designing, configuring, deploying, managing, and consuming data from these networks, as well as for monitoring and controlling the deployed sensor networks. Such management tools need to be able to manage sensor networks in many different deployment topologies, and manage and visualize both small and large deployments across sensor manufacturers.

Scientific workflow systems [1], [2] provide tools for authoring, executing, documenting, and archiving analysis and modeling processes. Tools such as Kepler [3] can be used to model many data processing tasks in an intuitive way by visually depicting the graph of steps in any scientific analysis. In previous work, Barseghian et al. [4] showed that scientific workflow systems like Kepler could be used to conveniently access sensor data from common sensor network middleware platforms such as DataTurbine [5]. However, this approach only partially solves the challenges facing scientists trying to manage sensor networks; complete solutions would address management of the full lifecycle of a sensor network, spanning both the systems engineering aspects of the lifecycle (e.g., network design, deployment, configuration, inventory, monitoring, and visualization) and the scientific use aspects of the lifecycle (e.g., data stream consumption, quality assurance, analysis, modeling, documentation, archiving, and visualization).

The major contributions of this work are to describe and evaluate the Kepler Sensor Platform, an extension of a scientific workflow system for full-lifecycle management of sensor networks. The work demonstrates the utility of the workflow system for graphical sensor site management, visualization, and analysis, as well as end-to-end management of sensor infrastructure, from sensors to data archives. The system provides a vendor-neutral client-side sensor management application to handle the sensor engineering lifecycle, and a suite of analysis, modeling, and visualization

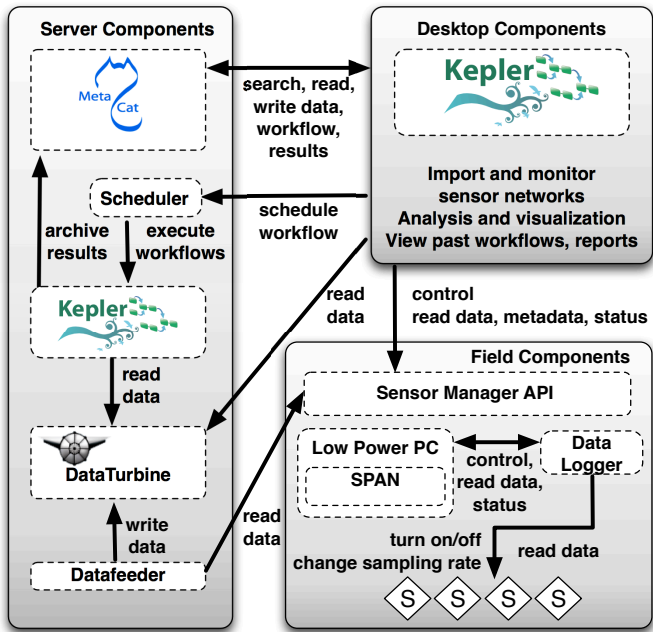


Figure 1. Architecture of the Kepler Sensor Platform. Users interact with sensors using the desktop Kepler application where they can import, layout, and visualize sensor networks, monitor the status of sensors, and get real-time data visualizations. From Kepler, they can schedule archiving and quality assurance workflows that periodically process data from the sensor network, provide metadata, and archive segmented snapshots of the data in the Metacat data archive.

tools to handle the scientific lifecycle. Together, these subsystems integrate sensor management with scientific analysis and modeling systems via the workflow paradigm in a visually intuitive and extensible manner.

In section II, we describe the systems across the lifecycle of sensor data, including the systems engineering aspects of the engineering lifecycle and the use of sensor data in the scientific lifecycle. We evaluate the scalability of the system for typical sensor loads in section III, discuss related work in section IV, and discuss conclusions and future work in section V.

