Astronomical Data Analysis Software and Systems X
ASP Conference Series, Vol. 238, 2001
F. R. Harnden Jr., F. A. Primini, and H. E. Payne, eds.

Converting FITS into XML: Methods and Advantages

Brian Thomas,[1] Edward Shaya,[1] and Cynthia Cheung

Goddard Space Flight Center/NASA, Code 631, Greenbelt, MD 20771

Abstract. We discuss how and why FITS data should be encapsulated in XML. Our goal is not to throw away the FITS standard entirely. Rather, we seek to re-map the FITS standard into an XML-based format. The advantages of doing so are legion and include: greater interoperability, parsing by XML aware browsers and applications, hierarchical structure for improved searchability, default values for header descriptions, extensibility for specialized usage and future development, and piggybacking on industry applications.

1. Why FITS and XML are Good for Each Other

The current life cycle of astronomical data is lived through a bizarre parade of data formats. A probable path in this cycle could be: data are taken at the telescope as FITS, they are submitted to a publisher in LaTeX, the publisher converts these data into SGML, data centers convert the SGML into some form of ASCII (such as HTML), and the end-user downloads this and converts the data back into FITS (!) for use in various analysis software.

So many format translations need not occur, since XML, the eXtensible Markup Language, could provide a reasonable interchange format to stand in at every step of the cycle. XML is designed to be data-centric (unlike HTML or SGML) with the layout/presentation of the information being provided by a separate file (often referred to as a "style sheet"). This simple divorced situation allows us then, at each step of the data cycle, to re-use the same core XML data file. Another advantage of XML is its description and validation via a DTD ("Document Type Definition"). The DTD may be used both to hold information about the data (such as keyword definitions) and to provide a template for querying databases holding the XML documents. Default values for header descriptions may be obtained from the DTD when the document is parsed, allowing the document to hold only those keywords with non-default values. All XML documents may hold URLs that point back to their respective DTDs, insuring that the correct DTD may always be found. Finally, as a recommendation of the World Wide Web Consortium (W3C), XML has the weight of the IT community at large behind it. Currently, this Internet standard is receiving massive development support that easily outstrips any spending that astronomers invest in FITS. It makes sense to leverage this resource.

[1]Raytheon ITSS, 4500 Forbes Blvd., Lanham, MD 20706

There is no mystery as to why FITS lacks many of the favorable characteristics of XML; the FITS standard was developed in the late 1970s for a different computing environment (VAX/Fortran/tape archives). While FITS has evolved, it still contains many limitations based on its origins that need not be adhered to today, including 8 character keywords, 80 character cards, a maximum of 999 table records, etc. Yet even for its limitations, FITS still contains a good understanding of the general needs of astronomy data. Because XML is designed to be the basis of other languages, it is generic in nature and will require development effort to adequately encapsulate scientific data with it.

There is no need to re-invent the wheel here. What is really needed is a marriage between XML and FITS with the goal of re-mapping (rather than redefining) the FITS standard into an XML-based format. We refer to this new hybrid data format standard to as "FITSML" and briefly discuss important characteristics of this project below.

2. FITSML: Using XDF to XMLize FITS

The Astronomical Data Center (ADC) at Goddard Space Flight Center is in the process of developing a science data interchange language in XML. This new language, XDF (eXtensible Data Format) is formulated to contain only the most basic needs of encapsulating scientific (not just astronomical) data.

We have found the translation of basic FITS data types into XDF-based FITSML is possible without content loss or redefining important FITS definitions (keywords). This is possible because the XDF data model is fairly sophisticated and allows for many types of ASCII and binary data, unevenly distributed data cubes, grouping of data with mixed data formats, and vector spaces, among other things. Furthermore, the XDF kernel is designed to allow discipline specific keywords to be included within XDF-based data formats. This property allows FITSML to look familiar to old hands. XDF also allows for two other important properties of FITS: the addition of user-defined keywords and extension of FITSML into sub-field/mission-specific varieties of FITS.

