

International Civil Aviation Organization



**THE SIXTH MEETING OF AERONAUTICAL
TELECOMMUNICATION NETWORK (ATN)
IMPLEMENTATION CO-ORDINATION GROUP
OF APANPIRG (ATNICG/6)**



Seoul, Republic of Korea, 16 - 20 May 2011

Agenda Item 6: Applications

EXTENSIBLE MARKUP LANGUAGE (XML)

(Presented by the Secretariat)

SUMMARY

Extensible Markup Language (XML) has been adopted for the exchange of OPMET and AIM data. This paper provides information on XML and discusses various attributes. It is expected that this paper will provide a preliminary information and help in implementing XML over AMHS.

1. INTRODUCTION

1.1 Air Navigation Commission, in its session held on 25 May 2010 concluded to resume the work relating to the migration to table-driven codes with the aim of meeting the needs of the NextGen and SESAR programmes using WXXM based on the use of XML. Furthermore, it was suggested that the use of BUFR code-form in Annex 3 for the exchange of METAR/SPECI and TAF under bilateral agreement be replaced by the use of XML as a part of Amendment 76 to Annex 3 as a first step.

1.2 Through Information Paper 4 presented to the Fifth meeting of ATNICG held from 31 May to 4 June 2010, Secretariat presented the status of development of XML and invited the meeting to assess if AMHS can be used for exchange of MET data on XML. Through Action Item 17, USA was requested to prepare a Working Paper for presentation to CNS/MET SG-15 through ATNICG/6 addressing the requirement of standards for using XML over ATN for the transmission of OPMET data.

1.3 USA, through Working Paper 11, presented to the Eighth Working Group Meeting of ATNICG held in Christchurch, New Zealand from 28 September to 1 October 2010 informed about the background information on Web Services Simple Object Access Protocol (SOAP) Messaging and the WXXM Weather Data Model. Results of the first phase of testing on delivery of XML based OPMET messages over AMHS conducted between USA and Hong Kong China in June 2010 were presented to the same meeting. Meeting was informed that as the next step, the testing will be

expanded to a tri-partite test between USA, Hong Kong China and Singapore. The timeframe of this tri-partite test is yet to be finalized.

1.4 In the Ninth Meeting of the Asia/Pacific OPMET Management Task Force held in Bangkok from 21 to 23 March 2011, USA through its Information Paper 7 discussed the transitional arrangements of Extensible Markup Language (XML)-based OPMET messages over Air Traffic Message Handling System (AMHS). Paper discussed limitations of AFTN in handling XML format and came to a conclusion that exchange of XML-based OPMET data, through the use of AFTN (with limitations) and AMHS was feasible, though a careful planning and attention to configuration will be required.

2. DISCUSSION

2.1 HyperText Markup Language (HTML), a markup language is being used quite extensively on the web. Other markup languages like Extensive Markup Language (XML) and XHTML have now been developed to meet the specific requirements.

2.2 XML has been adopted for the exchange of OPMET and the AIM data globally. Trials were conducted to assess the performance of XML over AFTN and it was observed that there were a number of restrictions. XML trials over AMHS conducted recently in the region indicate a much better performance. The trials will be carried further.

2.3 In the Attachment to this Information paper, the concept behind Extensible Markup Language, which is proposed to be adopted for the transmission of OPMET and AIM data is being discussed. Attachment also discusses some of the benefits of using XML.

3. CONCLUSION

3.1 Extensible Markup Language (XML) has been adopted for the exchange of OPMET and AIM data and transmission of information in XML format is being promoted in the region. Information provided in this paper, it is expected will help the participants in understanding the context of the issue.

4. ACTION REQUIRED BY THE MEETING

4. Meeting is invited to note the information provided on Extensible Markup Language and support exchange of OPMET and AIM data in XML format over AMHS.

Extensible Markup Language (XML)

1. Introduction

HyperText Markup Language (HTML) is a markup language which is being used quite extensively on the web. The initial idea behind the implementation of a markup language (HTML) was modest: it was meant to describe only basic page content, providing rudimentary control over how that content should appear in the browser. In fact HTML was established to perform following basic tasks only:

1. Allow tight control over document display
2. Provide the flexibility to describe different, specific types of information and data
3. Convey information from a variety of media and in various formats
4. Define complex linking relationships between documents
5. Publish a single set of information across a variety of media

1.1 Web designers found these tasks very useful but restrictive. Main restrictions are

1. HTML lacks fine control: It does not include any mechanism for fine control. For example, one can not specify a document's display size or control the size of a browser window. User's monitor size and display settings can dramatically affect how a browser displays HTML documents. HTML 4.01, has introduced some provision to control the font.
2. Display vary: Web pages on different platforms such as Windows XP, Unix or Mac OS are displayed differently.