II. LIFECYCLE OF SENSOR DATA

To effectively manage sensor networks, we designed the Kepler Sensor Platform to provide features targeting both the systems engineering portion of the lifecycle, focused on design, deployment, and monitoring of sensor networks, and the scientific usage portion of the lifecycle, focused on access to sensor data for analysis, modeling, and visualization. Figure 1 shows the main components of the Kepler Sensor Platform, including Field Deployed Components (directly interfacing with sensors), Server Deployed Components (to provide archival and automated processing systems), and Desktop Components (that provide Kepler as a client user interface to the other system components). Communication with the Field Deployed Components is handled through a Sensor Manager interface; this abstraction supports different types of hardware from various vendors. The Sensor Manager communicates

with a SPAN (Sensor Processing and Acquisition Network) server that provides drivers and a control interface for each of the sensors in the network [6]. Each of these components is used in both the engineering lifecycle and the scientific lifecycle of sensor data. For example, the server deployment includes components to transfer data from the field Sensor Manager to a DataTurbine server, and a Workflow Scheduler to manage and execute workflows on a Kepler execution engine, which is used to execute a workflow that segments the sensor data and metadata from DataTurbine and archives these to a Metacat data repository [7].

A. Engineering Lifecycle (Sensor Site management)

To manage sensor networks, scientists need to be able to design, inspect, monitor, and control suites of sensors deployed in the field. The Kepler Sensor Platform supports these functions through a client-side graphical interface to visualize a sensor deployment site as a workflow using the Kepler GUI (Figure 2). Hardware components such as sensors, and dataloggers can be dragged-and-dropped onto the canvas and connected to one another to represent the actual hardware configuration. Users can provide metadata such as make, model, location, and firmware for each of the hardware components. The canvas may also be annotated with lines, shapes, and text to further document the deployment site. This can be used to convey contextual information about a site, for example to depict spatial layout of sensors, experimental treatments, relevant geographic features, and obstacles like locked gates. Further, an engineering workflow can be exported to KML and viewed in Google Earth to display a satellite view of site components.

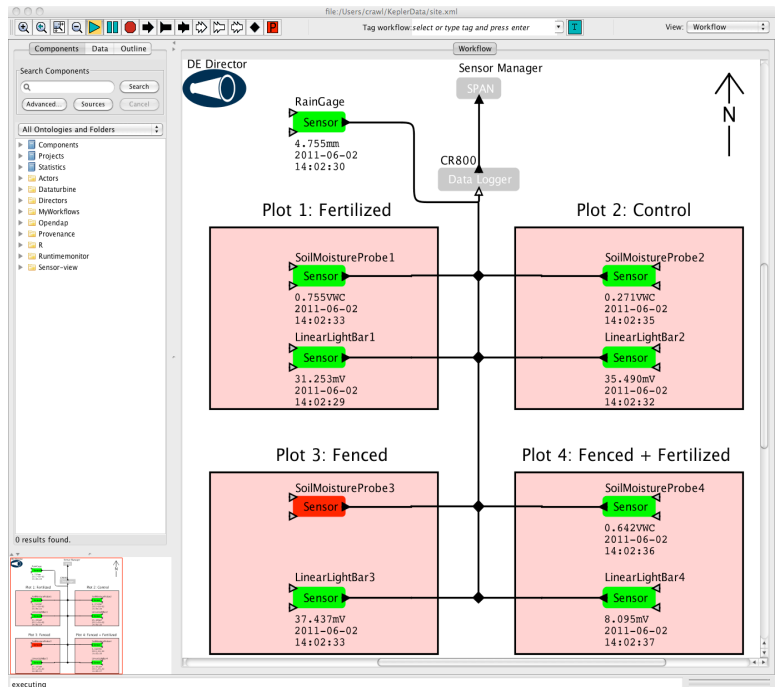


Figure 2. A sensor deployment site in the Kepler Engineering View. Nine sensors connected to a datalogger depict their relationship to real-world experimental plots. Right-clicking on sensors allows their metadata to be viewed and edited. Current sensor values display below the icons, and sensors that are inoperable are shown in red.

