Received Date : 05-Nov-2015
Revised Date : 14-Mar-2016
Accepted Date : 02-Jun-2016
Article type : Editorial

Corresponding author mail id:- ***bkerkez@umich.edu***

Real-time Data Services for the Geosciences (CHORDS)

Branko Kerkez[1], Michael Daniels[2], Sara Graves[3], V. Chandrasekar[4], Ken Keiser[3], Charlie Martin[2], Michael Dye[2], Manil Maskey[3], Frank Vernon[5]

[1]University of Michigan, [2]National Center for Atmospheric Research, [3]University of Alabama in Huntsville, [4]Colorado State University, [5]University of California San Diego

While modern sensing and communication technologies are enabling the observations of geophysical processes at unprecedented spatiotemporal resolutions, the development of these technologies is significantly outpacing their actual use across the geosciences. This is particularly true of real-time data systems, which are now permitting the streaming and analysis of data at the instant of their measurement. Though the use of real-time scientific data are limited, their importance is ever increasing, particularly in mission critical scenarios where informed decisions must be made rapidly.

Beyond applications tied to disaster resilience (earthquake prediction, flood forecasting, etc.), now more than ever there is potential to leverage real-time data to fundamentally change how scientific experiments are conducted. For example, in many geoscientific experiments, faulty sensors are often only detected too late, forcing experiments to be repeated. In settings where mobile sensor nodes are used, or where sampling frequencies need to be adjusted to capture events of interest, few tools are available to adaptively guide the experimental process. This often results in missed observations and wasted experimental investments but can be remedied rapidly by enabling means analyze and respond to streaming data.

While real-time data stand to enable a paradigm-shift in geoscientific experimentation, they rarely, if ever, form the first step in a geoscientific workflow. The vast majority of existing data platforms are inherently tuned to non real-time applications, where data are often stored in large databases for retrospective analysis and visualization. The few existing real-time data platforms, however, are either proprietary, feed into mission-specific tools, or are otherwise not available to broader stakeholders within the geosciences. While the complexity of these platforms presents a major barrier to the broader adoption of real-time data systems, there are also a number of technical challenges that must be addressed before the use of real-time data becomes commonplace across the geosciences.

Existing real-time data platforms

While interoperability standards such as the *Open Geospatial Consortium*[1] *(OGC) Sensor Web Enablement (SWE)* specifications (Nittel, Labrinidis, & Stefanidis, 2008), have created interfaces and metadata encodings to fuse real-time sensor streams into information infrastructures, a common set of tools to couple these streams with workflows and models has yet to be developed. To that end, pioneering efforts are underway by groups such as 52°North [2] to develop standards and reference implementations for real-time data within the field of Geoinformatics (Andres, Jirka, & Utech, 2014; Jirka, Bröring, Kjeld, Maidens, & Wytzisk, 2012; Reed, Botts, & Davidson, 2007). Beyond SWE-based initiatives, a number of other platforms have also been developed to address the emergence of real-time data within the geosciences.

---

[1] www.opengeospatial.org/standards

[2] http://52north.org

UNIDATA's Local Data Manager (Davis & Rew, 1994) provides an event driven infrastructure to manage streaming data. While it has served the purpose of specific projects for many years, the system can be difficult even for an experienced user to install and maintain. Since LDM queues data, the system is not suited for environments in which the stability of networks cannot be assured, which may often be the case with data originating from real-world sensor networks. Its queuing process may also lead to situations where the latest real-time data are not accessible until the queue buffers are flushed, thus causing a backlog of data that prevent timely use.