XDF provides more than a means to XMLize FITS however. It brings in new (and we feel needed) properties that FITS currently lacks. Some of the most important of these features include the following:

Inheritablity: While discipline-specific keywords may be layered on top of XDF to create new discipline specific languages these new languages still share common base conventions which are understood by the others. This property in the object-oriented world of programming is known as inheritance and provides that software designed to read FITSML will also be able to read any other XDF file, regardless of whether the data is astronomical in nature or not. Properly formulated, the structure of FITSML may be designed such that the interchange works (in a limited fashion) in reverse, e.g., any XDF software may read in FITSML.

Infinite hierarchical nature: XDF allows for both infinitely regressing hierarchical structure/array and keyword associations. This means that FITSML files may contain more sophisticated data structures than FITS which in turn provides greater freedom to application developers and

archivists to seek appropriate software solutions for storing data. As for the keywords, this kind of flexible keyword hierarchy, if properly implemented, may provide improved searchability of the data by allowing for a more "natural-language" understanding of the indexing and easier formulation of queries by a human to the database. An example of the kind of keyword hierarchies that may be held in FITSML is shown below.

```
<!-- some FITS keywords, but in hierarchy -->
   <observation>
      <telescope>VLA</telescope>
      <observer>Syke</observer>
      <imageType>object</imageType>
      <datesAndTimes>
          <observationDate>27/10/1982</observationDate>
      </datesAndTimes>
      <positions>
          <astroObject>3C405</astroObject>
      </positions>
   </observation>
   ... continues ...
</XDF>
```

Machine understandable scientific units: One of the central problems in comparing science data is the determination of compatibility from data units. All science units break down into eight principle SI units (meter, gram, second, radian, Ampere, degree Kelvin, candela) and "number". These basic units are defined in the XDF DTD and are used to build up other common scientific units. For example, the unit "Newton" can be defined as an entity:

```
<!ENTITY newton
'<unitGroup name="N">
     <apply><times/>
            <meter />
            <kilogram />
            <apply><power/><second /><cn>-2</cn></apply>
     </apply>
</unitGroup>' >
```

Unit entities may also be used to create other units, such as for the unit "Pascal" in the example below which uses "&newton;" entity in its definition:

```
<!ENTITY pascal
'<unitGroup name="Pa">
    <apply><divide/>
           &newton;
           <apply><power/><meter /><cn>2</cn></apply>
    </apply>
</unitGroup>' >
```

With units expressed as entities the FITSML parser may easily examine and decompose unit definitions into the constituent nine basic units.

3. Summary, Future Progress, and Resources

3.1. The FITSML Project and its Future Direction

XDF and FITSML are works in progress. Although we have examined the encapsulation of standard forms of astronomical data, such as images, spectra, tables, and sky atlases, we continue to examine other types of data in order to give XDF the generality we desire, and to work with the FITS community to further develop the FITSML DTD in order to meet their needs. Towards this end, we look forward to developing a prescription for FITSML to wrap legacy FITS files and to test translation of more advanced FITS data formats into FITSML. Other plans include investigating various combinations of style sheets for viewing (and perhaps editing FITSML) within web browsers, and releasing a beta software package for XDF and an alpha software package for FITSML (both releases have software written in Java/Perl) in Spring 2001. We hope to provide a beta software package and simple translation tools between FITS and FITSML before the end of 2001.

3.2. Resources

Space limitations prevent our fully describing FITSML or disclosing the FITSML DTD or samples within this brief article. Please refer to the following web pages for more information:

- XDF Homepage[2] (including links to FITSML/XDF DTDs)
- Software Download[3] (including samples of FITSML/XDF):
- ADC Homepage[4]

[2] `http://xml.gsfc.nasa.gov/XDF/`

[3] `http://xml.gsfc.nasa.gov/ADCSoftwareDownload.html`

[4] `http://adc.gsfc.nasa.gov/`