Building Sensor Filter Grids: Information Architecture for the Data Deluge

Geoffrey C. Fox, Mehmet S. Aktas, Galip Aydin, Andrea Donnellan, Harshawardhan Gadgil, Robert Granat, Shrideep Pallickara, Jay Parker, Marlon E. Pierce, Sangyoon Oh, John Rundle, Ahmet Sayar, and Michael Scharber

***Abstract*—We discuss a general architectural approach to knowledge and information management and delivery in distributed systems. Our approach is based on the recognition that time-stamped, streaming information message units form the core of seemingly disparate systems that range from online sensors and scientific instruments to Web information retrieval. Globally distributable Grid services manage these information streams. Geographical Information System services provide exemplary realizations of this picture and may be used as a model for other scientific domains. With this unified architecture in place, we may begin to consider the problems of information integration as equivalent to sensor federation.**

***Index Terms*— distributed messaging, information systems, real time data processing, sensors**

Manuscript received March 15, 2006. This work was supported in part by the U.S. National Aeronatical and Space Administration's Advanced Information Systems Technology program.

G. C. Fox is with Indiana University Departments of Computer Science and Physics, Bloomington, IN 47401. (Phone: 812-856-7977; Fax: 812-856-7972; email: gcf@indiana.edu.)

M. S. Aktas is with Department of Computer Science and the Community Grids Laboratory, Indiana University Bloomington, IN 47404. (email: maktas@cs.indiana.edu).

G. Aydin is with the Department of Computer Science and the Community Grids Laboratory, Indiana University, Bloomington, IN 47404 (email: gaydin@cs.indiana.edu).

A. Donnellan is with the NASA Jet Propulsion Laboratory (email: Andrea.Donnellan@jpl.nasa.gov)

H. Gadgil is with the Department of Computer Science and the Community Grids Laboratory, Indiana University, Bloomington, IN 47404 (email: hgadgil@cs.indiana.edu).

R. Granat is with the NASA Jet Propulsion Laboratory (email: Robert.A.Granat@jpl.nasa.gov)

S. Oh is with the Department of Computer Science and the Community Grids Laboratory, Indiana University, Bloomington, IN 47404 (email: ohsangy@cs.indiana.edu).

S. Pallickara is with the Community Grids Laboratory, Indiana University, Bloomington, IN 47404 (email: spallack@cs.indiana.edu).

M. E. Pierce is with the Community Grids Laboratory, Indiana University Bloomington, IN 47404 (email: mpierce@cs.indiana.edu).

J. Rundle is with the Departments of Physics, Civil Engineering, and Geology of the University of California, Davis (email: jbrundle@ucdavis.edu)

A. Sayar is with the Department of Computer Science and the Community Grids Laboratory, Indiana University Bloomington, IN 47404 (email: asayar@cs.indiana.edu).

M. Scharber is with Scripps Orbit Permanent Array Center, San Diego, CA 92093-0225 (Email: mscharber@ucsd.edu)

I. Sensor Streams and Information Management

Automated knowledge and information management in networked systems has a wide range of applications. As we present in this paper, all these sources of information can be thought of as the equivalent of online sensor instruments generating time-stamped information. In this view, the generalized architecture for text based information retrieval and data streams that are produced by scientific instruments are fundamentally equivalent.

Before proceeding, we first clarify the definition of the term "sensor". In the normal scientific sense, a sensor is a device that automatically records information about its environment. For example, earth-bound Global Positioning System (GPS) sensors ("receivers") regularly record their current positions, as measured using accurate timing and positioning information from orbiting satellite arrays. At some fixed interval (say, once or twice per second), a GPS sensor will record its current position.

The so-called "data deluge" [1] results from the large numbers of sensor devices that are now Web enabled. High Energy Physics is an important example of massive experiments involving petabytes of data and thousands of scientists across the world [2]. The Large Hadron Collider is (in terms of our discussion) a very large sensor. Interferometric Synthetic Aperture Radar (InSAR) satellites are another example of a very high end scientific instruments potentially contributing to data deluge. Space-based InSAR instruments can be used to measure small scale deformations in the earth's surface over wide areas. This may be used to measure both pre-seismic and post-seismic motion near earthquake faults, which are undetectable by seismometers and may even be previously unknown. Proposed satellite missions may generate as much as 200 GB of data per day. Many more examples of the data deluge can be found in the literature on Grid computing [3][4]. However, to truly understand the data deluge, one needs to realize that much simpler, much less expensive sensor instruments are now also commonly network-enabled [5]. Similarly, audio-video collaboration systems (or simply web cameras) are excellent sources of real-time data streams [6].