Other recent real-time efforts have been undertaken through the *DataTurbine* (Tilak, Hubbard, Miller, & Fountain, 2007) and Antelope[3] initiatives. DataTurbine is based on a ring-buffer architecture and is implemented in Java as an open-source, server-side platform for the transport and management of real-time data originating from heterogeneous sensors. While powerful, the ring-buffer architecture does not actively support real-time database operations or coupled model-sensor applications. Furthermore, local server resources can limit the size of the ring-buffer, making it possible to drop incoming data. Cloud-based functionalities and OGC standard support are yet to be implemented as features. Significant overhead exists on the part of users, as *DataTurbine* has to be individually ported to field-specific data loggers and instruments. While these examples may appear specific to one platform, they are echoed by all the other real-time data systems as well. The complexities associated with the deployment and operation of existing real-time data platforms present an overhead too large for most research groups to take on, thus significantly limiting the broader adoption of real-time data across the geosciences. The emergence of commercial data platforms under the Internet of Things (IoT) is beginning to provide easier to use alternatives, but these platforms are not directly tailored to the demands imposed by geoscientific applications (Gubbi, Buyya, Marusic, & Palaniswami, 2013; Palattella et al., 2013).

Challenges

---

[3] http://www.brtt.com

A workshop was held in the summer of 2013 as part of the U.S. NSF's *EarthCube Initiative*[4], entitled "*Integrating Real-time Data into the EarthCube Framework*." The *EarthCube* program seeks to build a common framework for the analysis, aggregation, and coupling of geoscientific data and models. The primary consensus of the workshop[5], as provided by over 75 participants spanning a broad set of geoscientific disciplines, revealed that while *EarthCube* will provide an unprecedented framework for disseminating historical data sources, the use of real-time data raises an additional set of complex challenges, which must be addressed explicitly. Furthermore, it was agreed that these challenges are not being addressed by existing real-time data tools.

Complexity of deployment is perhaps the biggest barrier to the adoption of real-time data. A key aspect of managing in-situ and dynamic sensor data in real-time is providing efficient discovery, access and processing of sensor observations. Ideally, scientists should not have to be concerned with heterogeneous formats, sensors and sources of data. Rather, easy-to-use systems must be developed to permit scientists to focus on analysis and experimentation rather than complex system maintenance. To that end, a number of core challenges should be addressed to facilitate the adoption of real-time data:

- Continued community discussion is required to build consensus around features and the real-world uses of real-time data platforms.
- Installation and configuration of these systems should be seamless and as easy as possible. This may be accomplished by cloud-hosted infrastructure that features pre-configured instances of the platform, thus reducing the need for complex, local user maintenance.
- Real-time data systems should provide standard interoperability interfaces to sensor data to minimize the custom software required for management, visualization and analysis of different types of sensor observations. These platforms should also adhere in as much as possible to common data and metadata formats that adhere to standards (such as the OGC's *Sensor Web*).

---

[4] http://earthcube.org
[5] https://www.eol.ucar.edu/news-and-events/workshops/earthcube-realtime-data-workshop

- Platforms should also provide a system to archive, navigate and distribute non real-time data streams via the Internet.

A reference implementation

Presently, a working group is spearheading the use of real-time data within *EarthCube* under the *Cloud-Hosted Real-time Data Services for the Geosciences (CHORDS)* project[6]. While the primary goal of CHORDS is to drive a community discussion around the adoption of real-time data, reference architecture is also being developed to serve as an example for future implementations of real-time data systems. A number of use cases are being evaluated within this platform to showcase the potential of real-time data toward improving scientific experiments. Examples include, but are not limited to, the analysis and visualization of measurements collected by scientific aircraft, real-time seismic sensor networks for the detection of tornadoes, GPS-based volcano monitoring, and data streaming services for a new generation of affordable 3D-printed weather stations.

One particular use case involves the coupling of real-time, distributed meteorological and hydrologic data. The use case is intended to illustrate the study of extreme events, such as flooding, where hydrologic models are forced by meteorological inputs. In such cases it is vital to couple precipitation data with local flow conditions to forecast flooding. This application couples complex raster data, time series, and metadata, which must be reconciled within the same framework. The CHORDS reference architecture (Figure 1) is explicitly developed with ease-of-use in mind, permitting even small research teams to have a turnkey path toward using real-time data. Three main layers comprise the architecture: 1) the CHORDS Portals, which are the entry and distribution points for all real-time data, 2) CHORDS Services, which provide optional, value-added features, and 3) powerful standards to interface with workflows and EarthCube building blocks.