Importing sensor sites. To efficiently create engineering workflows, the sensor site description may be imported from a Sensor Manager. In this case, the user need only provide the URL of a Sensor Manager. The Kepler Engineering View queries the hardware descriptions from the Sensor Manager and automatically populates the canvas with components representing the site. The user may then add annotations or edit existing metadata parameters.

Sensor Monitoring. In addition to describing a sensor network site, the Kepler Engineering View provides an interface to monitor deployed hardware components. When an engineering workflow is executed, the Kepler Client queries the sensors' status from the Sensor Manager. As shown in Figure 2, the icons for each hardware component change color based on their status: green for on, red for off, blue for changing parameters, etc. Additionally, for each active sensor, live data values and their associated timestamps are displayed below the icon. Live data may also easily be plotted to show changes over time and to compare data from different sensors.

Sensor control. The Kepler Engineering View allows users to turn a sensor on or off, or change its sampling rate. As described previously, a user can also edit a sensor's metadata parameters. Additionally, a sensor may be controlled using its sensor actor in a scientific workflow (Figure 3). In Kepler, actors read inputs, perform a task, and write outputs. Actors can be connected so that the output of one is read in as input to another. A sensor actor may accept two inputs: sampling rate and a boolean value indicating if it should be active. When a sensor actor executes, it reads these inputs and, if the values have changed, communicates them to the Sensor Manager, which in turn enacts the changes at the site. If active, a sensor actor also outputs the last data value sampled through its output port. This is a powerful feature that allows users to design workflows to monitor sensor values and control sensor sites. A workflow can make changes to sensors based on previous data values from the same or other sensors, creating feedback loops that can be used for adaptive, event-driven sampling. Such adaptive workflows may also contain more involved analyses, e.g., comparison of live data against archived datasets.

Sensor data archiving. Archiving data from sensor networks can be tedious and data loss is difficult to avoid. One challenge is that data collection is continuous, which stresses existing systems that are more transaction oriented. The Kepler Sensor Platform system solves this problem by providing a server-side temporary storage buffer (DataTurbine, an open-source streaming middleware application that provides network ring-buffers for data storage [5]) to reliably accumulate sensor observations and multiplex data from all sensors at a site. The Sensor Manager stores sensor data to DataTurbine as a reliable, short-term cache of the data.

DataTurbine's ring-buffer is necessarily finite in size, so the data must also be archived for permanent long-term storage. For long-term storage, the Kepler Sensor Platform segments each of the data streams into a consistent size, generally based on temporal or spatial windows, generates detailed metadata describing that segment of data, and archives the segment in a Metacat server [7]. Metacat provides a federated storage

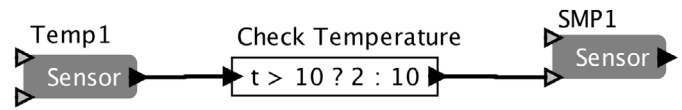


Figure 3. A scientific workflow that contains two sensor actors. In this example workflow, the sampling rate of one sensor is changed based on the output of another. The sensor Temp1 measures the temperature, and SMP1 measures soil moisture. When the temperature measured by Temp1 rises above 10 C, the sampling rate of SMP1 is decreased to 2 Hz; when the temperature goes below 10 C, the sampling rate is increased to 10 Hz.

solution for the Knowledge Network for Biocomplexity (KNB) data federation, and can be used as Member Nodes in the DataONE network [8], thereby making it easy for users to connect the Kepler Sensor Platform to national and international data federation initiatives.

This archival process is accomplished by executing a Kepler workflow that can inspect the sensor site metadata to determine appropriate archiving intervals, connection parameters, and other necessary metadata. The archival workflow compares the currently available data against previously archived data segments, and when appropriate intervals have been reached, automatically downloads, documents, and archives a new segment of the data. A metadata document is created for each data segment. This document is an instance of Ecological Metadata Language (EML) [9], and provides information to describe the data segment, such as sensor and site name, geographic location, temporal period of data collected, measurement units, and a link to access the data. The metadata document also contains a SensorML [10] description of the sensor metadata, such as the device type and manufacturer, and other relevant sensor metadata. The data segments and associated metadata documentation are stored in data packages on the Metacat server. These packages can later be searched for and retrieved with Kepler and data tools like Morpho [11], and via the web.