Regardless of the sophistication of the sensor or instrument, we can identify some universal properties:

1. The sensor must be described with metadata: the type

of sensor it is, the type of data it generates, the frequency of the generated data, the location of the sensor, and so on;
2. The sensor's data must be time-stamped as it is published;
3. The data records may be archived permanently, available as they are published (i.e., real-time), or both; and
4. The data goes through several additional filtering operations that transform and analyze it.

This last point will be stressed in this paper. Filters obviously play an important role in transforming the original raw data of the sensor. Usually these filters are external to the sensor and also on the network, and so they may be thought of as sensors themselves. Filters are typically grouped into chains of dependent operations that produce specialized data products through a process commonly called "workflow".

Basic filter applications typically include simple transformation operations (which change the data formats) and predictable data extraction operations (which extract out known parts of the data stream). Data format converters may be relatively simple in the case of simple GPS-like data (which consists of only a few numbers), or they may be more computationally complex as in the case of codec converters in video streams [6].

Simple filters are characterized by the certainty of their output: the transformation should always take place in a predictable way, and the output rate is directly related to the input rate. More sophisticated filters interpret the data in some fashion and so will not necessarily possess the simple filter characteristics. These sophisticated filters include scientific applications that attempt to make forecasts based on the incoming data, image processors that can detect interesting events in monitoring cameras (such as the appearance of a person in a security camera), data mining and other adaptive applications, statistical analysis tools, and so on. Data mining on real-time data streams to detect interesting features is an area of important current research. We will discuss this in more detail in below for GPS sensors. For related approaches to weather forecasting and geospatial image processing, see Refs. [7] and [8], respectively.

It is important to realize that the sensor concept should also be applied to Web accessible information. Many approaches such as RDF [9] and OWL [10] have attempted to define ontological information (or, metadata) models for the Internet. Internet search techniques such as PageRank™ [11] have actually shown significantly more success in terms of scaling and (effectively) inferred metadata determination through text processing. In any case, the goal of these information system approaches is to help identify information of interest.

The sensor approach to Web information processing provides a higher level abstraction over Semantic Web and PageRank™ schemes. Here a sensor may be a particular collection of important Internet Web sites or the entire Internet itself: Google is (in our terms) a sensor. Information is may be passed through a series of filters (which themselves become sensors), transforming the information at each step. These filters may, for example, use Google services as their initial data source but may pass harvested information through, for example, RDF and OWL filters and perform higher level inference operations to refine the data. The tools based on Semantic Web representations thus become sensor filters.

The sensor filter approach provides an important additional capability over standard Internet search techniques: instead of relying on users to *pull* information that they want, the sensor filter approach *pushes* information to the user, delivering information that he or she may find interesting without the user directly making the request. When this is combined with topic-based publish/subscribe systems, we may create very focused, context-based information push that is tailored to specific end users.

Interestingly, the technique for publishing time-stamped information on the Internet has been around for several years but has taken some time to catch on. RSS feeds [12] and their cousins are now ubiquitous and form the basis of so-called "mash-up" Web pages that aggregate content from many different sources. As we will discuss below, RSS-like feeds are also quite amenable to expressing scientific sensor information.

II. Sensors and Grid Computing

We now turn to the subject of Grid computing and its relationship to sensors and filters. The key concept in Grids (messaging using SOAP) is directly related to real-time data produced by sensors and their filters.

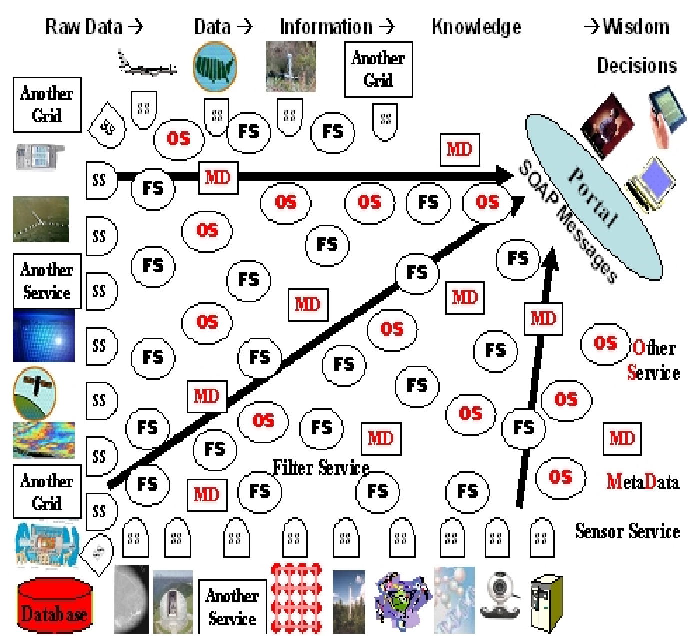


Figure 1. Vision of Sensor Filter Grids with filters (FS) shown from many fields, processing SOAP messages from sensors (SS), Grid(let)s and services controlled by meta-data (MD) and supported by other services (OS).