---

[6] http://chords.earthcube.org

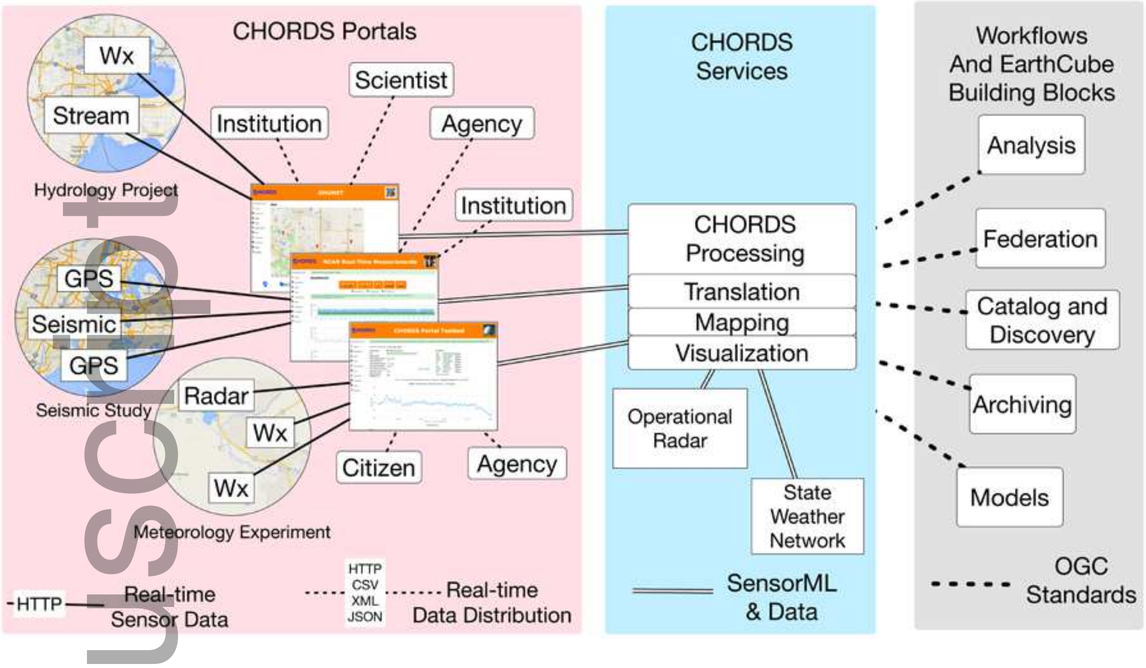


Figure 1: CHORDS Architecture: Sensors push real-time data to CHORDS Portals, which provide easy web-services access to the data streams. Portals can optionally interface CHORDS Services, which provide additional functionality and interoperability with popular standards and EarthCube services.

A CHORDS portal can be launched as a pre-configured instance on a commercial cloud platform, allowing users to deploy it with minimal setup overhead. Each CHORDS user owns and manages their own CHORDS instance and interfaces it with their data streams. A preconfigured web server on the instance hosts a user interface, which is used to define data streams that will be ingested by the instance. This interface is used to generate simple URL schemes, which can be loaded directly into data sources (data loggers, instruments, algorithms, etc). A corresponding ingester is generated for each URL scheme, which translates the incoming sensor data into a common CHORDS format that is then hosted by the portal for external distribution. This permits users to keep their data sources relatively unaltered, having only to push a simple HTTP/REST post when new measurements are made. Data can be written to and read from the CHORDS portal via a set of standard encodings, such as JSON and XML. Data can even be pushed into CHORDS using simple CSV or binary formats to limit programming of field-deployed devices. The support for other popular data formats is continually expanding, with plans to incorporate formats such as netCDF in the near future.