Workflow scheduling. A workflow scheduler was created to automate the process of archiving sensor data periodically. A user can specify the start time, end time, and execution interval at which the archival workflow should be run. This schedule is passed to a remote Scheduler Server, which will trigger the Kepler Workflow Run Engine to execute the archival workflow to segment, document, and store the sensor data to Metacat. Users can search for and retrieve the archives in the Metacat server through the standard data search interface in Kepler.

Kepler also has a general workflow scheduler, which can be used to execute any workflow periodically. This allows sensor site administrators to automate the execution of, e.g., QA/QC workflows at an appropriate frequency.

B. Scientific Lifecycle (Sensor Data Usage)

To be useful within the scientific lifecycle of sensor data, a sensor management tool ideally provides powerful analysis, modeling, and visualization capabilities. Kepler provides hundreds of analysis and modeling functions, ranging from atomic signal and image processing functions to integration [3]. For example, to accomplish a quality assurance analysis within Kepler, one can use the sensor actor to feed a stream of data in real-time from a sensor, connect this to the R system to use R's excellent time series analysis tools to detect anomalies, and

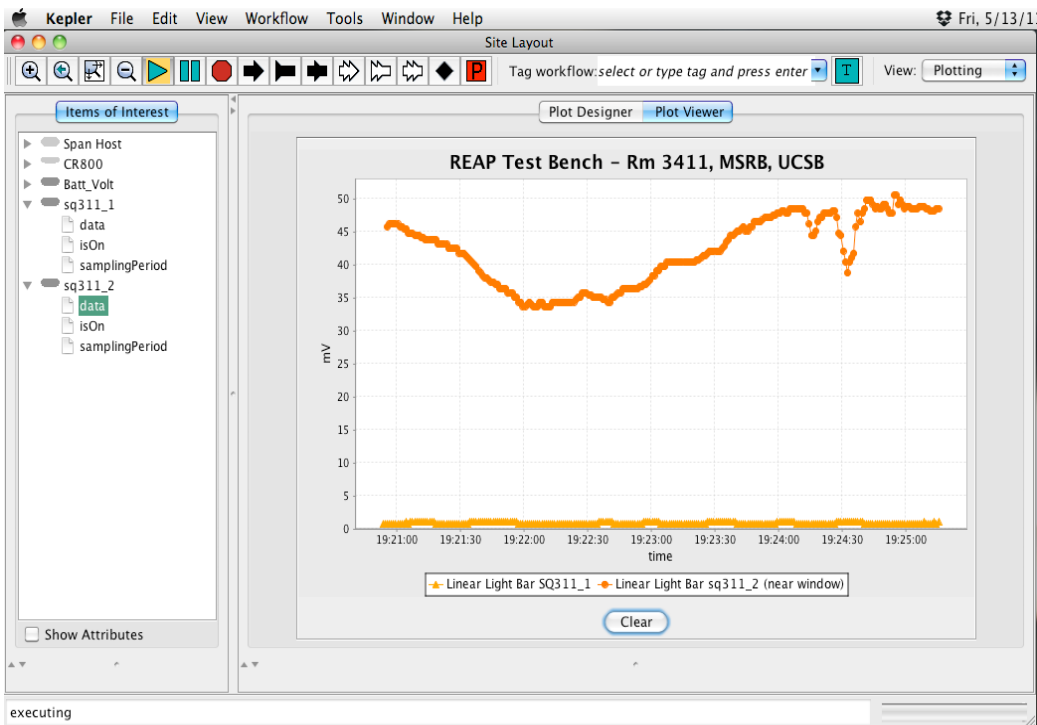


Figure 4. Kepler displays near real-time plots of sensor values over time, allowing scientists to quickly get a sense of the status of data collection at a site. When sensors are configured (e.g., to increase sampling rate), the results are immediately visible.

