Compressing Aviation Data in XML Format

Hemil Patel [1], Derek Lau [2] and Deepak Kulkarni
NASA Ames Research Center
MS 269/1 Moffett Field CA 94035
patel@email.arc.nasa.gov

Abstract - Design, operations and maintenance activities in aviation involve analysis of variety of aviation data. This data is typically in disparate formats making it difficult to use with different software packages. Use of a self-describing and extensible standard called XML provides a solution to this interoperability problem. XML provides a standardized language for describing the contents of an information stream, performing the same kind of definitional role for Web content as a database schema performs for relational databases. XML data can be easily customized for display using Extensible Style Sheets (XSL). While self-describing nature of XML makes it easy to reuse, it also increases the size of data significantly. Therefore, transferring a dataset in XML form can decrease throughput and increase data transfer time significantly. It also increases storage requirements significantly. A natural solution to the problem is to compress the data using suitable algorithm and transfer it in the compressed form. We found that XML-specific compressors such as Xmill and XMLPPM generally outperform traditional compressors. However, optimal use of Xmill requires of discovery of optimal options to use while running Xmill. This, in turn, depends on the nature of data used. Manual discovery of optimal setting can require an engineer to experiment for weeks. We have devised an XML compression advisory tool that can analyze sample data files and recommend what compression tool would work the best for this data and what are the optimal settings to be used with a XML compression tool.

1. INTRODUCTION

Aviation problem-solving activities include engineering troubleshooting, incident and accident investigation, routine flight operations monitoring, safety assessment, maintenance procedure debugging, and training assessment. A variety of information is typically referenced when one is engaged in these activities. Some of this information includes flight recorder data, maintenance data, pilot logs, weather data, air traffic control information, safety reports, surface data, manufacturer data sheets, and FAA advisories. This data is typically in disparate formats making it difficult to use the data with other software packages and applications. The use of a self-describing and extensible standard called XML [6] provides a solution to this interoperability problem. XML provides a

[1] Science Applications International Corporation (SAIC)
[2] De Anza College (Work done while the author was an intern at NASA Ames Research Center)

standardized language for describing the contents of an information stream, performing the same kind of definitional role for Web content as a database schema performs for a relational database. XML data can be easily customized for display using Extensible Style Sheets (XSL). While the self-describing nature of XML makes it easy to reuse, it also increases the size of data significantly. Therefore, transferring a dataset in XML form can increase both data transfer time and storage requirements significantly. A natural solution to this problem is to compress the data using a suitable algorithm and transferring it in the compressed form.

There are a few tools available for compressing XML data. Of these, Xmill [2] and XMLPPM [7] are widely used. Xmill groups XML fields based on their name and path, ahead of compression. Hence, Xmill usually does much better compression than conventional compressors such as gzip.

The Xmill compression can be fine-tuned by several options. Two important categories of options are grouping and semantic options. Grouping options [8] specify which fields should be grouped together during compression. With semantic options [8], the user can also specify how to "pre-compress" the specific text item.

Manually investigating both the compression program and corresponding option set best suited for a particular DTD is at best a trial and error process that requires a person in the loop, which is time consuming. We have devised an XML Compression Advisory Tool called XCAT, it can analyzes sample data files in a particular domain and rooted to a common DTD, it then establishes the compression tool that is best suited for this data and along with the optimal options needed. The rest of this paper is organized as follows. In the next section of the paper, we will describe the method used in XCAT. In the section 3, we will describe the results of using the XCAT on different data set. The final section provides a conclusion.

2. XCAT method

The Input to XCAT can be a user-selected set or a single file of a particular DTD from a given domain. The output from XCAT is an analysis of which compression method is best suited for each individual sample file as well as which method will best serve the group of files as a whole. After XCAT has completed its analysis, the user may choose the recommended methods of compression appropriate for single or for a group of files.

To determine the best compression method for one or more files, XCAT executes compression programs against the file(s), recording the run time and the size of the resulting compressed file, it, then determines which method was optimal. In the case of XMLPPM, it has no user definable options, so XCAT simply calls XMLPPM and records the performance. On the other hand Xmill, does have expandable user-definable compression options corresponding to each xml data fields. To determine which of these options is best suited, XCAT must interpret the file structure and develop options for all applicable fields as determined by their data types.