The Grid is built around streams supported by a powerful messaging system to achieve high interactivity and performance for distributed analysis. The approach is applicable to management and analysis of general distributed information resources including those found in data analysis as well as the documents and presentations associated with a scientific activity. Our approach exploits the multi-tier federated and hierarchical structure of most problems of the type sketched in Fig. 1.

Sensor Filter Grids can support the scenario of Fig.1 and correspond to a particular paradigm for building grids. One can define a particular Sensor Filter Grid paradigm to correspond to a particular workflow linking services together and produce the associated support for this workflow. Note that this figure uses a "Grid of Grids" context. The full Grid is made of sub-Grids, each of which is yet another sensor.

The Sensor Filter Grid system consists of the individual filter and information services, the portal as well as Grid systems support in areas like fault tolerance, security, notification and the workflow engine itself. Different instances of this paradigm correspond to different configuration parameters and to different choices of component application services in the workflow. The different instances share the overall paradigm of workflow, sensors and filters as well as general Grid system services.

Sensor Filter Grids are built around three distinctive features – a) information services that present data through traditional service interfaces; b) filters that accept data with these interfaces, transform them and present them with the same interfaces; and c) streaming connections between all services that provide on the fly archiving, high-performance transport, security and fault tolerance. A very common feature is that the filters are composable in a distributed, federated, hierarchical fashion. We suggest that scientific data analysis has this characteristic whereas for example distributed simulations would typically not be composable in this fashion.

Sensor Filter Grids are built from information services and filters that support identical service interfaces. The source, structure and processing of data in information services is opaque while filters transform or aggregate data from other filters or information services. Full filter services are built from basic filter services using application dependent dataflow. The filters in this Grid style are composed hierarchically: in information retrieval one can merge ranked lists to obtain a new ranked list while in statistical analysis, moments and histograms can easily be combined.

This composability of the filters supports natural scaling and federation algorithms and defines the Sensor Filter Grid structure. Note that superimposed on the tree of filters and information sources, one has a Grid of system services like security and reliability which are shared between the different applications. One supports this paradigm with an interactive administrative interface allowing system configuration, information sources and filters to be defined [13].

The system can be elegantly virtualized so that in specifying the information sources, one can either give very specific resources or Semantic Web style specifications for a discovery service. Of course this type of support is independent of the Sensor Filter Grids application area; one must just distinguish service specification from the location and number of services. Note that there is no implication that all filter actions are identical; one will design the messages on the data streams between filters so they are self describing. For example, in information retrieval, the filter services are designed to accept full documents (specified by a URI), titles, or simple metadata. This flexible construction allows filter trees to be built that don't depend on the number and exact features of information resources.

III. The Role of Streams

A critical feature of the architecture is the use of a powerful messaging infrastructure, NaradaBrokering [14][15], for linking between the services. This allows all services to be linked by managed reliable streams. NaradaBrokering can be deployed as a distributed set of brokers or as Axis handlers [19]. It supports the OASIS Web Service specifications for WS-ReliableMessaging [16], WS-Eventing and WS-Notification [17][18]. These capabilities allow fault tolerance and asynchronous messaging with publish-subscribe semantics [21] that allow user-sensitive push.

We are developing a sophisticated management environment that controls and monitors all streams in a Grid [20] and extends fault tolerance across streams, services and message brokers. The latter allows one to control the flow of data into filters so that there is no overflow. NaradaBrokering supports the subscription of redundant services to aid in fault tolerance. The Web Service messages flowing in NaradaBrokering can be archived at any link. This provides for dynamic caching to support system performance and is also used in message throttling. This NaradaBrokering archiving service resembles the replica management approaches developed for particle physics Grids [22][23][24][25][26]. The archive service is of course supported by a metadata catalog to manage it.

NaradaBrokering supports software multicast, and so it is straightforward to build collaborative sessions. NaradaBrokering has been successfully used for audio-video conferencing and other collaborative tools in the commercial Anabas product [27] and the open source GlobalMMCS project [28][29][30]. As described below, NaradaBrokering is also used for managing GPS sensor data streams [31] [32]. We deploy simple versions of the architectures proposed here with sensors, filters and data mining codes linked by real-time streams. These time sensitive applications have spurred the development of several interesting technologies including a metadata catalog that supported (with different implementations but the same interface) both scalable large scale access and low (a few tens of millisecond) latency access needed for interactive use by the relatively small number of services involved in an individual workflow.