then feed the results of that analysis to Matlab for visualization. One could also annotate the data as the processing occurs, and output the resultant derived data set as a new data product, and save this in a data repository such as Metacat. In addition to real-time data access, Kepler provides tools to access the archived data sets from the engineering lifecycle, and a wide variety of other data from repositories around the world. Scientists can therefore combine data from historical periods with the data from real-time streams to detect changes in data trends over time and space. This flexibility allows scientists to mix and match the best analytical tools for the job while Kepler handles all of the orchestration and connectivity among the components. The end result is a workflow that fully documents the entire process that was employed to filter, transform, and analyze sensor data. In addition to these standard capabilities, we have added some additional features to Kepler specifically to help manage the scientific lifecycle of sensor data: real-time plotting, workflow scheduling for remote execution, and workflow run management.

Real-time plotting. The Kepler Engineering View provides the capability to plot live sensor data (Figure 4). The plotting view allows users to configure multiple plots and choose which sensors' data to display in each plot. A single plot can be configured to show the data from multiple sensors, allowing visual comparison of live data in near real-time. Useful plot interactions such as zooming, auto-range, labeling of title and

axes, adjusting point shapes and colors, clearing data, and exporting to static image files are supported.

Scheduling and remote execution. The same scheduling subsystem that handles periodic data archiving workflows from the engineering lifecycle can be used in the scientific lifecycle to periodically run analyses and models as needed by the scientist. When scheduling a workflow to execute remotely, the scientist can choose the time period for the executions and the interval at which the workflow should be re-run. For sensor data that is being continuously generated, this is extremely useful to periodically produce statistical summaries, generate or update summary plots for display on websites, and run forecast and hindcast models.

Workflow run management. By allowing scheduled workflows to be run on remote servers, it can be difficult to track how many times a workflow has run, and for each run whether it succeeded or failed with particular error conditions. The Kepler Workflow Run Manager provides an interface to browse through all of the workflow runs that were executed on a local or remote instance of Kepler. A complete provenance record of each workflow run is recorded, and the Workflow Run Manager provides a graphical view of these runs. Runs can be tagged in order to cluster related runs together, and they can be searched based on the provenance metadata (e.g., to find all runs for a temperature data archiving workflow that were run after June 21, 2011). From the Workflow Run Manager, one can also open the workflow as it was when executed, view any reports that were generated, and save a workflow run from a local instance of Kepler to a remote repository for backup or to share with colleagues.

III. EVALUATION

A sensor simulator was created to simulate different types of sensor network deployments and to aid implementing and testing Engineering View components. The simulator provides a virtual sensor network. Configuration parameters include the number of sensors, sampling rates, and sensor metadata such as make, model, location, etc.

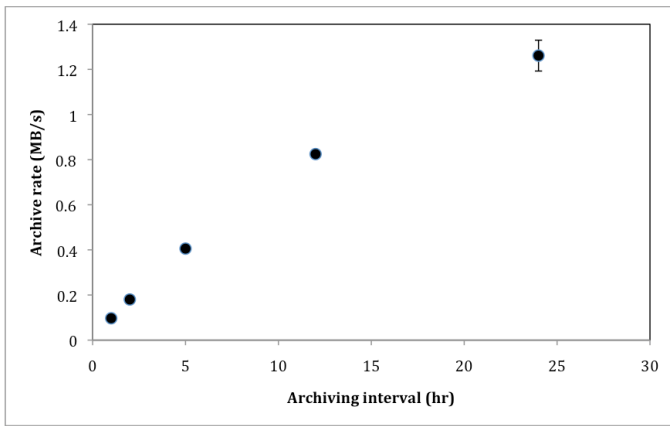


Figure 5. The archival rate as a function of archiving interval. The archival rate is the ratio of data size to the archiving workflow execution time.

