Making Sensor Data Available Using Web Feeds

Erik Wilde
School of Information
UC Berkeley
dret@berkeley.edu

ABSTRACT

The setup of and processing within sensor networks often requires sophisticated and specialized system designs and implementations, but the service provided by them should be as accessible and repurposable as possible. If the increasing number of available sensor-based data sources can be accessed in a simple and universal way, the network effect of aggregating, filtering, and republishing data from these sources will significantly increase their value. We propose an architecture where sensor-based data sources publish their data based on feeds, but extended with query capabilities. Using the well-known and widely supported Atom feed format and extending it with query capabilities allows us to lower the barrier-of-entry to sensor-based data sources, opening this data to a wider audience of clients.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Web-based Services*; H.2.3 [**Database Management**]: Languages—*Query Languages*

Keywords

Atom, GeoRSS, Web Services, Loose Coupling

1. INTRODUCTION

Many of the challenging problems of sensor networks require sophisticated and specialized system designs and implementations, but another interesting question is how to make the *service* provided by these networks available. One of the recurring features of sensor networks is that they not only implement ways to aggregate sensor data, but also provide services beyond just forwarding raw sensor data, allowing applications to aggregate or filter this data, or to set thresholds for data they are interested in. This service perspective can be taken as far as regarding a sensor network as a temporal distributed database, and allowing queries over it [1].

While this approach is appropriate for exposing the full power of a sensor network to people needing access to that, this paper looks at a more lightweight approach and asks how this abstract view of a sensor network as an information stream could be exposed on the Web. The idea of *loose coupling* [5] in this case leads us to attempt to reuse or extend existing and established technologies to expose the wealth of information made available by sensor networks. We propose to use Web feeds (RSS or *Atom* [3]) as a way to expose sensor network services.

2. FEEDS AS A SERVICE

Feeds can be regarded as a primitive Web service; they always provide information about a fixed topic, and do so in a latest-first sort order, optionally providing support for feed paging. The most important benefit of this simple model is good scalability: feed providers update the feed as new entries become available, and feed consumers must not be known to the feed provider and do not trigger any query processing; they just retrieve the latest version of the feed. Caching can optimize this scenario even further. The downside of this “one size fits all” approach is that any data source that should be addressed in a slightly more complex way cannot be easily made available through a feed.

We thus propose an service model for feeds that extends feeds in two essential ways: Feeds support *queries* so that feed consumers can become more specific in the kind of information they want to consume; and the feed query data model takes any service into account that the *collection* supports, not just the collection data exposed in the *feed* itself. The essential difference is that a *collection* is data as it is managed by a feed provider, and this data might support rich interaction and query methods; whereas a *feed* is just a serialization of a number of entries from that collection.

One example where this is used already is *GeoRSS* [4], an extension to feeds which allows feeds to carry spatial information. In this case, a collection can expose more of its information in a feed because of an extension of the feed data model and format. Another example where this has been proposed is the *Feed Item Query Language (FIQL)* [2], which allows feeds to be queried using URI-based queries. Our approach combines these technologies and additional concepts into a feed service that essentially turns the "one size fits all" feed model of today into a more flexible and powerful model of how to interact with feeds.

On the Web architecture level, this model makes it possible to represent the information needs of a feed consumer in a declarative and well-defined way and it also allows a feed provider to advertise their query support in a declarative and well-defined way. This allows a looser coupling between providers and consumers than any architecture with a more tight coupling that requires providers to know the specifics and details of the way how various providers' access to their sensor networks' data is designed.

3. FEED QUERY ARCHITECTURE

Our approach of richer interactions with feeds is based on three main principles:

- *Data and Query Model:* There is a data model for feeds, feed extensions (such as GeoRSS), and basic datatypes. This data model represents the searchable access to a collection. The query model defines query capabilities, and can be mapped to whatever query mechanism is used by the provider; typically this will be general-purpose query languages such as SQL, or more specialized ones for special types of feeds, such as sensor network query languages for feeds providing access to sensor data.

- *Query Capability Advertisements:* As an extension of the standard feed data format, feed query providers can advertise the query capabilities they provide for their collections. This allows client to adapt to queryable feeds at runtime. For example, a mobile client can access any feed advertising its query capabilities, and can generate a search form from the published capabilities. Feature discovery is thus possible through the feed itself, and not through some separate registry.

- *Query Opacity:* Since query information is represented declaratively, it does not really matter where queries are executed. It can be done on the back-end, or by an intermediary. The difference might be that an intermediary cannot query information that is in the collection but not in the feed, but such an intermediary can either pass on these queries upstream, or it can remove these query capabilities from the capability advertisements.

This is only a broad overview of our proposed architecture of queryable feeds. For demonstration purposes, we are proposing an implementation as an intermediary which is crawling and caching feed entries. This approach, however, is incapable of working in the more advanced scenarios of sensor networks, which have very different characteristics from the more traditional scenarios of feed-based publishing on the Web, for example news sites.

For these scenarios, feed query processing has to be implemented by the service provider directly, and probably can be implemented rather easily as long as the underlying sensor network has some high-level abstraction such as a sensor network query language. The main advantage of a feed query interface to sensor networks would be the fact that diverse and heterogeneous collections of sensor networks could be accessed in a unified way, allowing applications to aggregate data from all of them without the need to address the implementation of any of these networks.

4. CONCLUSIONS

The architecture proposed and described in this paper uses the successful and widely known concept of Web feeds, and extends it to a query-enabled model. This allows the services of sensor networks to be exposed in an easily accessible and universal way. By mapping feed queries to the specialized processing features of sensor networks, these queries provide the benefit of a declarative way of accessing sensor networks, but can still take advantage of the optimizations and advanced processing that is implemented within the sensor networks.

5. REFERENCES

[1] Hideyuki Kawashima. KRAFT: A Real-Time Active DBMS for Signal Streams. In *Proceedings of the 4th International Conference on Networked Sensing Systems*, pages 163–166, Piscataway, New Jersey, June 2007. IEEE Computer Society Press.

[2] Mark Nottingham. FIQL: The Feed Item Query Language. Internet Draft draft-nottingham-atompub-fiql-00, December 2007.

[3] Mark Nottingham and Robert Sayre. The Atom Syndication Format. Internet RFC 4287, December 2005.

[4] Open Geospatial Consortium. An Introduction to GeoRSS: A Standards Based Approach for Geo-enabling RSS feeds. OGC 06-050r3, July 2006.

[5] Cesare Pautasso and Erik Wilde. Why is the Web Loosely Coupled? A Multi-Faceted Metric for Service Design. In *Proceedings of the 18th International World Wide Web Conference*, Madrid, Spain, April 2009. ACM Press.