XCAT analysis consists of three processes shown in Figure 1.

1. The file structures are mapped in order to gain knowledge of the fields and their data
2. The fields are empirically tested
3. File compression options are combined to find the best group option.

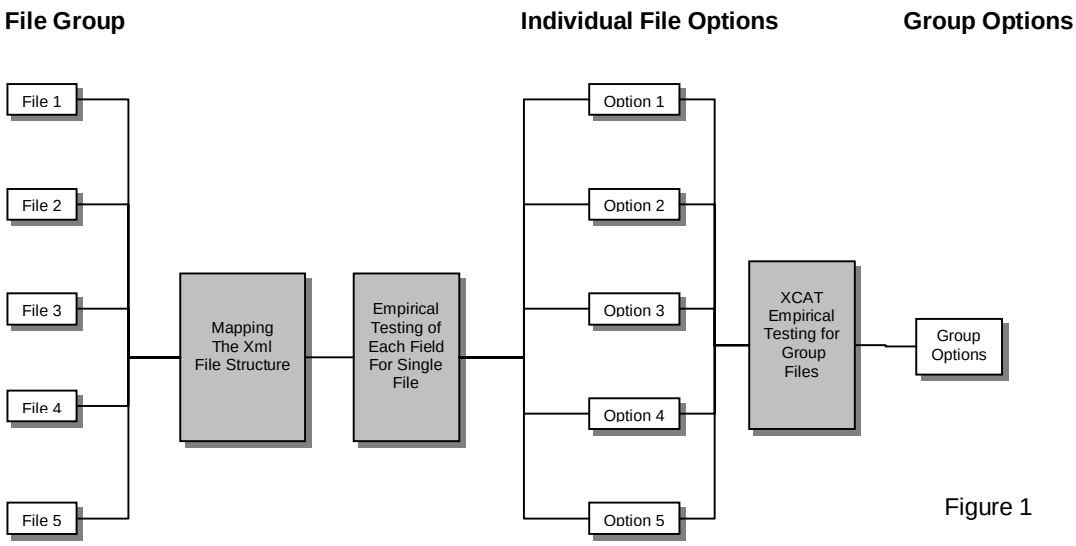

Figure 1

2.1 Mapping The Structure

To comprehend the structure of an XML file and develop Xmill's user defined options, XCAT will read a file and determine the fields and corresponding data content. To gain knowledge about each individual field XCAT records sample data to determine which possible options will correspond to each data field. Presumably, XCAT will be dealing with large files, so this sample data is recorded in a numbers respective to the file size and are recorded evenly spaced throughout the file. Once the entire file has been read, the basic file structure has been mapped and sample data has been collected for every data field. Each field is categorized by their primitive types (*Integer, Floating Point Number, String, and Alphanumeric String*). Figure 2 shows a sample xml file structure and the field classification generated by XCAT.

Xml File Structure

```
<record>
        <key_ATIS>17</key_ATIS>
        <UTC_time>4/11/2002 12:43:00 AM</UTC_time>
        <ICAO_id>LAX</ICAO_id>
        <altimeter_ins>3003</altimeter_ins>
        <wind_dir>240</wind_dir>
</record>
```

Field Classification

```
Key Atis       = Integer
UTC_time       = Alpha Numeric
ICAO_id        = String
altimeter_ins  = Integer
wind_dir       = Integer
```

Figure 2

2.2 Empirical Testing

Once all the field types have been determined, an empirical test is performed for all possible compression options against each field type recording the resulting compressed file size. At the present, XCAT only handles numeric fields which consist of integers and floating point numbers. Integers are handled simply by using the following numeric options that Xmill offers.

di - Delta compressor for signed integers
i - Compressor for signed integers
u - Compressor for unsigned integers
u8 - Compressor for integers between 0 and 255

Figure 3

Additional parameters are tested with the numeric options specifying the minimum number of digits. Floating-point numbers are separated by the decimal and both sides of the decimal point are handled individually using the sequence operator along with the numeric compression options that Xmill supply. For example, the compression option chosen by XCAT for 432.34 is shown in Figure 4.