To evaluate the performance of archiving sensor data, we measured the execution time of the archival workflow processing different amounts of data. We first ran the sensor simulator configured with one hundred sensors, each generating one sample per second (1 Hz). The simulator executed this configuration for 1, 2, 5, 12 and 24-hour periods, storing generated data into a DataTurbine server. We then ran the archival workflow to retrieve the samples from DataTurbine, and to create and upload the datasets into Metacat. The DataTurbine server and archival workflow ran on an iMac with a dual-core 3.2 GHz Intel CPU and 4 GB of memory. Metacat ran on an iMac with a dual-core 2.8 GHz Intel CPU and 2 GB of memory.

Figure 5 shows the archival rates for the different amounts of sensor data collected. The archival rate increases almost linearly with the sensor data. While the rate slows down for the 24-hour interval, we believe this shows good scalability when running the Kepler Sensor Platform server-side components on desktop hardware for the typical scale of sensor networks that we expect at the single laboratory level.

The archival workflow was executed three times for each archiving interval, and each point in the graph was calculated by averaging the execution times. For all intervals except for the 24-hour period, less than 3% standard error was observed. In the latter case, we believe the larger variability is due to occasional retries by the Metacat client uploading the dataset. The client attempts to transfer the entire dataset at once, and for larger sizes, retrying is more costly. We are planning to update the client to address this issue.

IV. RELATED WORK

Although automated sensing is being widely used in environmental sciences, most of the existing technology focuses on data acquisition and analysis. However, as mentioned before, the sensor lifecycle also includes stages that relate to the engineering and health monitoring of a sensor network. There are commercial solutions targeting vendor-specific technologies and protocols, e.g., LabVIEW, LoggerNet and Simulink, which can be utilized for particular infrastructures. A drawback of commercial solutions is they are often too specific, hard to extend, or costly for small-scale

scientific projects. In addition, there are open source initiatives like the Osiris and OOSTethys projects that are using the Sensor Web Enablement (SWE) specifications to build interoperable sensor webs. We will now discuss three commercial systems, followed by three open-source systems. To the best of our knowledge, our Kepler workflow-based solution is the only freely available open-source system that is vendor-independent, customizable, and extensible, allowing users to connect to, monitor, and control field-deployed hardware in an environment that also supports sophisticated statistical and modeling operations.

Campbell Scientific’s LoggerNet [12] provides data access, monitoring and control for large datalogger networks. However, along with being a proprietary solution, LoggerNet does not allow complex analysis and models to be run on data values, and it does not allow automatic sensor network adaptation as a result of such analysis. In addition, LoggerNet does not document and archive data packages into a repository such as Metacat.

National Instruments LabVIEW [13] has a graphical user interface that integrates block diagrams with a dashboard interface. With its extensive hardware support that involves Field Programmable Gate Arrays (FPGAs), microprocessors and special purpose digital signal processors (DSPs), LabVIEW is a very versatile environment for custom data acquisition and analysis. However, LabVIEW is also proprietary software, and provides no archival capabilities.

Simulink [14] provides sensor platform support via the Simulink Coder (formerly Real-Time Workshop), targeting specific sensor hardware and architectures. Simulink Coder can generate embedded source code from Simulink diagrams and MATLAB scripts. Generated code can be used for real-time and other applications, including rapid development, simulation optimization, and testing with hardware in the loop. Although it provides some functionality needed for field-deployed hardware, the process for code generation and deployment can be cumbersome and requires specific target language compiler programs that are proprietary.

SensorKit [15] is an open source platform for sensor network management and data archival. As with the Kepler Sensor Platform, SensorKit uses SPAN to interface with dataloggers for data acquisition and device management. Data is archived to a SensorBase Database, and while this includes a web-based interface to graph, share, and export data, it does not provide sophisticated analysis and modeling capabilities.