1.2 To provide some handle to the users, XML (Extensible Markup Language) and XHTML were subsequently developed. HTML, XML and XHTML, in fact represent stages in the development of markup languages. HTML, designed to display contents in the web browser came first. It uses *markup* tags, but these specialized bits of markup are limited to a predefined set created by the W3C (Worldwide Web Consortium). XML, intended for data exchange came next. Although the rules of XML syntax are also defined by W3C, the tags are however defined by the creator of a specific XML document. Then came XHTML – which uses the markup tags of HTML and the strict syntax of XML, and is considered a transition language between HTML and XML.

1.3 As has been mentioned earlier, HTML is a markup language which includes a set of predefined tags that format information for display on the web. XML, on the other hand, does not have any predefined tags – instead the tags can be created on XML to structure the XML document so that on display its contents are in a form that meets the expected needs. Basically one designs one's own custom markup language (actually an XML application) to exchange data in a way that suits the requirements best. Though XML does not include predefined tags, but it does include very specific rules about the syntax of an XML document.

1.4 HTML and XML were in fact designed with different goals and hence it cannot be said that XML replaces HTML. They are similar but not identical. HTML and XML both use tags and attributes. But whereas HTML defines basic text elements and includes defaults tags for how the text may be displayed in a browser window, XML tells us only what each element means. XML says nothing about how the

elements should or must be displayed – XML separates content and the presentation of that content. Unlike HTML, XML is not limited to any fixed set of tags or element types. That besides, where as

- XML was designed to transport and store data, with focus on what data is
- HTML was designed to display data, with focus on how data looks

Note: Tag is the opening <p> and closing expression in the statement, element is <p>text<p>. An element includes the opening and closing tags of a tag pair – and everything in between.

There are some advantages of HTML over XML, like it can create a reasonably consistent web-page presentation for the users. In addition, HTML is quick, easy and cheap. Anybody can create an HTML document by using a text editor. It is very easy to create a fairly good webpage on HTML. Let us see what XML gives us over HTML.

2. Extensible Markup Language (XML)

2.1 XML is a *markup language* – it uses tags to label, categorize, and organize information in a specific way. *Markup* describes documents or data structure and organization. *Content*, such as text, images, and data is that part of the code that the markup tags contain; it is also what is of greatest interest to the addressee who reads or interacts with data or document.

XML is not limited to a particular set of markups, one creates one's own markup to suit one's data and document needs. XML can be used to send information to many types of applications like the mobile phone or a web browser. It can also be used to customize the information so that it is displayed appropriately on many types of devices.

Benefits

There are many benefits of using XML

- a) Unlimited elements: One can create one's own elements and attributes instead of working with restricted, predefined set (like in HTML)
- b) Structured data: Different applications can extract information that they need from XML document
- c) Data exchange: XML enables exchange of database contents and/or other structured information across the internet or between dissimilar applications
- d) XML complements HTML: XML data can be used in HTML pages
- e) XML documents are well formed: XML documents must follow certain rules. This consistency makes such documents easier to read and create
- f) Self describing: No prior knowledge of an XML application is needed. Of course, knowing HTML can really help in understanding more about XML

- g) Search Engines: XML delivers a noticeable increase in search relevance because it provides ample contextual information and explicit labels for document elements
- h) Updates: No need to update an entire site page by page: the Document Object Model (DOM) built into XML documents permits individual elements to be accessed and updated.
- i) User selected view of data: Different users can access different information or can present the same information in various ways.

Let us see, what it really implies in real-life usage:

1. Separating data and context: One of the benefits of using XML is that it automatically separates data from the context. An XML document, by itself does not include any instruction as to how the contents will be displayed, it only defines the structure of the document, a *stylesheet* however can be added to provide information on *style* – formatting instructions for displaying the contents. This is a very big advantage, where one can change the display instructions without having to make any changes to the XML document.
2. XML enables collection of information once and reuse it in a variety of ways, not requiring to manipulate the information subsequently for usage in different context
3. XML is not limited to one application format. One can design an XML document that allows one to collect data online for use in all other documents, databases and spreadsheets. For example, if NOTAM data has been collected for all the facilities of all the stations, then the same data can be used to create station NOTAM records or facility NOTAM records etc. in a variety of formats

Microsoft Office 2003 (and possibly all the subsequent versions) includes XML tools for office applications. Using this, office document can be created in XML format and information tagged and collected for re-use in other office applications as well as on the web.