We need to ensure that our architecture delivers streaming information as fast as possible between services while retaining the advantage of Grids and Web services. This can be achieved as all service messages are handled by the "system" through NaradaBrokering which formally one "binds" to SOAP as the transport. There are two important aspects that need to be optimized – the transport protocol and the representation of the information in the message. For the protocol we exploit NaradaBrokering's ability to support general protocols which can be chosen independently of the application with in a service model the last handler in a container choosing the protocol. UDP, TCP, parallel TCP, HTTP, and HTTPS are currently supported. NaradaBrokering can also link to network monitoring tools, and here we should study links to Caltech's MonALISA agent-based global monitoring and control system [35], which is widely deployed in the physics community.

Now we turn to the wire-representation used in messaging. Not surprisingly, transport of XML and SOAP messages encoded in conventional ASCII/Unicode "angle-bracket" representation is too slow for applications that demand high performance [36]. Several groups are developing ways of representing XML in binary formats for fast message exchange [36][37][38]. We are developing [39] a Web Service negotiation language for higher performance Web Services to negotiate both the protocol described above as well as the representation scheme such as choice between Fast and Binary XML. Initial negotiation is done using standard SOAP angle-bracketed messages to determine the supported representation and transport protocol capabilities. We will employ handlers to take care of the conversion and transport issues, which will make the negotiation and transport process transparent to the services. Once the services agree on the conditions of the data exchange, handlers convert XML data into an appropriate binary format and stream it over a high performance transport protocol using NaradaBrokering.

IV. A Generic Information Architecture

As stressed by Birman [40], Web services provide key low level capability but deliberately do not define an information or data architecture. This is left to domain specific specification activities such as CellML/SBML for biology, and GML for Geographical Information Systems (GIS). The Semantic Grid combines basic meta-data and service discovery capabilities with data-mining and reasoning. Here we generalize the architecture of Geographical Information Systems. Open GIS service specifications are defined by the Open Geospatial Consortium [34]. Specifications include a geography language syntax (GML) and ways to store, access, query, manipulate (via the Web Feature Service) and display (via the Web Map Service and its advanced versions) geographical features. Sensors and sensor webs have a further set of XML descriptions, collectively referred to as "Sensor Web Enablement" [41]. SensorML [42] is the most well known of these. In a service-oriented architecture, GIS corresponds to a domain specific XML language and a suite of services for the needed functions. We suggest that one can define a GIS-style information model in many application areas. This leads to concepts like Biological Information Systems, Military Information System, Service Infrastructure Information System for the overall systems, Physics Analysis Information Systems, an so forth.

We generalize our experience with GIS in a fashion shown in Fig. 2, defining an ASIS for Application Specific Information System. In each application, there are some services like discovery and notification that do not need to be made application specific. However we will need a generalization of the Web Feature Service from GIS that can be applied to domain specific data and metadata models. Operations of this service include search/store/access functions. We call this generalization Application Specific Feature Service, and all interfaces in Fig. 2 must use ASFS for domain specific data.

We also need to define the analogy of the Geography Markup language, which is a language expressing the domain specific features. We call this Application Specific Language (ASL). One will need a set of basic tools, termed Application Specific Tools and Transformations (ASTT), to manipulate information expressed in language and key data of application that correspond to coordinate transformations for GIS. For physics analysis, for example, conversions between event types would be in the Physics Analysis Tools and Transformations, and extension of the ASTT.

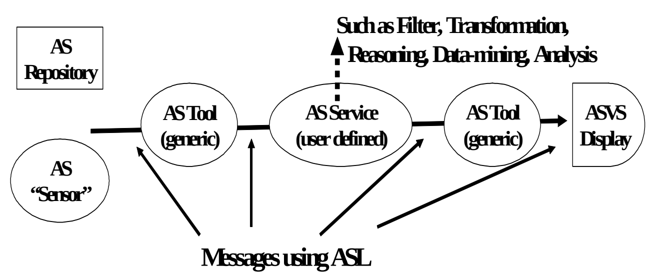


Figure 2: Application Specific Information System architecture for application specific Sensor Filter Grids.

We note that the ASL must support data sources such as sensors (in analogy to GIS metadata and data sensor standards) and repositories. Further, sensors need support of streams of data which can be common across applications. Queries in the application domain's information system need to support archived (all relevant data in past) and streaming (all data in future with given properties) styles. Note that all application services behave like sensors, and all sensors are wrapped as services as shown in Fig. 1.

We finally introduce Application Specific Visualization Services, generalizing the Open Geospatial Consortium's Web Map Service to both visualize information and provide a way of navigating Application Specific Feature Services and their underlying databases (cf. the *GetFeatureInfo* operation for the Open Geospatial Consortium's Web Map Service). The Visualization Service can itself be federated and presents an output interface. As in Fig. 2, all user and system services will input and output data in the appropriate domain-specific XML language using filters to process raw data. The high performance web service techniques described earlier allow one to combine high performance with the expressivity of XML.

V. Geo-Informatics Examples

We consider here briefly the application of these ideas to seismic archives and GPS sensor arrays. More technical descriptions are given in [32][34].