While CHORDS portals provide a rapid way to ingest and share data from multiple real-time sensor networks, their functionality can be vastly expanded by interfacing with the CHORDS services layer. This layer is hosted by EarthCube's CHORDS team and provides a central registry of all deployed portal instances. It serves as a repository of metadata and expands the portal functionality with additional higher-level features such as visualization, mapping, and basic resampling or filtering algorithms. It even provides access to some popular real-time feeds, such as radar data or operational weather networks. A GIS framework (*GeoServer*[7]) is built into the services layer to facilitate the visualization, retrieval and discovery of data based on geographic regions of interest (Figure 2). The services layer interoperates with the larger family of evolving web-based OGC data services and standards, a feature that is continual maintained and updated by the CHORDS team to support a growing set of external services and workflows, such as those offered by EarthCube.

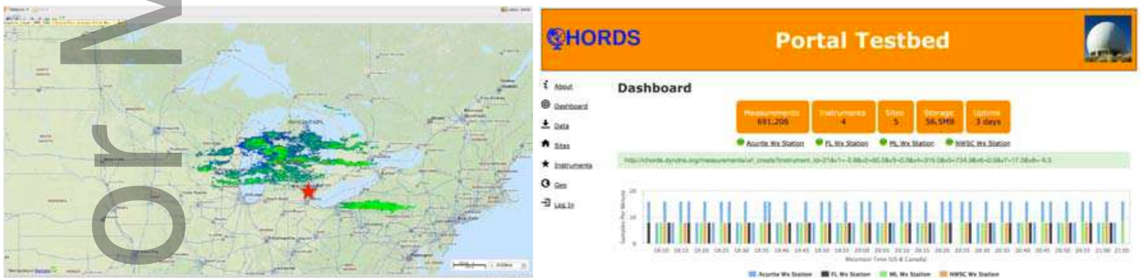


Figure 2: Example use case, where data from a hydrologic sensor node (red star) is overlaid on radar data to predict local precipitation and forecast flooding conditions.

A major advantage of CHORDS will be that the end user can work in whatever environment is most effective for them. No specific programming languages are forced onto data producers or end users, as the only requirement is the ability to process HTTP/RESTful requests. This permits the seamless integration of CHORDS services into

---

[7] http://geoserver.org/

most existing instrumentation, models, and visualizations. Once configured, research teams can then easily incorporate a suite of algorithms into their real-time workflows. These workflows could include systems ranging from highly integrated command and control systems, data assimilation into models, field project control centers, standalone applications, web visualizations, or spreadsheets.

While the project is still in its infancy, initial use-case assessments are very favorable. CHORDS does not aim to be a one-size-fits-all solution for real-time data, nor is its present implementation an operational real-time data platform. A number of limitations currently exist, which will be addressed in the future based on community feedback. All of the current use cases are based on low latency requirements. The current implementation does not support photo or video data, which may be relevant to studies that require real-time image analysis. While existing system could readily support data rates at 10-60Hz per feed, data rates at higher magnitudes, especially for spatial data, would require further testing and improvements. Model integration has also not been tested yet, but use cases are underway to investigate how to best couple CHORDS with publically hosted modeling services. For example, work is underway to connect the real-time hydrometeorological application with hydrologic models for flood forecasting. Bi-directional communications are currently not supported, which means that CHORDS can receive data from remotely-deployed instruments but not control them. More advanced OGC SWE functionalities, such as Sensor Planning Services, are also planned for implementation to enable remote tasking of a field sensors, which will enable adaptive sampling of geoscientific phenomena (Andres et al., 2014). Given the infancy of the project, there are many more features that will be required to make CHORDS a fully hardened real-time data platform. This will require the need for built-in security and encryption, which will be vital in protecting field-deployed scientific assets and servers. CHORDS is also not a storage repository, data discovery or cataloging service, as those features are expected to be addressed by existing domain-community repositories and services. . Rather, its goal is to serve as a reference for community feedback, which will ultimately lead to consensus on architectures for real-time data.