The Viptos toolkit [16], derived from Visual Ptolemy, is an open-source system that is similar to Simulink Coder. Viptos models TinyOS-based wireless sensor networks via a graphical development and simulation system. TinyOS is an event-driven runtime environment used to build wireless sensor networks. Viptos builds on Ptolemy to enable the creation of flow diagrams, which are then used to create TinyOS programs from TinyOS components written in nesC, a programming language derived from C. Because Viptos is TinyOS-based, it has limited flexibility in sensor hardware choices. Nor does it provide documentation and archival features for sensor data streams.

The OSIRIS project [17] has developed a demonstration system for management of in-situ sensing data using the Sensor Web Enablement (SWE) suite of specifications from the Open Geospatial Consortium. For example, they developed web-based applications that use the Sensor Observation Service (SOS) [18] to access observations data from field-deployed sensors, and the Sensor Planning Service (SPS) for controlling sensors and sensor network components [19]. Similarly, the OOSTethys project has deployed SWE-based systems for managing sensors in ocean observing systems [20]. These and similar projects demonstrate the flexibility of OGC standards for accessing and controlling sensors, but do not provide graphical network visualization, data access, and control features within an application that supports sophisticated analysis and modeling. The union of standardized sensor network access with analysis and modeling tools as provided by Kepler would significantly strengthen these approaches.

V. CONCLUSION

We have described an extension to the Kepler scientific workflow system that supports full-lifecycle management of sensor networks. This extension addresses the needs of a wide audience, from technicians interested in monitoring and adjusting a site to keep it functioning effectively; to scientists that want to conduct complex analyses on sensor data streams, or compose workflows that will intelligently adapt a site's configuration in real-time in response to events of interest. The Kepler Sensor Platform supports scheduling QA/QC workflows to be run periodically on remote servers, provides an easy to use plotting view for quick comparisons of live data streams, and provides functionality for documenting, archiving, and retrieving sensor data into and from long term archives. Our tests have shown this extension effectively handles sites with many sensors, each sampling at a high frequency. Our work is entirely open-source, and thus may be utilized and extended by anyone with an interest. Future work will focus on interoperability with the Sensor Web Enablement suite of standards from the OGC.

ACKNOWLEDGMENT

This work was conducted as part of the REAP project with funding from the National Science Foundation (Grant # 0619060) and was supported by the National Center for Ecological Analysis and Synthesis (NCEAS), a Center funded by NSF (Grant #EF-0553768), the University of California, Santa Barbara, and the State of California.

