Paper 39-26

A Case Study in the Use of XML™ to Integrate SAS® with Third-Party Applications

Michael C. Palmer, Zurich Biostatistics, Inc., Morristown, NJ
Cecilia A. Hale, Zurich Biostatistics, Inc., Morristown, NJ

ABSTRACT

SAS sometimes provides an optimal environment for part of an assignment but third-party applications provide a better environment for other parts. The creation of report-quality statistical tables is such a situation. SAS is optimal for statistics but clumsy for publishing report-quality tables. XML offers a new way to integrate SAS with publishing software. To implement, SAS was taught a standard XML vocabulary for tables. This means adding three capabilities to SAS: importing the vocabulary, exporting the vocabulary, and manipulating the XML vocabulary as XML. SAS has an experimental capability to import and export some XML vocabularies but no capability to manipulate XML as XML. The work reported used no experimental features. SAS programs were written to provide a general capability for the import, export, and manipulation of the XML vocabulary used. With these programs, SAS and publishing software converse about table content and style. SAS does statistics and the publishing software creates report-quality tables taking advantage of the full feature set of the publishing software. This conversation produces report-quality tables in an automated hands-off/lights out process. The technique of teaching SAS how to import, export, and manipulate specific XML vocabularies so that it can carry on a dialogue with a third-party application will be useful where SAS is an optimal solution to part of a problem but the third-party application would be a better solution for other parts.

INTRODUCTION

Statisticians often think of their work as having two substantial components. The first is statistical analysis per se, that is, the design and analysis of experiments and surveys and the generation of statistical results from the data. The second is the presentation of those results. SAS has proven to be an excellent environment for the analysis component of statistical work but a clumsy environment for the presentation component.

LEGACY PRACTICES

Results presentation typically involves programming with DATA steps and reporting PROCs such as REPORT and TABULATE. The addition of ODS to SAS has added reporting flexibility but the creation, revision, and reuse of reports in SAS is still very much a programming-intensive activity. Typically, in a final step or steps, SAS output is delivered to publishing software like Microsoft® Word for final formatting and incorporation into larger documents.

The flow of information in this process is one-way, from SAS out to the publishing software. SAS tells the publishing software what content the statistical table will have and how the table will be formatted on the page. No information about table content or style can go back to SAS in an automated way. As a consequence, the formatting and styling capabilities of the publishing software are not available for use, unless SAS happens to support them. In addition, this process often has manual setup and programming steps.

NEW TECHNOLOGY, NEW OPPORTUNITIES

Recently, new software technology has become available that makes it possible to use SAS for statistical analysis only, not presentation, and to use specialized publishing software for presentation of statistical results. The new technology makes it possible for SAS and third-party publishing software to have a very detailed, two-way conversation about statistical table content and style. The new technology is XML, eXtensible Markup Language.

XML is a standard for computerized documents and data. XML was released for use in February 1998 by the World Wide Web Consortium (W3C®), a standards organization for the World Wide Web. XML is a non-proprietary, platform-independent meta-language, that is, a standard to use in developing vocabularies for specific applications.

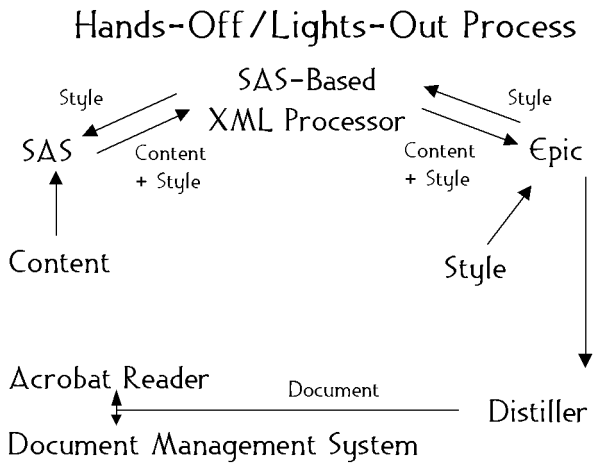
The OASIS table model is an XML vocabulary specifically designed to describe publication-quality tables. Like XML, the OASIS table model is non-proprietary, platform independent, and developed by a non-profit consortium, OASIS.