Accurate seismic event catalogs dating back to the 1930s are available from the United States Geological Survey and the Southern California Earthquake Center. Techniques for analyzing the underlying patterns in the data ("Pattern Informatics") have shown great promise in predicting areas of future significant earthquake activity [44] without detailed knowledge of the underlying techniques. Forecasts are made for 10 year windows. These analysis techniques and their associated GIS Web Feature and Web Map Services can be integrated into an automated system of interacting services, such as described in [45]. The output of the Pattern Informatics application may be considered another sensor feed. The advantage of this automated system is that one may treat the components as sensors (run at regular monthly intervals, for example) to continuously refine and update the seismic forecast and associated hazard maps.

The Southern California Integrated GPS Network (SCIGN) maintains over 250 continuously operating GPS stations throughout California. These stations are partitioned into sub-networks. This data is available through both online data archives and through real-time network access. Each of these is suitable for Geographical Information Servers as described above. For example, the GPS data archives are amenable to storage in a Web Feature Service. This data is available on demand and may be integrated into a larger Grid application using Web Service workflow techniques.

However, for our current discussion, the streaming data is more interesting. SCIGN GPS instantaneous position data is published at the rate of 1-2 Hertz, depending on the sub-network. Unfortunately, because these stations are grouped into sub-networks of several different stations, at any given second (or half-second), we will get all of the stations' position information on a particular sub-network, in the RYO binary format.

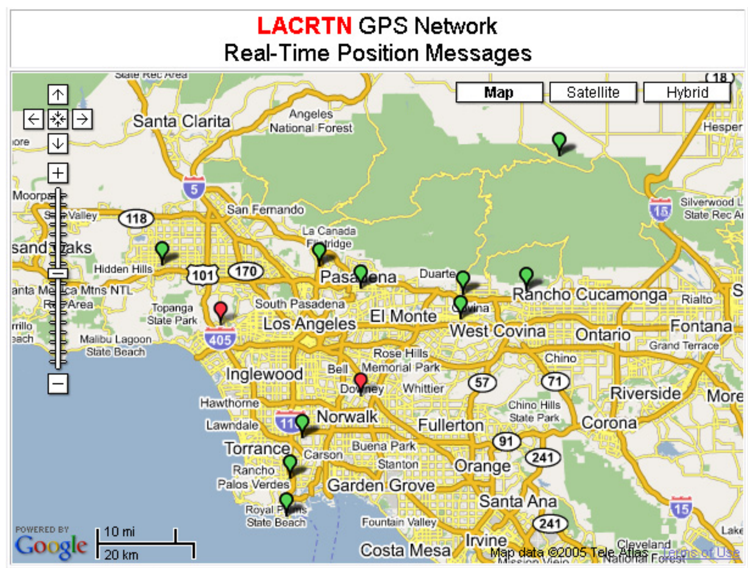
As we describe elsewhere, our approach has been to build a number of filters that can in succession,

1. Translate the RYO feed into ASCII;
2. Divide the ASCII feed for a sub-network into individual station feeds;
3. Process the individual feeds, formatting them in various XML languages, including RSS; and
4. Perform higher level time serious analysis techniques to detect interesting events in individual stations.

As can be seen, each of these filters is itself another sensor. The RSS feeds form a very simple but powerful way of publishing the time-stamped data coming out of the GPS stations, after sufficient translations. A sample RSS-like feed that we use in the work with SCIGN is shown below:

```
<message>
  <pos>
    <name>DSME</name>
    <lat>33.03647257927002</lat>
    <lon>-117.24952051832685</lon>
  </pos>
  <pos>
    <name>P482</name>
    <lat>33.24017400862219</lat>
    <lon>-116.67139746579954</lon>
  </pos>
…
</message>
```

RSS header fields, such as RDF metadata, can be inserted into the header of this text. This feed (again, generated after a series of filter operations) can be delivered to processing applications such as client applications built with Google Maps. The simple XML of the feed is easily manipulated by JavaScript to produce images such as Figure 3.



Station Name	Latitude	Longitude
CIT1	34.136704066944354	-118.12727056543028
VTIS	33.71263510818192	-118.29380819073941
CSDH	33.86147285146598	-118.2567022891129
LAPC	34.18191711345405	-118.5746357455248
HBCO	33.783641115047615	-118.28577976028518
LORS	34.13332225618835	-117.75405526659878
TABL	34.3818442137128	-117.67826751113105
AZU1	34.12601387118592	-117.89647244872606
CVHS	34.082006516429516	-117.90170243294764
VDCY	34.178560678897604	-118.21998511263479

Figure 3 Google maps are used to display live GPS station feeds, after several filtering steps.

It is worth emphasizing that filter chains do not have to be linear. For example, once we have the individual GPS station feeds in ASCII format, this can be delivered to many different subscribing filters that may perform uncorrelated operations.

