XML and APRS

Steve Dimse K4HG
189 Le Grand Lane
Cudjoe Key, FL 33042
k4hg@tapr.org

Abstract

Recently, a number of pundits have been calling XML (extensible Markup Language) the “Next Big Thing” on the Web. New XML aware applications are being released all the time. Over the few years, more and more information will be made available in XML. The ability to easily machine-interpret this information will be a watershed event in the development of the Internet. APRS will not be left out! In this paper I detail the first efforts in making APRS data available in XML.

Introduction

While some have called XML a more advanced form of HTML (HyperText Markup Language), this shortchanges XML, and fails to describe the true relationship between the two. HTML grew out of an important standard known as SGML (Standardized General Markup Language). This is a DoD standard that is used to maintain vendor-independent structured documentation. It is very powerful, very complex, and very difficult to implement. A very limited and well defined subset of SGML was initially selected for inclusion in HTML, to specifically tailor it to the needs of hypertext documents served over the Internet. Over time HTML has evolved to meet the specific needs of web users.

A serious limitation of HTML is that it only describes the way a document looks. Formatting instructions are enclosed between the characters ‘<’ and ‘>’. The tag `<h1>` defines a top-level header, and causes the following text to be printed in large, bold letters. However, HTML provides no way to identify what that information represents.. . it might be a rock group, the title of a scientific paper, or a disease. The human reader is able to identify the meaning from the context, but this is a very difficult process for a computer to duplicate.

XML addresses this shortcoming of HTML. While sharing a common derivation from SGML, XML allows the user to define arbitrary structures since, as the name suggests, it is user extensible. Rather than formatting, the focus in XML is on the structure and meaning of the underlying data. The HTML phrase `<h1>Counting Crows</h1>` might become `<h1><rock group>Counting Crows</rock group></h1>` in XML. Now, for example, a Web indexing engine would be able to tell that this page would be of no interest to a corn farmer concerned about rising pilferage by birds!

Data Tags for APRS

The most essential task is creating a standard set of data tags. There are many efforts underway to standardize the tags for specific applications. Before beginning this effort for APRS, I searched the Internet for an existing standard, but I was unable to find any XML specifications pertaining to position or mapping information. Therefore I decided to start from scratch. Part of the: design philosophy of the XML standard is bandwidth is cheap. There are no cryptic tags in the interest of saving a few bytes. The result of this is that the data is very verbose, but very readable.

Sample APRS data converted to XML

```
<station>
      <callsign>K4HG</callsign>
      <position>
            <timeReported>1999-01-16T22:32:06Z</timeReported>
            <timeReceived>1999-01-16T22:32:11Z</timeReceived>
            <positType>Mic-E</positType>
            <icon>6,43</icon>
            <lat>24.5434</lat>
            <lon>-81.5432</lon>
      </position>
      <weatherReport>
            <timeReported>1999-01-16T22:32:06Z</timeReported>
            <timeReceived>1999-01-16T22:32:11Z</timeReceived>
            <windSpeed>21</windSpeed>
            <windDirection>326</windDirection>
            <humidity>86</humidity>
            <barometer>1004.3</barometer>
            <rain1Hour>0.02</rain1Hour>
            <rain24Hour>0.25</rain24Hour>
            <rainMidnight>0.14</rainMidnight>
            <temperature>86</temperature>
      </weatherReport>
</station>
```

As you can see, this format takes considerably more storage than the native APRS formats. However, instead of the 6 different position formats defined in APRS, there is a single format that a program must parse.

Uses of XML and APRS

Perhaps the most successful portion of my Internet/APRS software is the map.aprs.net server. This allows the display of street level maps for any station whose position is known to the APRServe, using any ordinary browser. There is a simple web server embedded directly into the software. The user requests a callsign, the server checks its internal database, and dynamically generates an HTML page, including calls to the MapBlast web server to generate the maps. Unfortunately, under this system there can only be a single page served to all users. To modify the page, I must change the APRServe software and reboot the server. There is no way for the user to customize his web page. With XML, it will be easy for people to develop their own web page, using an HTML feature called a template, which will parse the XML data and display it in any fashion they choose.

Another possibility is taking the data, and importing it into other programs. For example, one could take the data for a balloon flight, import it into Excel, and generate a graph based on altitude.

XMLserve

Every day, APRServe handles about 25 MB of data. No systematic archive is made of it. XMLserve will be a huge database of this data, and allow sophisticated queries to be made, for example give me my position every fifteen minutes for the last month. The results will be returned in XML, allowing the user to then manipulate or display the data in any fashion they wish. At this time I am still trying various database programs for the back-end.

Present Status

There are examples of the preliminary XML data format, and an embryonic parser written in C++ available on my server at http://www.aprs.net/xml. My finalization of the specification is awaiting the completion of the APRS Working Group's protocol document. Comparing the XML tags to the protocol will ensure that no data format is missed. Also, I invite input from others interested in this subject to comment on the specification. Once it is complete, I plan to submit it to the APRS Working Group as a proposed standard.

The APRS XML suite and XMLserve will do nothing on their own.. . these are enabling technologies. What is needed are parsers both into and out of XML for other popular languages, like Perl, C, Visual Basic, and Java. I hope others will fill these gaps with open source programs. Also needed are examples of template HTML files for position and weather data. Finally, real-world uses for the data need to be dreamed up and implemented.