Astronomical Data Analysis Software and Systems XIV **O4-1**
ASP Conference Series, Vol. XXX, 2005
P. L. Shopbell, M. C. Britton, and R. Ebert, eds.

Configurable format conversion tool from XML to FITS with binary extensions

K. Matsuzaki, H. Baba[1], K. Shuto and A. Miura

Institute of Space and Astronautical Science, Japan Aerospace Exploration Agency, 3-1-1 Yoshinodai, Sagamihara, Kanagawa, Japan, Email: matuzaki@plain.isas.jaxa.jp

H. Nakamoto, R. Kanamori and Y. Kobayashi

Systems Engineering Consultants Co., 22-14, Sakuragaokacho, Shibuya,Tokyo, Japan

Abstract. We have developed a configurable format conversion tool from XML document to FITS document with binary extensions. Structure of input data of this tool is specified by an XML schema language – RELAX NG – and that of output is also specified by a schema described in XML. This tool is configured by a definition of simple mapping created from schema of input and output. Since RELAX NG is general purpose language and this tool accepts mapping of arbitral data item in XML data structure, we expect this tool can be widely used in data processing in the astronomy.

1. Introduction

In development of complex data systems, designer often encounters with several document formats determined by external requirements. For example, application of virtual observatory (VO) must handle with XML document received from web service and may create FITS file used by observers in down stream. Although conversion of document format itself does not add new information item, development and testing of software take some amount of time. To date, there are few solutions to connect ‘ modern ’ XML technologies with ‘ traditional ’ FITS file processing easy enough in the sense of development of system. Thus, we have decided to develop a configurable format conversion tool from XML document to FITS document with binary extensions toward incoming era of VO. We named this tool XML2FITS.

There are many different types of document format. Types of format conversion spread over the matrix of these formats. In order to minimize incoming effort for format conversion in general, we have designed specification of conversion with XML2FITS to be subset of specification of conversions that cover much wider document formats (e.g. text and binary document). Although we deal

[1] Waseda University

with various types of document format, there were no suitable schema languages which covered wide range of document format. Thus, we started from definition of framework for defining schema language named Mapping Schema. In the next section, we describe parts of Mapping Schema required for understanding of XML2FITS.

2. Mapping Schema

2.1. Schema of input and output

In developments of software, description of interface is often the most important. We can strictly describe interfaces of input and output with schema language only strict enough. Parameter specification of format conversion on framework of Mapping Schema also starts from description of schema of input and output.

For XML document, there is a standard schema language RELAX NG which is endorsed by regular tree grammar. We have designed Mapping Schema so that Mapping Schema of XML document is exactly compatible with RELAX NG. Correctly speaking, we intend to extend philosophy of RELAX NG carefully into other document formats. In the case of RELAX NG, several syntaxes for presentation of a schema exist. In this paper, we concentrate our description on schema presented by XML (the XML syntax). In the rest of this section, we describe how we divide language of RELAX NG into common portion and XML document specific portion together with how we introduce FITS document specific portion.

A schema of Mapping Schema consists of a nest of schema elements. In the XML syntax, one schema element is presented by one element of XML. There are several species of schema element. In the XML syntax, species of schema element is specified by name of XML element. A schema element can have attribute. In the XML syntax, name and value of attribute is presented by name and value of attribute of XML element, respectively. All of document formats share species of schema element in the core definition of Mapping Schema. Individual document format has additional species of schema element. The core definition and format specific definitions restrict allowed combination of nest of schema element.

Structure element is a kind of schema element that specifies conditional appearance of information item. Structure element belongs to the core definition. There are several species – `zeroOrMore`, `oneOrMore`, `optional`, `choice`, and so on. Some document format has its own specific way to present some of these structures. For example, FITS document has several rows in a binary extension. Repeat (of row) is described by `zeroOrMore` in a schema.

Container is a kind of schema element that specifies how data is hold by an instance of schema (i.e. input or output file). Each type of document format has its specific containers. There are two species, `element` and `attribute`, in XML portion of Mapping Schema. There are several species, `primaryHeaderValue`, `primaryHeaderComment`,`BTEHeaderValue`, `BTEHeaderComment` and `BTEData` in FITS portion of Mapping Schema.

Sub container is a kind of schema element which specifies data itself. There are two species, `data` and `value`, in the core definition of Mapping Schema. `data` has an attribute `type`. It is possible to restrict allowed range of data with

```
<element name="data"
  datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">
    <zeroOrMore>
        <element name="rec">
            <attribute name="t">
                <data type="dateTime"/>
            </attribute>
            <element name="Voltage">
                <data type="double"/>
            </element>
        </element>
    </zeroOrMore>
</element>
```

Figure 1. Sample of schema of XML document

```
<BTE
  datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">
    <zeroOrMore>
        <BTEData ttype="t" tunit="sec">
            <data type="dateTime"/>
        </BTEData>
        <BTEData ttype="doubleItem" tunit="Volt"
          ttypeComment="Double Precision Value">
            <data type="double"/>
        </BTEData>
    </zeroOrMore>
</BTE>
```

Figure 2. Sample of schema of FITS document

schema element `param` in the core definition. Allowed value for `type` and `param` can be extended by external datatype library.