The RDAHMM application [46][47] provides an example of a higher level data analysis filter. A full description of RDAHMM is out of scope here, but the gist of the application is to characterize, with no fixed parameters, the underlying modes in a time series. For GPS stations, “underlying modes” may include earthquakes but also more mundane (“aseismic”) events like the draining and refilling of nearby aquifers. These modes changes are typically not discernable by visual inspection. RDAHMM input files are typically collected from online data archives (such as our Web Feature Service), and modes may endure for several months.

There is also the chance that earthquakes may be signaled by much smaller, harder to detect mode changes as precursors. This is an area of very preliminary research, but it requires in essence that RDAHMM be modified to consume real-time data instead of data archives. RDAHMM in this case would potentially be able to detect mode changes that could not be seen by visual inspection. RDAHMM has not been validated in this application, but we have taken the first step of enabling its consumption of the real-time data streams.

VI. Sensors for Journal Filtering

We conclude with an example of a system we have started developing that applies the sensor filter grid approach to scientific journal articles.

Consider first a motivating scenario from chemical informatics. In the early stages of drug development, a chemist may search online public chemical compound databases and apply data mining techniques to identify compounds with similar desirable properties and structures. These compounds may be identified by unique text signatures such as InChIs™ [48]. The chemist may then search the Internet for all available literature on the InChIs that were obtained in the data-mining step. Even better, the chemist may want to develop a way of sorting the literature matches by, for example, working with his colleagues to rate and share the value of various journal articles. Finally, the chemist may want additional information (such as the appearance of new citations that include the InChIs of his interest, or perhaps InChIs from new compounds identified in the data mining step) to be pushed out as an RSS feed to all the collaborators in his group.

Our “Semantic Scholar” concept is intended to enable these sorts of interactions. The basic components are shown in Figure 4. The “Gather” filter fetches documents and their metadata from various sources. These can then be processed with text analysis programs to infer additional custom metadata tags. The resulting metadata can then be searched or pushed to users who sign up for the text as information feeds. Again, each of these components is a filter and is itself an online feed of information.

VII. Conclusion

In this paper we have examined the general nature of sensor filter grids. Starting from well-established realizations of Web Services for data access (the Geographical Information System services), we have identified components of a more abstract system. Alternative realizations of these abstractions in turn may be applied to diverse domains. For example, we may transfer the ideas of GIS systems to another domain such as chemistry. However, as we have discussed, the same general approach to filtering and transforming data as it is transported to subscribing applications can also be applied to human readable documents on the Web. With this identification made, we have as a next step the task of federating these various information sources.

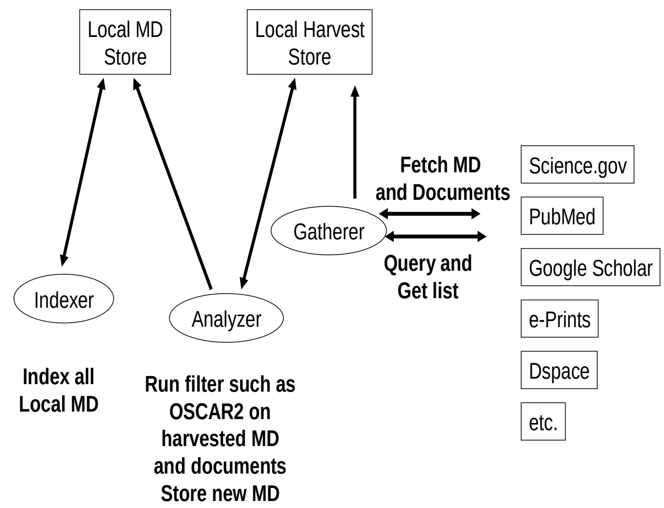


Figure 4 "Semantic Scholar" uses online scientific journals as time dependent sensor data soruces.

References

[1] Tony Hey and Anne Trefethen, “The Data Deluge: an e-Science Perspective” in *Grid Computing: Making the Global Infrastructure a Relaity*, Fran Berman, Geoffrey C. Fox, and Anthony J. Hey (eds), page 808-822 (2003).

[2] Julian J. Bunn and Harvey B. Newman, “Data-Intensive Grids for High Energy Physics” in *Grid Computing: Making the Global Infrastructure a Relaity*, Fran Berman, Geoffrey C. Fox, and Anthony J. Hey (eds), page 859-906 (2003).

[3] The Grid 2: Blueprint for a new Computing Infrastructure, edited by Ian Foster and Carl Kesselman, Morgan Kaufmann 2004.

[4] Grid Computing: Making the Global Infrastructure a Reality edited by Fran Berman, Geoffrey Fox and Tony Hey, John Wiley & Sons, Chichester, England, ISBN 0-470-85319-0, March 2003. http://www.grid2002.org.