The availability of XML and the OASIS table model offer a new way to integrate SAS with a third-party, commercial, off-the-shelf publishing application. The resulting integrated system offers a richer set of features than each application by itself. The techniques of that XML-based integration may be useful in integrating SAS with other third-party applications, thereby broadening its usefulness.

IMPLEMENTATION

To implement the integration, SAS had to be taught XML and the OASIS table model. The publishing software that was used, Epic by Arbortext, Inc., out of the box understands XML and the OASIS table model. Teaching SAS XML and the table model means adding three capabilities to SAS: the capability to import the table model as XML, the capability to export the table model as XML, and the capability to manipulate the XML table model. In version 8.0, SAS has an experimental capability to import and export some XML vocabularies but no capability to manipulate XML as XML.

FIGURE 1. APPLICATION INTEGRATION VIA XML



For the work reported here, it was decided to use stable, fully-supported features of SAS, not experimental features. SAS programs were written to provide a general capability for the import, export, and manipulation of OASIS tables in XML. With these programs, SAS and Epic can converse about table content and style. SAS does what it does best, statistics, and the publishing software does what it does best, create report-quality tables taking advantage of the full feature set of the publishing software. This conversation can produce report-quality tables in an automated hands-off/lights out process.

Using XML as an intermediate format, a variety of final formats can be produced including PDF, Postscript, RTF, and the venerable paper.

FOUR COMPONENTS

The underlying technology for integrating SAS with third-party publishing software involves teaching SAS how to import, export,

and manipulate XML. The implementation of the automated process elaborates on this underlying technology. The implementation has four components, a table content database, expandable table shells, database maps, and publishable XML..

TABLE CONTENT

The table content database holds absolutely all of the table content, both numbers and text, in SAS datasets. No content at all, including table titles and footnotes, is embedded in table programs.

FIGURE 1. TABLE CONTENT DATABASE

Data set 1

Key1	Key2	…	Content	Admin1	Admin2	…

Data set 2

Key1	Key2	…	Content	Admin1	Admin2	…

Data set 3

Key1	Key2	…	Content	Admin1	Admin2	…

Data set 4

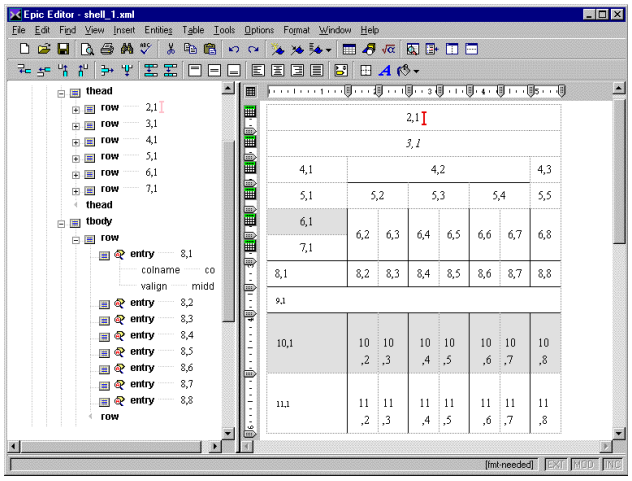
Key1	Key2	…	Content	Admin1	Admin2	…

The table content database may have several datasets, but they all have a simple, identical, extensible structure: keys, content, administrative variables. The number of key variables and administrative variables may vary from dataset to dataset, but each record in every dataset has one and only content variable. The structure facilitates the writing of general purpose SAS programs for the automated system.

TABLE STYLE

The expandable table shell is an XML document that is created in Epic's visual interface, similar to the way a table is created in MS Word. Epic, unlike Word, automatically saves the table shell as XML.

FIGURE 3. EXPANDABLE TABLE SHELL



The table shell shows how a final table will look but has none of the content of the final table. It is naked style. The shell has the number of columns of the final table, the borders, shading, fonts, horizontal justifications, vertical justifications, etc. but no content. Compositionally, the shell is essentially the table header and the smallest repeatable row structure of the final table.