3. XML Applications

a. Classifying information

One of the most useful applications of XML is in classifying information. For example the NOTAMS can be classified into many heads like

- i) Region
- ii) Station
- iii) Facility
- iv) Class
- v) Duration
- vi) Originator
- vii) Authentication

Tags can be created to classify this information. Thus giving 'tags', which are meaningful to the contents making it easier to search for information.

- b. XML is very good in creating rules for the data format. Like, we can impose number format for the NOTAM number, date format for duration, selection for NOTAM class etc.

- c. XML documents are not limited to any particular form of output, they can end up in a variety of different places in the desirable form – for example in a database, a computer monitor, a printer etc. Post-processing, the process of taking information from a document and using it in some other programme is a very big facility with XML. For example, a NOTAM application created in XML can provide input to other applications like pre-flight bulletin automatically. Or the information can be taken to a speech synthesizer to produce input for a ATIS broadcast etc. In short, data in XML format can be used by many applications which can read XML for different purposes.

What XML is not

- a) XML is not designed for web pages only. In fact, without stylesheet, an XML message will look just like any other programming language on display. XML is a markup language that allows the user to organize information by creating tags to construct a specific document structure. Unlike HTML, XML documents are not limited to web. For display on web, it has to use another language like the CSS or XML technology (XSLT) to format the display.
- b) Though XML is a collection of data, but it is not a database management system as powerful as Microsoft Access. It has some DBMS functions like storage, queries etc. but it does not have many others like security, indexes etc.
- c) XML is a markup language and not a programming language. A markup language is essentially descriptive, a programming language is for issuing logical commands. Programming language, for example includes variables, data-types, operators, loops, functions, conditional statements etc. XML does not have any of these features.

Building an XML document

In XML, for getting the tags in a row, regular text editors (such as notepad) can do the job quite satisfactorily. If Microsoft Windows is being used, notepad can be accessed by choosing Start → Programs → Accessories → Notepad. After the file has been created, it can be saved with .xml extension. For advanced applications, regular XML editors are also available. There are many such editors like XMLSpy, Turbo XML, XML Pro which can add markup to text in a very convenient way. All XML editors provide the capability to select text with a cursor and choose which markup one wants to apply from a menu.

For many applications, XML editors can determine which element types can appear in certain contexts. In this way, the possibility of syntax or structure mistakes can be avoided. For example, if it is specified that NOTAM number element can only be in the header of the NOTAM and cannot figure within the NOTAM text, then the editor will take care that accidental mistake of inserting this element in the text is not made.

XML Applications

Data is generally generated in many formats, like it could be in the form of a text reports, it could be a set of forms, it could be spreadsheets etc. Using this variety of data in different processes is often difficult. Like if the data is in text format, a spreadsheet programme that can create a graph from the same information may not be able to get at it, that will require retyping the data all over again, which is not only in-efficient, but may also result in errors. In XML, however it is possible.

Creating XML enabled web-pages

In XML, the content (that is the data) is separated from its context (the way it is presented). That means formatting information is required to be added to make the data-file presentable. One of the methods is to link the XML document to a style-sheet, say CSS (Cascading Style Sheet) which would make the information easier to read as well as visually more presentable. Another alternative is to use XSLT (Extensible Style-sheet Language Transformations). Following example (Figure 1) shows the raw XML file in the left and the figure on the right shows the same file when linked with XSLT