[5] NASA/JPL Sensor Webs: http://sensorwebs.jpl.nasa.gov/

[6] Geoffrey Fox, Galip Aydin, Harshawardhan Gadgil, Shrideep Pallickara, Marlon Pierce, Wenjun Wu: Management of Real-Time Streaming Data Grid Services. GCC 2005: 3-12

[7] Beth Plale, Dennis Gannon, Daniel A. Reed, Sara J. Graves, Kelvin Droegemeier, Bob Wilhelmson, Mohan Ramamurthy: Towards Dynamically Adaptive Weather Analysis and Forecasting in LEAD. International Conference on Computational Science (2) 2005: 624-631.

[8] Quinn Hart, Michael Gertz: Querying Streaming Geospatial Image Data: The GeoStreams Project. SSDBM 2005: 147-150

[9] Graham Klyne and Jeremy J. Caroll, “Resource Description Framework (RDF): Concepts and Abstract Syntax,” W3C Recommendation 10 February 2004. See also http://www.w3.org/RDF/.

[10] Deborah L. McGuinness and Frank van Harnelen, “OWL Web Ontology Language Overivew,” W3C Recommendation 10 February 2004. See also http://www.w3.org/2004/OWL/.

[11] Sergey Brin, Lawrence Page: "The Anatomy of a Large-Scale Hypertextual Web Search Engine." Computer Networks 30(1-7): 107-117 (1998). See also www.google.com.
[12] RSS 2.0 Specification: http://blogs.law.harvard.edu/tech/rss
[13] Harshawardhan Gadgil, Geoffrey Fox, Shrideep Pallickara, Marlon Pierce Managing Grid Messaging Middleware Technical Report February 8 2006
[14] NaradaBrokering Open Source Messaging System http://www.naradabrokering.org
[15] Geoffrey Fox, Shrideep Pallickara, Marlon Pierce, Harshawardhan Gadgil, Building Messaging Substrates for Web and Grid Applications to be published in special Issue on Scientific Applications of Grid Computing in Philosophical Transactions of the Royal Society of London 2005 http://grids.ucs.indiana.edu/ptliupages/publications/RS-CGL-ColorOnlineSubmission-Dec2004.pdf
[16] WS-RM Web Services Reliable Messaging Protocol (WS-ReliableMessaging) http://www-106.ibm.com/developerworks/webservices/library/ws-rm/
[17] WS-Eventing. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnglobspec/html/WS-Eventing.asp
[18] WS-Notification. http://www-106.ibm.com/developerworks/library/specification/ws-notification
[19] Beytullah Yildiz, Shrideep Pallickara, Geoffrey Fox Experiences in Deploying Services within Apache Axis Container Technical Report October 2005
[20] Geoffrey Fox, Galip Aydin, Harshawardhan Gadgil, Shrideep Pallickara, Marlon Pierce, and Wenjun Wu Management of Real-Time Streaming Data Grid Services Invited talk at Fourth International Conference on Grid and Cooperative Computing (GCC2005), held in Beijing, China, Nov 30-Dec 3, 2005. http://grids.ucs.indiana.edu/ptliupages/publications/gcc2005paper-final_long.pdf.
[21] Shrideep Pallickara, Geoffrey Fox, Ahmet Uyar, Hongbin Liu, Xi Rao, David Walker, Beytullah Yildiz: Performance of a possible Grid message infrastructure. Concurrency - Practice and Experience 17(2-4): 193-214 (2005)
[22] Data Management and Transfer in High Performance Computational Grid Environments. B. Allcock, J. Bester, J. Bresnahan, A. L. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnal, S. Tuecke. Parallel Computing Journal, Vol. 28 (5), May 2002, pp. 749-771.
[23] A Metadata Catalog Service for Data Intensive Applications. G. Singh, S. Bharathi, A. Chervenak, E. Deelman, C. Kesselman, M. Mahohar, S. Pail, L. Pearlman. Proceedings of Supercomputing 2003 (SC2003), November 2003.
[24] GriPhyn Grid Physics Network http://www.griphyn.org/index.php
[25] iVDGL - International Virtual Data Grid Laboratory, http://www.ivdgl.org/index.php
[26] PPDG Particle Physics Data Grid http://www.ppdg.net/
[27] Anabas e-learning and Web conferencing http://www.anabas.com
[28] Wenjun Wu, Hasan Bulut, Geoffrey Fox, Ahmet Uyar: Adapting H.323 Terminals in a Service-Oriented Collaboration System. IEEE Internet Computing 9(4): 43-50 (2005).
[29] Hasan Bulut, Wenjun Wu, Geoffrey Fox, Ahmet Uyar, Shrideep Pallickara, Harun Altay: A Web Services Based Streaming Gateway for Heterogeneous A/V Collaboration. International Conference on Internet Computing 2004: 493-502.