Discussion and Conclusions

Given resource constraints of existing experiments, real-time data has the potential to play a pivotal role in the future discovery of geoscientific processes. This will be achieved by responding to data as soon as they are collected to detect faulty instrumentation and adaptively allocate in-situ measurement resources. Furthermore, many geoscientific data streams have the potential to change how information is consumed by non-scientific stakeholders (during disaster events, for example). Given the complexity of existing platforms however, much work remains to be done on simplifying the use of real-time data platforms, so that scientists may focus on experimentation, rather than platform maintenance. Over the coming years, the CHORDS initiative will seek to carve out a vision and reference implementation of real-time data. During this process, community engagement will be the most critical mechanism toward making real-time data in the geosciences a reality.

Acknowledgements

The authors would like to acknowledge the support of the United States National Science Foundation in supporting this work through award 1440116.

References

Andres, V., Jirka, S., & Utech, M. (2014). *OGC Best Practice: OGC Sensor Observation Service 2.0 Hydrology Profile (OGC 14-004r1)*. Wayland, MA, USA.

Davis, G., & Rew, R. (1994). The Unidata LDM: programs and protocols for flexible processing of data products. *International Conference on Interactive Information and Processing Systems.*

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, *29*(7), 1645–1660. doi:10.1016/j.future.2013.01.010

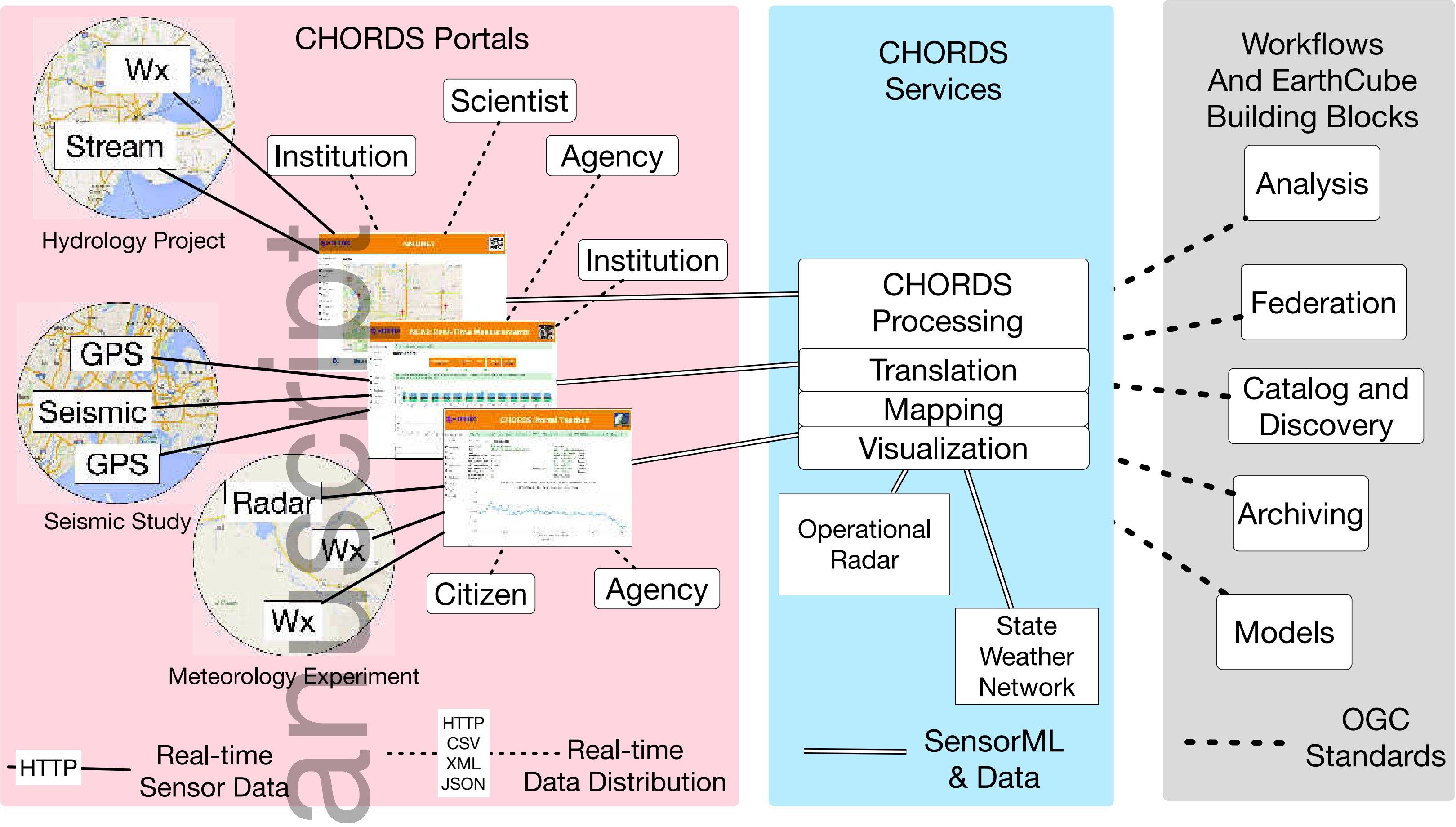
Jirka, S., Bröring, A., Kjeld, P., Maidens, J., & Wytzisk, A. (2012). A Lightweight Approach for the Sensor Observation Service to Share Environmental Data across Europe. *Transactions in GIS*, *16*(3), 293–312. doi:10.1111/j.1467-9671.2012.01324.x

