Kepler Sensor Platform

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

¹ National Center for Ecological Analysis and Synthesis, UCSB
² San Diego Supercomputer Center, UCSD
³ University of California, Davis

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

An open-source, vendor-neutral extension to the Kepler workflow system for full lifecycle management of sensor networks.
Kepler Sensor Platform Capabilities

Provides tools for

• monitoring and controlling sensors

• sensor data analysis, modeling, visualization, documentation, archival, and archive retrieval

• scheduling workflows (QA/QC, alerting, processing, archival, etc.) for periodic, remote execution
Kepler Sensor Platform Architecture
Desktop Components

Server Components
- MetaCat
- Scheduler
- Kepler
- DataTurbine
- Datafeeder

Desktop Components
- Kepler
- Import and monitor sensor networks
- Analysis and visualization
- View past workflows, reports

Field Components
- Low Power PC
- SPAN
- Data Logger
- Control, read data, status
- Turn on/off, change sampling rate

Interactions:
- Scheduler: execute workflows
- Kepler: read data, control data
- DataTurbine: read data, write data
- Datafeeder: write data
- Desktop Components: search, read, write data, workflow, results
- Field Components: read data
- Low Power PC: control, read data, status
- Data Logger: read data
Automatically generate Site Layout
Realtime Monitoring
Site Layout with context added
Sensor configuration
Sensor configuration

- isOn: 
- samplingPeriod: 1
- dataLogger: CR800
- sensorServer: gpp.msi.ucsb.edu
- coefficients:
- conversion-type: no conversion
- daq-method:
- latitude: 34.412291
- longitude: -119.842335
- measurement-unit: mV
- sampleMethod: average
- samples-per-measurement: 1
- sensor-make: Apogee Instruments
- sensor-measurement: Photosynthetic Photon Flux (PPF)
- sensor-model: SQ-311 (sun)
- serial-number:
- altitude: 0.000000

[Options: Cancel, Help, Preferences, Restore Defaults, Remove, Add, Commit]
Datalogger configuration

SPAN Server:
gpp.msi.ucsb.edu

Active Program:

'CR800 program to sample battery voltage, and 2 sq311 (apogee linear light sensors)
into table1 scans every 5s, samples every 30s.

'wiring:
'sq311_1 – red to sel1, black to power ground, clear to signal ground
'sq311_2 – red to sel6, black to power ground, clear to signal ground

'Declare Variables and Units
Public Batt_Volt
Public sq311_1
Public sq311_2

Units Batt_Volt=Volts
Units sq311_1=mV
Units sq311_2=mV

'Define Data Tables
DataTable(Table1,True,-1)
    Datainterval(0,30,Sec,10)
    Average(1,Batt_Volt,FP2,False)
    Average(1,sq311_1,FP2,False)
    Average(1,sq311_2,FP2,False)
EndTable

'Main Program
BeginProg
    Scan(1,Sec,1,0)
        'Default Datalogger Battery Voltage measurement Batt_Volt:
        Battery(Batt_Volt)
        'Generic Single-Ended Voltage measurements sq311_1
        VoltSe(sq311_1,1,mV2500,1,True,0,-60Hz,1,0,0,0)
        'Generic Single-Ended Voltage measurements sq311_2
        VoltSe(sq311_2,1,mV2500,6,True,0,-60Hz,1,0,0,0)
    'Call Data Tables and Store Data
    CallTable(Table1)
    NextScan

EndProg
Sensor as streaming data source
Workflows to Automate Site Control

This workflow changes the sampling rate of the sensor based on the data values.

NOTE: to run this workflow, the sensor simulator must be running. Instructions are here: https://kepler-project.org/developers/incubation/kepler-engineering-view-for-reap/running-sensor-simulator

Daniel Crawl, SDSC/UCSD, July 2010
Sensor Simulator

cd kepler.modules/senor-view/
ant sensorsim
Realtime Plotting
Plotting – Plot Designer
Plotting – Plot Viewer

REAP Test Bench – Rm 3411, MSRB, UCSB

- Linear Light Bar SQ311_1
- Linear Light Bar sq311_2 (near window)
Archival Workflow - DataTurbine to Metacat
kepler.modules/sensor-view/workflows/

This workflow archives streaming data from a DataTurbine server into a Metacat.
For each sensor, a datapackage is created. The workflow keeps track of what data have already been archived. DataTurbine channels must follow a naming convention, i.e. this workflow expects to be run against a DataTurbine server that is receiving data from SpanTodt (details here: TODO), It's intended one person schedule this workflow to be run periodically.
To schedule, use the Workflow Scheduler, from the Tools menu.