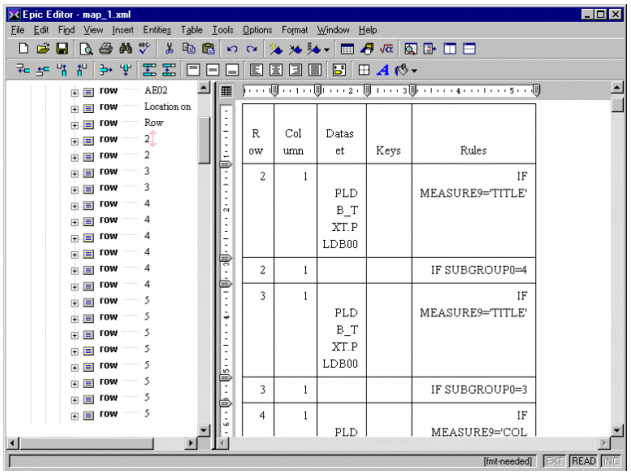
In our implementation, SAS has been taught how to match the row structure in the shell to the data and to expand the shell into a final table preserving the style of the shell.

A shell is reuseable without modification wherever its style is needed. It can be used and reused multiple times for a given table content database or in several table content databases. It is not modified for any of these multiple reuses.

COMBINING CONTENT AND STYLE

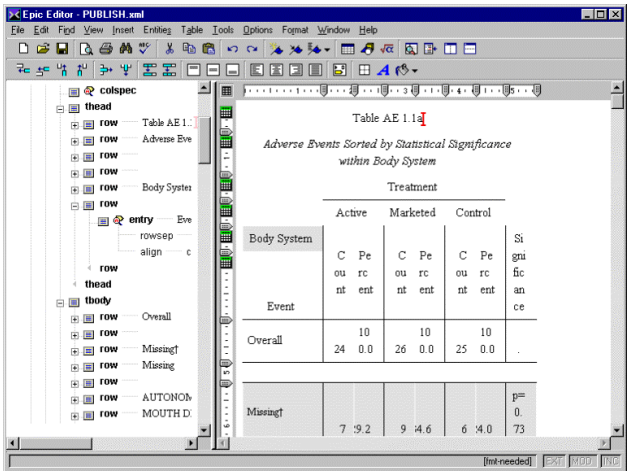
A database map is an XML document that lists each cell in a given expandable table shell and states where in the table content database data for that cell are. The map is built in Epic's visual interface and processed in SAS that has been taught the import, export, and manipulation of XML.

FIGURE 4. DATABASE MAP



In our implementation, these two XML documents, the map and a shell, define a statistical table, both its content and its appearance. This is in contrast to the traditional way of creating reports in SAS which is to program them. Documents are generally easier to create, revise, and reuse than programs and often easier to validate and archive as well.

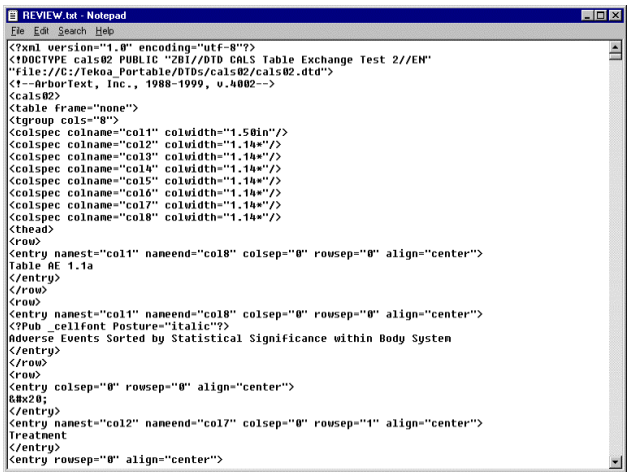
FIGURE 5. PUBLISHABLE XML IN AUTHORING TOOL



The fourth and final component of the implementation is publishable XML. It is created in SAS that, again, has been taught how to import, export, and manipulate XML. SAS programs read and interpret the content information in the database map XML document and fetch the data that are specified. The programs read and interpret the style information in the XML document that is the expandable table shell. The programs then expand the table shell to match the fetched data, preserving the style information in the shell. The expanded,

populated document is the publishable XML. It is in turn imported by Epic for publishing to PDF, Postscript, RTF, or paper.