Float Point number

432.34

seq (u “.” u8)
432 . 34

Figure 4

Since XCAT tests these fields individually, the best compression option for each field is selected by its effectiveness and is saved for later use. After all fields within the file have been empirically tested, XCAT will have the best compression options and then will combine them together to run the combined option set. XCAT records the results for the next step.

2.3 Group Options

After all files have gone through XCAT's empirical testing process, XCAT will have a Xmill option set that is individually tailored to best compressing each individual file. XCAT will examine the common fields within the file group and empirically test the different saved compression options on the entire group of files. In some cases the option could be the same throughout the file group, so there would be no need for further empirical analysis. In most cases there will be a small variant of different options that will be tested among the file group. XCAT will take the compression option that performs the best on the file group and save it. After all combinations of compression options have been tested on the file group XCAT will have an Xmill compression option set that will work for the file type. The compression options are stored in a file and can be used without further use of XCAT when compressing any file of that particular type.

2.4 Output

After the Xmill analysis has been processed, XCAT will look at XMLPPM's and XCAT's Xmill user defined options results and output them to the user. Generally in cases where the xml file contains more text data than numeric XMLPPM will out perform Xmill with XCAT's compression options. In other cases Xmill may outperform XMLPPM by compressing numerical data more efficiently. The program will display the results and recommend the best method of compression to the user. The user may take this knowledge and use is as a method of compression in the future, on files of that structure.

3. Experimental Results

As stated earlier, our experiments entailed performing XML compression using Xmill and XMLPPM. The results presented and discussed herein, are from using XML Compression Advisory Tool, XCAT. We developed this tool in order to facilitate and optimization of the underlying tool set, based on the identified data structures of the data files.

In this section, we present an extensive experimental evaluation of XCAT using the following aviation related data sets:

- Radar Track Data
- Digital Automatic Terminal Information Service (D-ATIS) data
- Runway Visual Range (RVR) data

Table 1 lists the results of different compression methods, on a given set of Radar XML files.

File Number	XML Size (Kb)	Compressed Size (Kb)				Compression Ratio			
	XML	Zip	Xmill	XMLPPM	Xmill With XCAT Option(s)	Zip	Xmill	XMLPPM	Xmill With XCAT Option(s)
1	72,317	3,631	1,933.9	1,475.3	1,421.3	19.92	37.40	49.02	50.89
2	81,237	4,056	2,036.7	1,632.8	1,561.3	20.03	39.89	49.76	52.04
3	83,002	4,183	2,085.3	1,662.8	1,618.2	19.85	39.81	49.92	51.30
4	84,976	4,357	2,218.7	1,745.9	1,745.0	19.51	38.30	48.68	48.70
5	85,215	4,326	2,166.1	1,711.9	1,698.8	19.70	39.35	49.78	50.17
6	86,515	4,380	2,193.8	1,737.3	1,715.6	19.76	39.44	49.80	50.43
7	87,961	4,443	2,216.7	1,770.9	1,718.0	19.80	39.69	49.68	51.20
8	93,887	4,744	2,383.0	1,890.2	1,850.1	19.80	39.40	49.68	50.75
9	96,201	4,902	2,457.9	1,937.6	1,930.2	19.63	39.14	49.65	49.84
10	98,925	5,021	2,584.4	2,066.2	2,003.0	19.71	38.28	47.88	49.39
Average	**87,024**	**4,404**	**2,227.7**	**1,763.1**	**1,726.2**	**19.77**	**39.07**	**49.38**	**50.09**

Table 1: Compression Performance of Radar Data using Single File Option

File Number	XML Size (Kb)	Compressed Size (Kb)				Compression Ratio			
	XML	Zip	Xmill	XMLPPM	Xmill With XCAT Option(s)	Zip	Xmill	XMLPPM	Xmill With XCAT Option(s)
* 1,2,3,4	321,532	27,816	8,186.4	6,516.6	6,346.8	11.56	39.28	49.35	50.67
5	85,215	4,326	2,166.1	1,711.9	1,711.8	19.70	39.35	49.78	49.78
6	86,515	4,380	2,193.8	1,737.3	1,737.2	19.76	39.44	49.80	49.81
7	87,961	4,443	2,216.7	1,770.9	1,770.8	19.80	39.69	49.68	49.68
8	93,887	4,744	2,383.0	1,890.2	1,890.1	19.80	39.40	49.68	49.68
9	96,201	4,902	2,457.9	1,937.6	1,937.5	19.63	39.14	49.65	49.66
10	98,925	5,021	2,584.4	2,066.2	2,003.0	19.71	38.28	47.88	49.39
Average	**87,024**	**4,404**	**2,227.7**	**1,763.1**	**1,739.7**	**19.77**	**39.07**	**49.38**	**50.06**