Configure these parameters:
Source DataTurbine:
- DataTurbineServerAddress: 'gpp.msi.ucsb.edu:3333'
- OnlyArchiveSpecificSensorIDs: true
- SensorName: ['CR800_sq311_2']
- DataLoggerName: ['CR800']
- SiteName: ['gpp']

Destination Metacat:
- EcoGridPutServiceURL: 'http://kepler-dev.nceas.ucsb.edu/kepler/services/PutService'
- EcoGridAuthServiceURL: 'http://kepler-dev.nceas.ucsb.edu/kepler/services/AuthenticationService'

PN Director
GetSensorIDs
GetLastArchiveInfo
GetArchivingTimeSpan
GenerateEndAndDateString
ToMillisecond: "1000"
DBConnectionURL: [driver = "org.hsqldb.jdbcdriver", password = "", url = "jdbc:hsqldb:file:"+property("KEPLERUSERDATA")+"DataturbineToMetacat", user = "sa"]
DateTimeStringFormat: "yyyy-MM-dd HH:mm:ss"
Ecogrid Writer
UpdateLastArchivingInformation
Cleanup
Schedule Sensor Data Archival
Select Sensors to be archived in the selected Repository at the specified Interval (sensors will be checked between the Start and End times):

<table>
<thead>
<tr>
<th>Archiving</th>
<th>Sensor</th>
<th>Interval</th>
<th>Interval Unit</th>
<th>Start</th>
<th>End</th>
<th>Data Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>Sensor1</td>
<td>Hours</td>
<td></td>
<td>MM/dd/YY hh:mm:ss</td>
<td>MM/dd/YY hh:mm:ss</td>
<td>localRepository</td>
</tr>
<tr>
<td>✔️</td>
<td>Sensor2</td>
<td>Hours</td>
<td></td>
<td>MM/dd/YY hh:mm:ss</td>
<td>MM/dd/YY hh:mm:ss</td>
<td>localRepository</td>
</tr>
<tr>
<td>✔️</td>
<td>Sensor3</td>
<td>Hours</td>
<td></td>
<td>MM/dd/YY hh:mm:ss</td>
<td>MM/dd/YY hh:mm:ss</td>
<td>localRepository</td>
</tr>
<tr>
<td>✔️</td>
<td>Sensor4</td>
<td>Hours</td>
<td></td>
<td>MM/dd/YY hh:mm:ss</td>
<td>MM/dd/YY hh:mm:ss</td>
<td>localRepository</td>
</tr>
<tr>
<td>✔️</td>
<td>Sensor5</td>
<td>Hours</td>
<td></td>
<td>MM/dd/YY hh:mm:ss</td>
<td>MM/dd/YY hh:mm:ss</td>
<td>localRepository</td>
</tr>
<tr>
<td>✔️</td>
<td>Sensor6</td>
<td>Hours</td>
<td></td>
<td>MM/dd/YY hh:mm:ss</td>
<td>MM/dd/YY hh:mm:ss</td>
<td>localRepository</td>
</tr>
<tr>
<td>✔️</td>
<td>Sensor7</td>
<td>Hours</td>
<td></td>
<td>MM/dd/YY hh:mm:ss</td>
<td>MM/dd/YY hh:mm:ss</td>
<td>localRepository</td>
</tr>
<tr>
<td>✔️</td>
<td>Sensor8</td>
<td>Hours</td>
<td></td>
<td>MM/dd/YY hh:mm:ss</td>
<td>MM/dd/YY hh:mm:ss</td>
<td>localRepository</td>
</tr>
<tr>
<td>✔️</td>
<td>Sensor9</td>
<td>Hours</td>
<td></td>
<td>MM/dd/YY hh:mm:ss</td>
<td>MM/dd/YY hh:mm:ss</td>
<td>localRepository</td>
</tr>
<tr>
<td></td>
<td>Sensor10</td>
<td>Hours</td>
<td></td>
<td>MM/dd/YY hh:mm:ss</td>
<td>MM/dd/YY hh:mm:ss</td>
<td>localRepository</td>
</tr>
</tbody>
</table>
Dataset for sensor "sensor0" at site "gpp"

Data Set Citation
REAP.Datset for sensor "sensor0" at site "gpp" for time period "2011-02-23 01:21:36" and "2011-02-23 01:23:34".

