MTConnect 's XML-based Approach

In a typical scenario in which instrumented equipment emits data files in a proprietary format, extensive knowledge of the schema (data layout, and what each piece of data represents) used by each piece of equipment would be required to accomplish even a simple task. For example, a temperature-analysis software module might want to look for specific temperature-related data in data files emitted by a variety of equipment, while ignoring other fields it does not care about. To accomplish this, all pieces of equipment would have to emit the specific schema for the analysis tool to read. Insisting on this type of a single “portable” schema is difficult to achieve and in the past has not been a universally accepted answer: in this case, advance agreement would be required by all participants as to what data would and would not be included in the schema, and how it would be represented. We are faced with a Hobson’s choice: if each schema is proprietary, interoperation is next to impossible, but trying to get advance agreement on a single “master” schema seems just as hard.

In contrast, the mass-market “explosion” of the World Wide Web in the early 1990’s has taught us important lessons in interoperability—in particular, the widespread adoption of XML (Extensible Markup Language) as a flexible representation for exchanging semistructured machine-readable data. Semistructured means that while the fields of an XML document emitted by a particular entity represent specific values, the description of what the values represent (i.e., the schema) is self-describing and carried by the document itself. In terms of the scenario above, if all vendors whose equipment produced temperature data agreed only on how that data would be named in the schema, they could still differ on other aspects of data representation. Indeed, even complete agreement on naming “temperature” is not strictly necessary, as the XML format makes it easy to insert “translators” that account for minor differences in how schemas are described by different vendors.

In other words, an XML-based representation for instrumentation data enables incremental progress towards interoperation without requiring full buy-in in advance. Support for XML in enterprise programming frameworks is ubiquitous and programmers skilled in the use of this technology and its rich tool repertoire are readily found.

How would existing telemetry facilities be converted to XML?

Conversion is not necessary: As long as any software interface exists that allows telemetry (transmitting of data captured by instrumentation and measuring devices to a remote station) to be received as a byte stream (whether over a serial port, proprietary interconnect, etc.), it is a simple matter to create a “wrapper” that provides an alternative XML-based representation of the same data. In other words, existing control software would not require modification in order to be integrated into the MTConnect process. In the future, as the value of MTConnect becomes clear, we would expect that equipment would “natively” export telemetry in an XML format, mirroring what has happened in the enterprise world: Microsoft Office, long a bastion of the use of proprietary formats, has converted all of its products to producing XML-based documents starting in 2006, to name only the most prominent example.

How would XML telemetry be exchanged and processed?

The Internet (and now intranet) standards for building networked applications using the ubiquitous combination of HTTP over TCP/IP would suffice to build the necessary infrastructure. Analysis software wishing to read, combine, or otherwise process the XML streams would have access to a wide variety of libraries (even numerical analysis packages like MATLAB now support XML data formats), and even if the software retained a proprietary data file format, conversion between such a format and XML would be simple for the same reasons outlined above.

Learn more about XML and MTConnect .