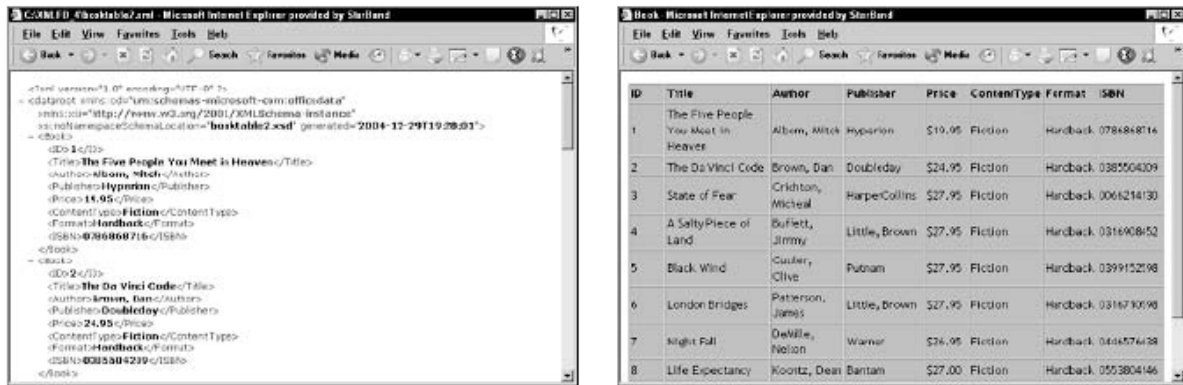


Figure 1: An XML raw file (left) and its presentation when linked with XSLT (right)

XML Group

XML includes a family of XML based language formats. Some members of the family are:

1. XLink and XPointer: XLink allows incorporation of sophisticated linking mechanism in XML documents, which goes far beyond the hyper-linking available on HTML. XPointer enables travelling to a specific item in a document by specifying element types, attribute values, character content, position etc. These two are not yet being supported by any of the present day browsers but are likely to be available in future.
2. XSLT, SPath and XSL-FO: These three are part of the Extensible Style Language (XSL). XSLT (XSL Transformations) transforms raw XML into complex display formats such as tables and indexes. XPath is an XML language to navigate an XML document. It is based on viewing XML document as a tree of nodes and using this node structure to navigate the document. XPath is used both with XSLT and XPointer. XSL-FO (XSL Formatting Objects) is used for completely formatting the layout and style of the documents that are rendered in print format.

3. XForms: These are XML language created forms to collect and submit information as XML data. XForms use both XPath and XML schemas.
4. XML Encryption and XML Signature: XML Encryption is an XML language developed for secure exchange of XML data. XML Signature is also used for secure data exchange. It provides syntax and processing rules for digital signatures.
5. XML Query: It is an XML language designed to query (request information from) any collection of XML data.
6. SOAP: (Simple Object Access Protocol) is an XML language used for communication between a Web Page requesting a Web service and the Web service application.
7. SVG and SMIL: SVG (Scalable Vector Graphics) and SMIL (Synchronized Multimedia Integration Language) are XML languages for multimedia. SVG enables display of 2-dimensional vector graphic images and animation from XML code (Vector graphics uses mathematical formula to create images) SMIL is used for integrating text, images, audio, and video content for multimedia presentations.

Types of data that can be stored in XML

There are two main groups of data that can be stored in XML – data intensive and the text intensive.

- a) Each collection of data intensive information consists of more or less arbitrary number of record structures, each containing
 - a. A unique identifier or key to help locate individual records
 - b. A common collection of named, organized values
- b) On document or text end, the content to be captured and represented fits typical notions of text or hypertext materials – that is, a collection of words, graphics and other information meant to be read or viewed as a structured object.

Viewing an XML Document

As has been explained earlier, without formatting instructions, XML documents are displayed as plain text on a web page – not particularly useful way to present data on the website. Adding Cascading Style Sheet (CSS) to the XML pages is an easy way of creating a specific appearance of the XML pages on the web. CSS is a stylesheet, which tells the computer as to how to format various elements of XML document when it is presented on the screen. It provides a method of defining display properties such as font family, text color, background images and colors and position of each element. Cascading, here refers to the capability of applying multiple stylesheets to any document – in a hierarchy of importance. If there is a conflict between two stylesheets, then the one with heavier weight or the one which is higher on the hierarchy list takes the precedence. The stylesheets can be embedded, that is the stylesheets which are defined in the document itself, or they can be external one created by the user to meet his/her preference. Following is the priority order of the stylesheets for XML, XHTML and HTML:

1. User defined stylesheets (a specific stylesheet created for the user)
2. Inline styles (created with HTML's style attributes and does not serve XML)
3. Embedded styles (contained within the head section of an HTML/XHTML document)
4. External Stylesheets
5. User's preferences
6. The browser's default stylesheet

CSS is a set of rules that are used to create stylesheets – specifically, those that web browsers recognize and can work with.

