Astronomical Data Analysis Software and Systems IX
ASP Conference Series, Vol. 216, 2000
N. Manset, C. Veillet, and D. Crabtree, eds.

XML in the ADS

G. Eichhorn, A. Accomazzi, M. Demleitner, C. S. Grant, M. J. Kurtz, S. S. Murray

Harvard Smithsonian Center for Astrophysics, Cambridge, MA 02138

Abstract. The ADS software system has been rebuilt in the last few months to introduce several improvements. One major part of the re-write was to use XML for storing internal data. This provides flexibility that was not available before to merge data from various sources into a unified framework. We describe our current usage of XML and future plans on how to utilize the new capabilities.

1. Introduction

The Astrophysics Data System (ADS[1]) provides access over 1.5 million references in Astronomy, Physics/Geophysics, and Space Instrumentation (Eichhorn 1997, Eichhorn et al. 2000, Kurtz et al. 2000). These data come from many different sources in many different formats. We have recently started to convert these data into XML (Harold 1998) with full support of Unicode (Unicode 1996) character representations (Accomazzi et al. 2000). This article describes how we apply XML and what advantages we expect from its use.

2. Background

The ADS receives references and abstracts from many sources (Grant et al. 2000). This includes different versions of the same reference from several sources. In order to utilize this information efficiently, we need to retain all the different versions of a reference, identify the origin of each reference item, and decide which version of a reference item is the best to show to our users. XML provides an ideal framework for such a dataset. It allows us to store several versions of the same item (for instance sets of keywords) that are tagged with their origin.

A second part of the upgrade concerns the use of extended character sets. Author names for instance use more than just the set of characters in the Latin-1 character set. In a discipline that uses mathematical symbols extensively, titles and abstracts frequently contain special characters like Greek letters and various other symbols. These can accurately be represented as Unicode characters. This will allow us to support any character in the original data.

Another aspect of the XML system will be to separate several of the data fields in more detail. For instance the journal field currently contains all the

[1] `http://adswww.harvard.edu/`

publisher information in one string. The XML version will eventually separate this field in several subfields like volume, issue, first page, last page, year, publisher, editor, etc. in a very consistent and straight-forward manner. Once this XML system is fully implemented, it will allow both users and software tools more control over the formatting of the returned references.

3. Conversion Issues

Converting the main fields of our references into XML can be done quickly. However, in order to fully utilize the potential of the detailed XML structure, we will have to parse the current data fields in great detail. With 1.5 million data records in the database, this parsing has to be done completely automatically. This requires the development of sophisticated software that can parse the many different formats used by our data suppliers in the different fields. For the older records in the ADS this will be quite difficult. Newer data records have been generated using more standardized formats like SGML, which makes parsing much easier, but by no means trivial.

A second part of the conversion of the data that will be difficult is the recovery of non-ASCII characters like accented letters in author names and special characters (e.g. Greek letters) in the title and text. All our old data had been converted into plain ASCII characters, since at the time when these data were accumulated there was no support for characters other than 7-bit ASCII. We will develop software to help us to recover as many of the special characters as possible, but expect this to work for only some of the non-ASCII characters that were converted to ASCII. For the remainder we will have to rely on help from our users to point out references where Unicode characters need to replace their translated representation.

Because of these significant difficulties in parsing and converting the current data into XML and Unicode, we have designed the system to allows us to proceed in steps with the conversion. In the first stage we will translate the main fields into the XML system, and parse only some of them (for instance author names) into subfields. Full conversion will then take place over the next year to improve the level of detail in the record structure. During that time the system will already use XML and Unicode character support, but only a subset of the record structure in our specification will be populated.

Character conversion to Unicode will also be done incrementally during the next few months. It will start when the XML/Unicode software is in place. It can be done at any time and incrementally without affecting the functionality of the system.

4. ADS XML DTD

In order to be able to store as much detail as possible, we decided to field the different data structures in fairly fine detail. For instance author names will be fielded in Prefix, First Name, Last Name, Suffix. The journal field, as mentioned above, will also be fielded in great detail. The full description of the ADS XML Document Type Definition (DTD) and character codes that we use is available on-line at: `http://adswww.harvard.edu/pubs/XML`

To increase readability, all tag names are in uppercase, attribute names are in lowercase in conformance with other DTDs currently being developed in astronomy (not submitted Koons et al. 2000, Ochsenbein et al. 2000, Shaya et al. 2000). In general, we have used full words for names of the outer tag of a field (e.g. *AUTHORS* for the author field of the reference). Individual members of the field (e.g. individual authors) have tag names of two letters (e.g. *AU* for an individual author). This is a compromise between legibility and the length that a record will have. The XML tagging adds considerable length to the data. Shortening the tags ameliorates this somewhat.