Data Tables, Images, and Other Entities:
Metadata download: [Ecological Metadata Language (EML) File]

Data Table:
Name: gpp_CR800_sensor0_2011-02-23_01:21:36-2011-02-23_01:23:34
Description: Dataset for sensor "sensor0" at site "gpp" for timeperiod between "2011-02-23 01:21:36" and "2011-02-23 01:23:34"

Online Distribution Info:
Download File: eecognd://knb/doc/129849749729279.1

Physical Structure Description:
Object Name: gpp_CR800_sensor0_2011-02-23_01:21:36-2011-02-23_01:23:34
Size: 2225 bytes
Character Encoding: ASCII
Number of Header Lines: 1
Record Delimiter: #x0A
Text Format: Maximum Record column
Length:
Simple Delimited: Field Delimeter: #x20
Number Of Records: 61

Attribute(s) Info:
- **date** - calendar date of each temperature measurement record, Type of Value: Format YYYY, Precision: 1 day
- **time** - Greenwich Mean Time of each temperature measurement record, Type of Value: Format hh:mm:ss, Precision: 1 second
- **data** - atmospheric temperature, Type of Value: float, Unit: celsius, Type: real

Data Set Owner(s):
Organization: REAP
Web Address: http://reap.ecoinformatics.org/
Abstract:
This metadata record describes the dataset for sensor "sensor0" located at site "gpp" for the period between "2011-02-23 01:21:36" and "2011-02-23 01:23:34".

Keywords: [NCEAS]
<table>
<thead>
<tr>
<th>Files</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dcd_mgr, cr1k_d, cr10x_d, crio_d, sdi12_d, vaisala_wxt_d, webcam_d</td>
<td>SPAN binaries</td>
</tr>
<tr>
<td>cr_parser.py</td>
<td>Script that converts native datalogger program into SPAN configuration file</td>
</tr>
<tr>
<td>/etc/init.d/span.sh</td>
<td>Script to start/stop SPAN server (dcd_mgr and cr1k_d)</td>
</tr>
<tr>
<td>rbnb-3.2b5.jar</td>
<td>DataTurbine</td>
</tr>
<tr>
<td>/etc/init.d/dataturbine.sh</td>
<td>Script to start/stop DataTurbine</td>
</tr>
<tr>
<td>sensorview-utils-1.0.jar</td>
<td>JAR containing SpanToDT and MirrorDT (created with ant from within sensor-view Kepler module)</td>
</tr>
<tr>
<td>/etc/init.d/spanTodt.sh</td>
<td>Script to start/stop SpanToDT</td>
</tr>
</tbody>
</table>
SPAN
(Sensor Processing and Acquisition Network)
Developed by ISI (Information Science Institute – Marina Del Rey, California)/USC (University Southern California)

Written in C
LGPL

ISI’s SVN no longer available, we host a fork with many bug fixes and additions:
https://code.ecoinformatics.org/code/span/trunk
SPAN TCP Ports for clients

55055 control
55056 saved data and metadata
55058 live data
55059 live metadata

# watch data stream
telnet gpp.msi.ucsb.edu 55058

# issue commands
telnet gpp.msi.ucsb.edu 55055
list-channels CR800 all

2011-12-08T03:02:01.692Z OK: Channels:
CR800_Batt_Volt,CR800_sq311_1,CR800_sq311_2
Server Components

- **Server Components**
  - MetaCat
    - Scheduler
      - archive results
      - execute workflows
    - Kepler
      - read data
    - DataTurbin
      - write data
    - Datafeeder
      - read data

- **Desktop Components**
  - Kepler
    - search, read, write data, workflow, results
    - schedule workflow
    - read data
    - control
      - read data, metadata, status
    - Field Components
      - Low Power PC
        - SPAN
          - control, read data, status
          - turn on/off
          - change sampling rate
      - Data Logger
        - read data

- **Field Components**
  - Datafeeder
  - DataTurbin
  - Scheduler
  - Kepler
  - Low Power PC
  - Data Logger

Import and monitor sensor networks
Analysis and visualization
View past workflows, reports
Server Components

MetaCat

- Storage for Data, Sensor Data, Workflow KARs, and Workflow-Run KARs

Workflow Scheduler (webservice)

- Clients schedule workflows for remote execution

Kepler Run Engine (webservice)

- Headless Kepler
- Invoked by Workflow Scheduler

DataTurbine

Datafeeder ("spanToDT")
Datafeeder (spanTodt)
Transfer data from SPAN to DataTurbine

# create sensorview-utils-1.0.jar (contains SpanToDT and MirrorDT)
cd kepler.modules/sensor-view
ant sensorview-utils-jar

# start
/etc/init.d/spanTodt.sh start
Acknowledgements

https://kepler-project.org  http://reap.ecoinformatics.org

This material is based upon work supported by the National Science Foundation under award 0619060.

The National Center for Ecological Analysis and Synthesis, a Center funded by NSF (Grant Number 0072909), the University of California, and the UC Santa Barbara campus.

Collaborators:

NCEAS (UC Santa Barbara); UC Davis; UC San Diego (San Diego Supercomputer Center); CENS, University of California, Los Angeles; OPeNDAP; Oregon State University