REFERENCES

- [1] B. Ludäscher, I. Altintas, S. Bowers, J. Cummings, T. Critchlow, E. Deelman, D. D. Roure, J. Freire, C. Goble, M. Jones, S. Klasky, T. McPhillips, N. Podhorszki, C. Silva, I. Taylor, and M. Vouk, "Scientific Process Automation and Workflow Management," In *Scientific Data Management: Challenges, Technology, and Deployment*, Computational Science Series, A. Shoshani and D. Rotem, eds. chapter 13, 2009.
- [2] Y. Gil, E. Deelman, M. Ellisman, T. Fahringer, G. Fox, D. Gannon, C. Goble, M. Livny, L. Moraey, and J. Myers, "Examining the challenges of scientific workflows", *Computer*, vol. 40 (12), pp. 24-32, 2007.
- [3] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger-Frank, M. Jones, E. Lee, J. Tao, Y. Zhao, "Scientific Workflow Management and the Kepler System," *Special Issue: Workflow in Grid Systems, Concurrency and Computation: Practice & Experience*, vol. 18, no. 10, pp. 1039-1065, 2006.
- [4] D. Barseghian, I. Altintas, M.B. Jones, D. Crawl, N. Potter, J. Gallagher, P. Cornillon, M. Schildhauer, E. Borer, E.W. Seabloom, P.R. Hosseini, "Workflows and extensions to the Kepler scientific workflow system to support environmental sensor data access and analysis," *Ecological Informatics*, vol. 5, pp. 3-8, 2010. doi:10.1016/j.ecoinf.2009.08.008
- [5] S. Tilak, P. Hubbard, M. Miller, and T. Fountain, "The Ring Buffer Network Bus (RBNB) DataTurbine Streaming Data Middleware for Environmental Observing Systems", in *Proc. eScience*, 2007, pp.125-133.
- [6] Ye W, Silva F, DeSchon A, Bhatt S. 2008. Architecture of a satellite-based sensor network for environment observation. In: *NASA Earth Science Technology Conference (ESTC2008)*, Adelphi, MD, June 24-26, 2008. California, USA. University of Southern California, Information Sciences Institute: http://www.esto.nasa.gov/conferences/estc2008/papers/Ye_Wei_A8P2.pdf
- [7] M.B. Jones, C. Berkley, J. Bojilova, M. Schildhauer, "Managing Scientific Metadata," *IEEE Internet Computing*, vol. 5, no. 5, pp. 59-68, 2001.
- [8] W. Michener, D. Vieglais, T. Vision, J. Kunze, P. Cruse, G. Janée, "DataONE: Data Observation Network for Earth — Preserving Data and Enabling Innovation in the Biological and Environmental Sciences," *D-Lib Magazine*, vol. 17, no. 1/2, 2011, doi:10.1045/january2011-michener.
- [9] E. Feagraus, S.J. Andelman, M.B. Jones and M. Schildhauer, "Maximizing the Value of Ecological Data with Structured Metadata: An Introduction to Ecological Metadata Language (EML) and Principles for Metadata Creation," *Bulletin of the Ecological Society of America*, vol. 86, no. 3, pp. 158-168, 2005.
- [10] A. Robin, S. Havens, S. Cox, J. Ricker, R. Lake, H. Niedzwiedek: *OpenGIS sensor model language (SensorML) implementation specification*. Technical report, Open Geospatial Consortium Inc. (2006)
- [11] D. Higgins, C. Berkley, M.B. Jones, "Managing Heterogeneous Ecological Data using Morpho," *Proceedings of the 14th International Conference on Scientific and Statistical Database Management*, July 24-26, 2002. J. Kennedy (ed). ISBN 0-7695-1632-7 ISSN 1099-3371.
- [12] *LoggerNet website*. <http://www.campbellsci.com/loggernet>
- [13] *LabVIEW website*. <http://www.ni.com/labview>
- [14] *Simulink website*. <http://www.mathworks.com/products/simulink>
- [15] F. Silva, A. Deschon, J. Chang, S. Westrich, Y.H. Cho, S. Gullapalli, T. Benzel, E.A. Graham, "SensorKit: A Flexible and Extensible System for In-Situ Data Acquisition," in *American Geophysical Union, Fall Meeting 2009*.
- [16] E. Cheong, E.A. Lee, and Y. Zhao. "Viptos: A Graphical Development and Simulation Environment for TinyOS-based Wireless Sensor Networks," *Demo Abstract, Proceedings of the Third ACM Conference on Embedded Networked Sensor Systems (SenSys 2005)*, San Diego, California, USA, November 2-4, 2005.
- [17] OSIRIS: Open architecture for Smart and Interoperable networks in Risk management based on In-situ Sensors. <http://www.osiris-fp6.eu/>
- [18] G. McFerren, D. Hohls, G. Fleming, T. Sutton, "Evaluating Sensor Observation Service implementations," *Geoscience and Remote Sensing Symposium, 2009 IEEE International, IGARSS 2009*, vol.5, pp.V-363-V-366, 12-17 July 2009. doi: 10.1109/IGARSS.2009.5417655
- [19] S. Jirka, A. Bröring, C. Stasch, "Discovery Mechanisms for the Sensor Web," *Sensors*, vol. 9, issue 4, pp. 2661-2681, 2009. doi:10.3390/s90402661
- [20] L. Bermudez, E. Delory, T. O'Reilly, J. del Rio Fernandez, "Ocean observing systems demystified," *OCEANS 2009, MTS/IEEE Biloxi - Marine Technology for Our Future: Global and Local Challenges*, pp.1-7, 26-29 Oct. 2009.