[30] Geoffrey Fox, Wenjun Wu, Ahmet Uyar, Hasan Bulut, Shrideep Pallickara: Global Multimedia Collaboration System. Middleware Workshops 2003: 245-250.
[31] Mehmet S. Aktas, Geoffrey C. Fox, Marlon Pierce Information Services for Dynamically Assembled Semantic Grids Proceedings of 1st International Conference on Semantics, Knowledge and Grid Beijing China November 27-29 2005
[32] Galip Aydin, Geoffrey C. Fox, Harshawardhan Gadgil and Marlon E. Pierce Streaming Data Services to Support Archival and Real-Time Geographical Information System Grids Technical Report February 10 2006
[33] Mehmet S. Aktas, Geoffrey Fox, Marlon Pierce Managing Dynamic Metadata as Context Istanbul International Computational Science and Engineering Conference (ICCSE2005) June 2005 http://grids.ucs.indiana.edu/ptliupages/publications/maktas_iccse05.pdf
[34] Galip Aydin, Ahmet Sayar, Harshawardhan Gadgil, Mehmet S. Aktas, Geoffrey C. Fox, Sunghoon Ko, Hasan Bulut,and Marlon E. Pierce Building and Applying Geographical Information System Grids 12 January 2006, Special Issue on Geographical information Systems and Grids based on GGF15 workshop, Concurrency and Computation: Practice and Experience.
[35] Harvey B. Newman, I. C. Legrand, Philippe Galvez, R. Voicu, C. Cirstoiu: MonALISA : A Distributed Monitoring Service Architecture CoRR cs.DC/0306096: (2003)
[36] Chiu, K., Govindaraju, M., and Bramley, R.: Investigating the Limits of SOAP Perform-ance for Scientific Computing, Proc. of 11th IEEE International Symposium on High Per-formance Distributed Computing HPDC-11 (2002) 256.
[37] R. Berjon, chair, "XML Binary Characterization Working Group Public Page." Available from http://www.w3c.org/XML/Binary/
[38] Paul Sandoz, Alessando Triglia, and Santiago Pericas-Geertsen, "Fast Infoset." Available from http://java.sun.com/developer/technicalArticles/xml/fastinfoset/.
[39] Oh, S., Bulut, H., Uyar, A., Wu, W., Fox G., Optimized Communication using the SOAP Infoset For Mobile Multimedia Collaboration Applications. in proceedings of the Interna-tional Symposium on Collaborative Technologies and Systems CTS05 (2005). http://grids.ucs.indiana.edu/ptliupages/publications/OptSOAP_CTS05.pdf
[40] Ken Birman, Robert Hillman, Stefan Pleisch, Building network-centric military applications over service oriented architectures SPIE Conference on Defense Transformation and Network-Centric Systems at Orlando Florida 31 March 2005 http://www.cs.cornell.edu/projects/quicksilver/public_pdfs/GIGonWS_final.pdf.
[41] Open Geospatial Consortium Sensor Web Enablement Functional Area Page: http://www.opengeospatial.org/functional/?page=swe.
[42] M. Botts (ed) "OpenGIS® Sensor Model Language (SensorML) Implementation Specification" OpenGIS Best Practices Paper OGC 05-086.
[43] Geoffrey Fox, Alex Ho, Shrideep Pallickara, Marlon Pierce, Wenjun Wu: Grids for the GiG and Real Time Simulations. DS-RT 2005: 129-138.
[44] Tiampo, K.F., Rundle, J.B., McGinnis, S., and Klein, W. Pattern dynamics and forecast methods in seismically active regions, Pure and Applied Geophysics, 159, 2002. See also http://quakesim.jpl.nasa.gov/scorecard.html
[45] Galip Aydin, Mehmet S. Aktas, Geoffrey C. Fox, Harshawardhan Gadgil, Marlon Pierce, Ahmet Sayar SERVOGrid Complexity Computational Environments (CCE) Integrated Performance Analysis Proceedings of Grid Computing Conference, 2005. The 6th IEEE/ACM International Workshop 13-14 Nov. 2005. Page(s): 256 - 261 DOI
[46] Granat, R. A., *Regularized Deterministic Annealing EM for Hidden Markov Models*, Ph.D. Thesis, University of California, Los Angeles, 2004.
[47] Granat, R. A., A Method of Hidden Markov Model Optimization for Use with Geophysical Data Sets, *Comp. Sci*., 2659, pp. 892–901, 2003.
[48] An Open Standard for Chemical Structure Representation: The IUPAC Chemical Identifier, Stephen E. Stein, Stephen R. Heller, and Dmitrii Tchekhovskoi, *in Proceedings of the 2003 International Chemical Information Conference (Nimes)*, Infonortics, pp. 131-143.