We show sample of schema for XML document and FITS document presented by Mapping Schema in Figure 1 and Figure 2, respectively. Valid instance of these examples hold same table with rows consist of a time and voltage. Two documents differ in capability of explanation of column. XML document only gives some hint by name of element. FITS document with binary extension can have unit and comments in header area.

2.2. Definition of Mapping

What we want to preserve between input and output of format conversion is data itself. In a schema of Mapping Schema, intrinsic structure and order of data can only be defined by schema elements in the core definition. Such core definition only schema can be easily obtained from a schema of specific document format by removing container preserving schema elements inside.

```
<fits:BTE
 xmlns:fits="http://isas.jaxa.jp/ums/xmlfits/20040825/fits" >
  <element name="data"
   xmlns="http://isas.jaxa.jp/ums/xmlfits/20040825/xml">
    <ums:zeroOrMore
     xmlns:ums="http://isas.jaxa.jp/ums/xmlfits/20040825"
 datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">
      <element name="rec">
        <fits:BTEData ttype="t" tunit="sec">
          <attribute name="t">
            <ums:data type="dateTime"/>
          </attribute>
        </fits:BTEData>
        <element name="Voltage">
          <fits:BTEData ttype="doubleItem" tunit="Volt"
           ttypeComment="Double Precision Value">
            <ums:data type="double"/>
          </fits:BTEData>
        </element>
      </element>
    </ums:zeroOrMore>
  </element>
</fits:BTE>
```

Figure 3. Sample of simple mapping definition

When we keep schema elements in the core to be the same between input and output, we can easily combine two schema files into a file of simple mapping definition. Conversion of XML2FITS is specified by such single mapping definition. When you use tool for simple mapping, intrinsic manipulation of data must be decoupled with format conversion and must be performed ether before input or after output in any other technology (e.g. using XSLT for XML).

Each schema element of a simple mapping definition belongs to one of three spaces 1) space for input data which does not appears on output, 2) space for data transferred from input to output, 3) space for output data which does not appears on input. From simple mapping definition, we obtain schema of input data by selecting schema elements in former two spaces. We obtain schema of output data by selecting schema elements in latter two spaces. In the XML syntax, we allocate one XML namespace for one space. URI which can be processed by a given tool may differ from a tool to another tool. For example, initial release of XML2FITS utilizes following three spaces

- http://isas.jaxa.jp/ums/xml2fits/20040825/xml,
- http://isas.jaxa.jp/ums/xml2fits/20040825,
- http://isas.jaxa.jp/ums/xml2fits/20040825/fits,

respectively.

```
<fits:BTE
 xmlns:fits="http://isas.jaxa.jp/ums/xmlfits/20040825/fits" >
  <element name="data"
   xmlns="http://isas.jaxa.jp/ums/xmlfits/20040825/xml">
    <ums:zeroOrMore
     xmlns:ums="http://isas.jaxa.jp/ums/xmlfits/20040825"
 datatypeLibrary="http://www.w3.org/2001/XMLSchema-datatypes">
      <element name="rec">
        <fits:BTEData ttype="t" tunit="sec">
          <attribute name="t">
            <ums:data type="dateTime"/>
          </attribute>
        </fits:BTEData>
        <element name="Voltage">
          <fits:BTEData ttype="doubleItem" tunit="Volt"
           ttypeComment="Double Precision Value">
            <ums:data type="double"/>
          </fits:BTEData>
        </element>
      </element>
    </ums:zeroOrMore>
  </element>
</fits:BTE>
```

Figure 3. Sample of simple mapping definition

When we keep schema elements in the core to be the same between input and output, we can easily combine two schema files into a file of simple mapping definition. Conversion of XML2FITS is specified by such single mapping definition. When you use tool for simple mapping, intrinsic manipulation of data must be decoupled with format conversion and must be performed ether before input or after output in any other technology (e.g. using XSLT for XML).

Each schema element of a simple mapping definition belongs to one of three spaces 1) space for input data which does not appears on output, 2) space for data transferred from input to output, 3) space for output data which does not appears on input. From simple mapping definition, we obtain schema of input data by selecting schema elements in former two spaces. We obtain schema of output data by selecting schema elements in latter two spaces. In the XML syntax, we allocate one XML namespace for one space. URI which can be processed by a given tool may differ from a tool to another tool. For example, initial release of XML2FITS utilizes following three spaces

- http://isas.jaxa.jp/ums/xml2fits/20040825/xml,
- http://isas.jaxa.jp/ums/xml2fits/20040825,
- http://isas.jaxa.jp/ums/xml2fits/20040825/fits,

respectively.