* Analyzed as a group using XCAT to produce semantic options that we used on files 5 to 10

Table 2: Compression Performance of Radar Data using Group File Option

The Radar data was obtained from FAA [3], each file consists of one day's worth of recording of radar data for SFO. The radar tracks contain time, geographical location (longitude, latitude and altitude), velocity, climb-rate, and other flight related information. Radar Data files are converted to XML format from database table format. In Table 1, we observe that the best compression is achieved for this data by use of Xmill with XCAT having optimized the semantic options for the tool. In Table 2, we observe that XCAT's Xmill group options perform similar to the single file options.

Table 3 lists the results of different compression methods, on a given set of ATIS XML files.

	XML Size (Kb)	**Compressed Size (Kb)**				**Compression Ratio**			
File Number	XML	Zip	Xmill	XMLPPM	Xmill With XCAT Option(s)	Zip	Xmill	XMLPPM	Xmill With XCAT Option(s)
1	451	32	22.1	13.9	21.0	14.10	20.41	32.45	21.48
2	471	31	22.3	13.7	21.2	15.20	21.13	34.38	22.22
3	421	28	20.4	12.7	19.3	15.04	20.64	33.15	21.82
4	431	30	21.3	13.3	20.1	14.37	20.24	32.41	21.45
5	500	34	24.2	15.3	23.2	14.71	20.67	32.68	21.56
6	448	30	21.9	13.7	21.0	14.94	20.46	32.71	21.34
7	336	22	16.2	10.5	15.5	15.28	20.75	32.00	21.68
8	415	28	20.4	12.4	19.4	14.83	20.35	33.47	21.40
9	421	27	19.8	12.3	18.6	15.60	21.27	34.23	22.64
10	466	32	22.2	13.9	21.2	14.57	21.00	33.53	21.99
Average	**436**	**29.4**	**21.1**	**13.2**	**20.1**	**14.87**	**20.7**	**33.11**	**21.76**

Table 3: Compression Performance of ATIS Data using Single File Option

The ATIS data was obtained from Skysource [1], for each airport, the data was downloaded from skysource website as an html page. Each ATIS data file consists of one day worth of recording of ATIS data for 85 major airports in USA, for an approximate one-hour frequency. Each record in the data file consists of weather information and summary of runway activities for an airport. These files are then parsed and converted to XML format using special parser designed by our team. In Table 3, we observe that the best compression is achieved for this data by use of XMLPPM.
Table 4 lists the results of different compression methods, on a given set of RVR XML files.

File Number	XML Size (Kb)	Compressed Size (Kb)				Compression Ratio			
	XML	Zip	Xmill	XMLPPM	Xmill With XCAT Option(s)	Zip	Xmill	XMLPPM	Xmill With XCAT Option(s)
1	28,113	754	175.4	271.2	167.7	37.29	160.28	103.67	167.64
2	29,032	768	178.1	277.1	170.6	37.81	163.01	104.78	170.18
3	32,474	819	180.8	297.8	172.0	39.66	179.62	109.05	188.81
4	33,368	830	181.8	301.7	172.8	40.21	183.55	110.60	193.11
5	26,758	694	159.3	254.1	151.0	38.56	167.98	105.31	177.21
6	25,670	658	148.4	240.9	141.8	39.02	172.98	106.56	181.03
7	23,254	601	132.4	219.7	125.7	38.70	175.64	105.85	185.00
8	29,748	733	147.3	270.3	141.1	40.59	201.96	110.06	210.83
9	30,880	776	167.8	283.1	160.4	39.80	184.03	109.08	192.52
10	31,679	805	177.3	290.9	169.4	39.36	178.68	108.90	187.01
Average	**29,098**	**744**	**165.0**	**271.0**	**157.0**	**39.10**	**176.77**	**107.38**	**185.33**