FIGURE 6. PUBLISHABLE XML AS XML



SUMMARY

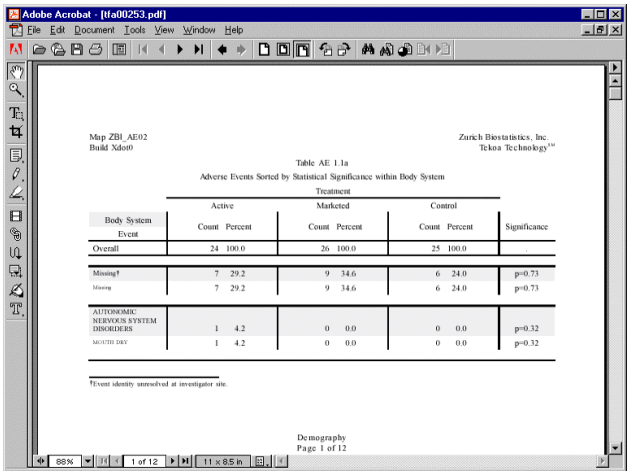
To summarize the implementation, two XML documents, the expandable table shell and database map, replace the traditional SAS table programs. Publishable XML replaces traditional SAS output. Unlike ODS, the publishable XML can take advantage of the full feature set of commercial, off-the-shelf publishing software and publish to any formats supported by the publishing software.

The implementation organizes the table content database, database maps, expandable table shells, and publishable XML into directories and processes them with a stable suite of general purpose SAS programs and Epic publishing software. During the processing, SAS and Epic have a continuing, two-way conversation about table content and style. This conversation replaces the traditional manual setup and programming steps needed to create report-quality documents that include output from SAS.

CONCLUSION

The availability and popularity of XML offers SAS developers new opportunities to have SAS work intimately with third-party applications. The technique of teaching SAS how to import, export, and manipulate specific XML vocabularies so that it can carry on a dialogue with a third-party application will be useful where SAS is an optimal solution to part of a problem but the third-party application would be a better solution for other parts.

FIGURE 7. PUBLISHED DOCUMENT AS PDF



AN XML FAQ

1. *What is XML?* XML is a non-proprietary, platform-independent meta-language for structuring information. XML vocabularies use the meta-language to define elements and the rules on how to use them. An XML instance is a text file of data formatted according to a specific vocabulary.

2. *What does XML stand for?* Extensible markup language. Extensible in the sense that anyone with an understanding of the rules of XML can make up their own XML vocabulary. A markup language because it provides the means to describe documents of data.

3. *What's it good for?* The contents of XML documents can go into a database with relative ease or the content of a database can go into XML documents.

4. *Where does XML come from?* The World Wide Web Consortium (W3C) guided the creation of XML in a joint effort of many industry and academic contributors. The XML standard was released in February 1998. The W3C is a non-profit consortium made up primarily of corporate and other organizations with an interest in the development of the World Wide Web.

5. *How do I buy XML?* XML is a standard that anyone can use for free, subject to licensing restrictions. Many vendors supply applications that use XML to do a variety of things. One place to start looking is www.xml.com.

6. *Does SAS do XML?* SAS version 8.0 includes an experimental XML driver as part of the Output Delivery System (ODS). It gives SAS the capability to export documents in limited XML vocabularies. An XML vocabulary for SAS datasets is also under development.

7. *XML, HTML, SGML, what's next?* Officially, XML is the successor to HTML, the language underlying the World Wide Web, although the transition from HTML to XML is years off. Both XML and HTML are derived from SGML, the grandmother of markup languages.

REFERENCES

1. Bosak, J. and T. Bray "XML and the Second-Generation Web" 1999
www.scientificamerican.com/1999/0599issue/0599bosak.html
2. World Wide Web Consortium (W3C) "Architecture Domain: Extensible Markup Language (XML)" 2000 www.w3.org/XML
3. SAS Institute, Inc. "Base SAS Software: ODS XML Primer" 2000
www.sas.com/rnd/base/topics/odsxml/index.htm
4. Zurich Biostatistics, Inc, "Papers and Presentations" 2000
www.zbi.net/NewFiles/Presentations.html

ACKNOWLEDGMENTS

.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Michael Palmer or Cecilia Hale
Zurich Biostatistics, Inc.
45 Park Place South
PMB 178
Morristown, NJ 07960
Phone: 973-727-0025
Email: mcpalmer@zbi.net or cahale@zbi.net
Web: www.zbi.net