Nittel, S., Labrinidis, A., & Stefanidis, A. (Eds.). (2008). *GeoSensor Networks* (Vol. 4540). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-79996-2

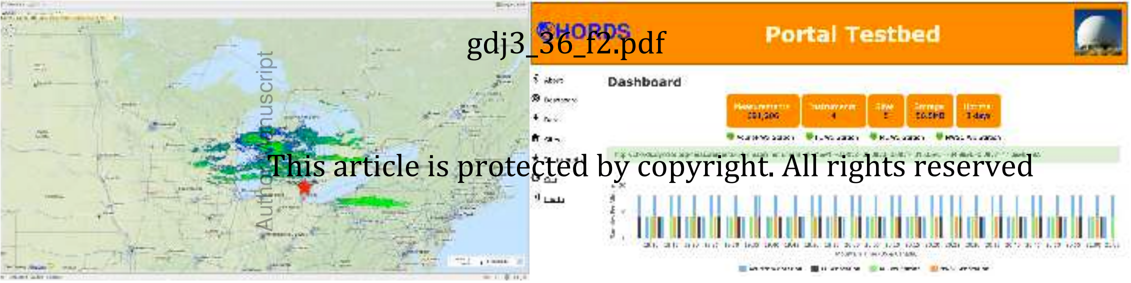
Palattella, M. R., Accettura, N., Vilajosana, X., Watteyne, T., Grieco, L. A., Boggia, G., & Dohler, M. (2013). Standardized Protocol Stack for the Internet of (Important) Things. *IEEE Communications Surveys & Tutorials*, *15*(3), 1389–1406. doi:10.1109/SURV.2012.111412.00158

Reed, C., Botts, M., & Davidson, J. (2007). Ogc® sensor web enablement:overview and high level achhitecture. In *2007 IEEE Autotestcon* (pp. 372–380). IEEE. doi:10.1109/AUTEST.2007.4374243

Tilak, S., Hubbard, P., Miller, M., & Fountain, T. (2007). The Ring Buffer Network Bus (RBNB) DataTurbine Streaming Data Middleware for Environmental Observing Systems. In *Third IEEE International Conference on e-Science and Grid Computing (e-Science 2007)* (pp. 125–133). IEEE. doi:10.1109/E-SCIENCE.2007.73



gdj3_36_f1.eps


CHORDS
Portal Testbed
Dashboard