Certain attributes can occur in many or most fields. For instance the attribute *origin* is used with many tags to describe where a particular instance of a field originated.

5. ADS Character Set and Encoding

Bibliographic data may contain many different characters. The ADS system will therefore support the complete set of Unicode characters. Several issues in connection with such support need to be addressed. The first issue, conversion of the legacy data has been described above. In this section we will briefly describe how the ADS system will handle incoming new data, how it will store the Unicode characters, and how the output of these characters will be handled.

First, we need to be able to recognize incoming special characters. We have built a table with all Unicode characters that we are aware of. This table contains the Unicode value, the corresponding entity names that we expect to see in SGML input data, the ASCII equivalent, and the equivalent TeX macros. This allows us to parse SGML input as well as TeX input and convert it into Unicode. The table contains all entities and macros that we get from different publishers. These often have different names for the same character. It represents an enormous amount of handwork, since the identification of all these entities needs to be done by hand.

The next part of the handling of special characters is the storage in the ADS XML data. We decided to store Unicode characters in the canonical form &#xxxx; rather than as a UTF-8 character combination, where xxxx is the hexadecimal representation of the value of the Unicode character. If the input parsing system should encounter an entity that is not represented in this conversion table, it will be stored in its original form. This will allow us to later convert such characters once we have assigned the proper Unicode value.

For internal handling of Unicode characters by our software system we decided to use UTF-8 encoding. Both the indexing and the search software will use this encoding for processing the data. The main reason for this choice was the availability of software that has support for a UTF-8 encoded Unicode character set,

The last part is the output of special characters in different formats. The same table that drives the input conversion is used for output conversion. For each Unicode character the output is chosen according to the selected output format. Output formats available currently are HTML, ASCII, TeX, and Unicode.

For HTML output any character that is defined in the HTML version that the browser accepts (HTML 4 or HTML 5) is output as an HTML entity in the form &#xxxx;. All other Unicode characters are converted to their ASCII representation according to our Unicode table.

For TeX and ASCII output, all Unicode characters are converted to their TeX or ASCII representation respectively, according to the table.

For Unicode output, all Unicode characters are encoded in UTF-8. This is currently not supported by browsers, but more than likely will be supported in the not so distant future.

6. Schedule

We are currently in the process of preparing our data for the XML conversion. This includes cleaning up any HTML tags and escaping other mathematical expressions that mimic XML entities (for instance <b> meaning the mean value of b). We expect to start converting the data into XML by the end of 1999. The new software system that uses XML records is scheduled to be in place by February, 2000. In the months after that we will continue to parse the fields in more detail. We expect to have this process completed by the middle of 2000.

7. Acknowledgment

This project is funded by NASA under NASA Grant NCC5-189.

References

Accomazzi, A., Eichhorn, G., Grant, C. S., Kurtz, M. J., & Murray, S. S. 2000, A&AS, in press

Eichhorn, G. 1997, Ap&SS, 247, 189

Eichhorn, G., Kurtz, M. J., Accomazzi, A., Grant, C. S., & Murray, S. S. 2000, A&AS, in press

Grant, C. S., Eichhorn, G., Accomazzi, A., Kurtz, M. J., & Murray, S. S. 2000, A&AS, in press

Harold, E. R. 1998, Xml: Extensible Markup Language (IDG Books Worldwide)

Kurtz, M. J., Eichhorn, G., Accomazzi, A., Grant, C. S., Murray, S. S., & Watson, J. M. 2000, A&AS, in press

Ochsenbaum, F., Albrecht, M., Brighton, A., Fernique, P., Guillaume, D., Hanisch, R., & Wicenec, A. 2000 this volume, 83

Shaya, E., Gass, J., Blackwell, J., Thomas, B., Holmes, B., & Cheung, C. 2000, this volume, 87

Unicode Consortium 1996, The Unicode Standard: Version 2.0 Addison-Wesley Pub Co. (Reading, MA)