Basic CSS formatting:

With the basic CSS, one can control the format and display of

- Color and background
- Fonts and text
- Lists
- Margins, padding and borders

Subsequently, in 1998 following new controls were added:

- Elements positioning
- Element visibility
- Support for specifying page breaks
- Table styles
- Aural stylesheets (used for defining how the page sounds when read in a speech enabled browser)
- Support for system colors and fonts
- Counters and automatic numbering

CSS contains *statements*, where each *statement* includes *selector* which specifies which element the *statement* applies to and a *declaration* which specifies which style property to apply. Let us start with an example as to how the CSS is used:

Example on book list:

```
<?xml version="1.0" encoding="UTF-8"?>
<books>
  <book>
    <title>The Da Vinci Code</title>
    <author>Brown, Dan</author>
    <publisher>Doubleday</publisher>
    <price>$24.95</price>
    <contentType>Hardback</contentType>
    <isbn>0385504209</isbn>
  </book>
  <book>
    <title>State of Fear</title>
    <author>Crichton, Michael</author>
    <publisher>Harper Collins</publisher>
    <price>$27.95</price>
```

```
<contentType>Fiction</contentType>
<isbn>07868716</isbn>
</book>
<book>
  <title>Night Fall</title>
  <author>Demille, Nelson</author>
  <publisher>Warner</publisher>
  <price>$26.95</price>
  <contentType>Fiction</contentType>
  <isbn>0446576638</isbn>
</book>
</books>
```

This document includes the following elements

- ✓ <books> it defines everything in the document and its style rule should include margin information for the entire document and other common specifications like color and base font
- ✓ <book> here it should be taken care that data for each book must be separated from the others, so that it is easy to read
- ✓ <title> may be a larger font with different color will be required to emphasize the importance of this element
- ✓ <author>, <publisher>, <price>, <contentType>, <format> and <isbn> should probably be displayed in the same size color etc.

So the stylesheet can probably look like follow:

The stylesheet "books.css" for the data "books.xml"

```
books{
display: block;
width: 100%;
background-color: gray;
background-image: url (officebooks.gif);
background-repeat: no-repeat;
background-position: 80% 10px;
margin: 0;
font-family: Verdana, Geneva, Arial, Helvetica, sans-serif;
}
book {
display: block;
background-color: Silver;
width: 50%;
margin-bottom: 20px;
margin-top: 15px;
margin-left: 10px;
)
author, publisher, price, contentType, format, isbn {
display: block;
background-color: black;
font-size: x-small;
```

```
padding: 2px;
}
title {
display: block;
color: maroon;
font-size: medium;
)
```

After, the stylesheet has been created, the next step is to use them with the XML file. To reference a CSS stylesheet in an XML document, a processing instruction is provided.

```
<?xml-stylesheet href="url" type="text/css"?>
```

For the present case, the statement will be

```
<?xml-stylesheet type="text/css" href="books.css"?>
```

To see the xml document in the stylesheet defined, save both the documents in the same directory.

Glossary of Terms

Attribute: In XML, a property associated with an element that is also a named characteristic of the element. An attribute also provides additional data about an element, independent of element content. For example: <NOTAM location = "VAAH"> ILS</NOTAM>. In this case, the element (NOTAM) content is ILS, but the attribute content (location) provides additional information.

Document Type Definition (DTD): This is a statement of rules for an XML document that specifies which *elements* (markup tags) and *attributes* (names and values associated with the specific elements) are allowed in the document. It also governs the order in which the elements and attributes may appear

Metalanguage: language used to communicate information about a language itself; XML can be considered to be metalanguage, because it can be used to define other markup languages

Nesting: Ordering of elements that opens and closes a child element before its parent element is closed (child element nests within the parent element).

Schema: An XML based statement of rules that represent how an XML document models its data and defines its elements (or objects), their attributes (or properties), and relationships between elements

Syntax: Rules that govern the correct construction of intelligible statements in a markup language

Tag, Empty Tag: Markup used to enclose an element's content. An *empty element* employs a single tag; a *regular element* (which is not empty) has an opening and a closing tag

Valid: a document is valid if it adheres to the rules outlined in the associated DTD or schema document

Well formed: a markup language document that adheres to the syntax rules for XML – which are explicitly designed to make documents easy for a computer to interpret