Table 4: Compression Performance of RVR Data using Single File Option

The RVR Data is obtained from FAA [5]. For each airport, the data was downloaded from an FAA website as an html page. Each RVR data file consists of one day worth of recording of RVR data for 48 major airports in USA. Each record in the RVR data file consists of runway visibility range and lighting information for all major runways for an airport. These HTML files are parsed and converted to XML format. In Table 4, we observe that the best compression is achieved for RVR data by use of Xmill with XCAT having optimized the semantic options for the tool.

From the results it is evident that on RVR data sets, Xmill without options performs better then XMLPPM. The semantic options evaluated by XCAT improve on the performance of Xmill. On Radar data sets, XMLPPM performs better then default Xmill. However, Xmill with XCAT suggested semantic options perform even better than XMLPPM. And on ATIS data sets, XMLPPM performs much better than Xmill. As XCAT optimization of Xmill options is for numerical fields, we were unable to use

XCAT to find Xmill options for text fields. One of the future directions is to extend XCAT to include text options.

After examining Xmill's options recommended by XCAT for single files from the same group, we found that often these options vary from file to file. For example, semantic option "Delta Compressor" improves compression on a field in one XML file where differences between consecutive numbers are usually substantially smaller then the numbers itself, but may not improve compression in another file with same DTD where that is not true. Therefore, if purpose of analysis is not to determine the effectiveness of semantic options for single file but for entire group of files with the same DTD, then effectiveness of the semantic option must be examined on the group or at least on more than one file.

4. Conclusion and Future Studies

This paper has described XCAT's capability to recommend the best XML Compression tool between Xmill and XMLPPM, and if the selected tool is Xmill then produce semantic options to the achieve optimal compression ratio. Whilst comparing semantic options recommended by XCAT for single file and multiple files with same DTD the single file's options performs best on that file only, although the group file options performs similarly but has the advantage of being able to be run on files of the same DTD without further use of XCAT. In summary, our study has shown that XCAT can be used to infer the best compression methods for files belonging to a particular DTD permitting fast compression of XML files with compression ratios higher then those normally achieved. However, XCAT can also be used to infer even better compression methods if it is asked to find best compression options for a single file offline.

For Xmill, comparing semantic options produced by manual analysis and XCAT; we observed that XCAT produces more efficient semantic options then manual analysis, which helps achieve further compression of the file. For a large XML file, >10Mb, with 80% of the file composed of numerical data; a manual process of optimization on semantic option would take several days. Thus processing large numbers of such files would in itself be tediously lengthy. XCAT overcomes this shortfall by delivering a reliable and fast mechanism, for automatically processing such data files, in order to achieve a high compression ratio. Thus, XCAT eliminates the need of time-consuming manual experimentation and at the same time improves the compression ratio.

Future work involves improving XCAT's capability to analyze fields with the string data type, and also to find patterns to produce more semantic options for Xmill.

5. References

[1] SkySource Login. (n.d.). Retrieved June 23, 2003, http://www.skysource.net.

[2] Liefke, H., Suciu D., "Xmill: an efficient compressor for XML data",

Proceedings of SIGMOD Conference, 2000.

[3] Federal Aviation Administration site. (n.d.). Retrieved June 23, 2003, http://www2.faa.gov.

[4] Lempel, A., Ziv J., "A Universal Algorithm for Sequential Data Compression", IEEE Transaction on Information Theory 23, 3 (May): 337-343, 1977.

[5] Runway Visual Range (RVR). (n.d.). Retrieved June 23, 2003, from http://rvr.fly.faa.gov.

[6] XML specification, (n.d.). Retrieved June 23, 2003 from http://www.w3.org/XML/.

[7] Cheney, J. Compressing XML with Multiplexed Hierarchical PPM Models. (n.d.). Retrieved June 23, 2003 from http://www.cs.cornell.edu/People/jcheney/papers/cheney-dcc2001.pdf.

[8] Xmill User Manual site. (n.d.). Retrieved June 23, 2003, http://www.research.att.com/sw/tools/xmill/MANUAL.txt.