

Building a Scalable Sensor Management Framework

Bill Huynh and Arno Puder

Background

NetBEAMS, funded by Agilent Technology, is a generic data acquisition framework developed by the collaboration between Computer Science Department of San Francisco State University, Agilent Technologies and Sun Microsystems. NetBEAMS (Networked Bay Environmental Assessment Monitoring System) is a sensor network application that use JDDAC (Java Distributed Data Acquisition and Control) framework, created by Agilent Technologies, to process and transport measurement data from sensors to a public JDDAC data server. NetBEAMS is currently being used by Romberg Tiburon Center in Tiburon, CA for monitoring sea water in San Francisco Bay.

Requirements

In environmental monitoring, important goals of collecting data include immediate access to the data and generation of interpretive products. Different environmental groups will most likely have their own data collecting system and data server. The challenge with data being on different servers is that the data are not easily retrieved and shared because each data server will probably have its own way of providing the data, e.g. JDDAC servers provide data using NRSS and CSV files through HTTP request. NRSS (Numeric Really Simple Syndication) is an extension of RSS which uses XML to syndicate the most recent sensor measurement's data and metadata. CSV, a text file which can be opened by excel application, is a method that JDDAC servers use to transport archived data. To provide a clean view of the data, such as posting on a web site or plotting a graph, we created tools that read NRSS and CSV files from JDDAC servers. And to be able to integrate our data with data from other servers, we created more tools to read data from those servers. Besides being able to read data from many different servers, we have to provide the data in many different formats, e.g. web site, OpenDAP, wireless phone, etc... However, the problem is not building this system because one can build this system if given enough time. The challenge is to build a system flexible enough for many applications that can be utilized by many organizations. We introduce Sensor Management

Framework, an open source framework that will allow the users to build a sensor data management system using existent components.

Sensor Management Framework (SMF)

The most important feature in SMF is sharing code. To allow code sharing, SMF is designed based on plug-and-play architecture. A plug-and-play system is a system that is extendable by adding plug-ins created by someone else. The two main purposes of a sensor data management system are to read the data from data servers, e.g. JDDAC server, and then redistribute the data to some type of data consumers, e.g. Web site or OpenDAP. Therefore, SMF requires two types of plug-ins, a data reader plug-in and a data consumer plug-in. Data reader plug-ins are components whose only task are to read data from a specific server and data consumer plug-ins are components that take the data from the data readers and produce some meaningful view of the data. Since plug-ins need to be able to work together, they have to follow some specific constraints. The center of SMF is the SMF platform which is a component that connects the data consumers to the data readers. The SMF platform provides one standard application programming interface (API) which both the data readers and data consumers have to follow. The data reader can use any way to read data from any server but once the data are read, the reader has to pass the data to the SMF platform using the API. The data consumer can then use the API to retrieve the data from the SMF platform and produce some view of the data. The benefit of this architecture is that all the detail is encapsulated. The data consumers do not need to know where or how the data are pushed to the SMF platform and the data readers do not need to know how the data will be used by the data consumers. This way the plug-ins can be built independently by different people as long as they follow the SMF platform API. Another benefit of using SMF is that once we have all different types of plug-ins for data readers and data consumers contributed by the community, one can build a sensor data management system by simply downloading the plug-ins needed for their system.