Microsc. Microanal. 8 (Suppl. 2), 2002
DOI 10.1017.S1431927602105782

XML for Microanalysis: EMSA/MAS Spectrum File Format 2.0?

John Henry J. Scott, Scott A. Wight, Barbara B. Thorne

Surface and Microanalysis Science Division, National Institute of Standards and Technology, Gaithersburg, MD 20899

In October 1991, an EMSA/MAS Task Force implemented the first spectral data exchange file format designated as an EMSA/MAS standard. This format, EMMFF v. 1.0, was designed to allow the free exchange of microanalytical spectra between laboratories [1].

We propose a new spectrum file format based on Extensible Markup Language (XML), a flexible, open standard for creating "self-describing data" that currently enjoys commitment and support from many databases, authoring tools (e.g. MS Office 2000), and the web community (including browsers such as Netscape and Internet Explorer). Despite its name, XML is not itself a markup language; it is a standard that describes how to build your own markup language uniquely suited to a specific application domain such as microbeam analysis. Like the existing EMMFF v 1.0 format, Microanalysis Markup Language (MML) — based on XML — has a clear, simple syntax with an intuitive structure. Files are human-readable and can be edited or created from scratch with an ASCII editor. Figure 1 shows an excerpt from the header of the EMMFF v 1.0 file format (left) compared with the equivalent data encoded in MML (right).

In addition to retaining most of the advantages of the EMMFF format, MML introduces many new desirable features. Because it uses Unicode as its standard character set, MML supports international scripts including relatively simple Latin, Greek, and Cyrillic characters as well as tens of thousands of Korean hangul, Japanese kana, and kanji and Han ideographs. For the majority of users without Unicode-aware editors, the UTF-8 encoding reassuringly provides backward compatibility with ASCII for transparent use with legacy systems.

The number of XML-compliant markup languages has exploded in the last few years, with hundreds now in use in the banking and financial sectors, among web developers, information technologists, publishers, database vendors, and the government and military [2]. This popularity and large user base has led to the development of high-quality tools and code libraries for reading, writing, validating, manipulating, transforming, and visualizing XML documents. In the guise of Netscape or Internet Explorer, most microanalysts already have at least one professional-quality application capable of parsing and validating well-formed MML documents installed on their desktop. Furthermore, because all XML-based markup shares a common structure, MML is inherently designed to interoperate with emerging scientific standards such as Chemical Markup Language (CML), Mathematical Markup Language (MathML), Material Properties Markup Language (MatML), and Spectroscopy Markup Language (SpectroML, a nascent markup currently limited to UV-VIS spectra). Support for XML in major relational database management systems ensures easy integration of MML into Laboratory Information Management Systems (LIMS), while the related technologies of XML Stylesheet Language (XSL), XSL Transformation Language (XSLT), and XSL Formatting Objects (XSL-FO) allow for translation, formatting, and repurposing of MML documents in modern, object-oriented and networked research environments.

https://doi.org/10.1017/S1431927602105782 Published online by Cambridge University Press

Unlike the current standard file format, MML supports multiple spectra in each file. While the primary intent of this feature is to store multiple analyses from a single sample or experiment (50 points in a homogeneity study, for example), it is also possible to store multispectral datasets containing energy-filtered TEM micrographs, or even hyperspectral datasets from spectrum imaging systems. While the MML file sizes will be quite large for the last two cases mentioned, the format is still quite practical for the transmission of data over networks or the exchange of files using removable data storage media. Once the data has reached its final destination, conversion to proprietary binary formats would enable much more efficient storage.

Because acceptance and adoption by the microanalysis community is necessary for the success of the standard, MML has been designed with the needs of all concerned parties in mind, including end users, equipment vendors, and software developers. This first version of MML is presented as a strawman and is intended to change fluidly as it is used, taking full advantage of the extensible nature of XML. Participatory input and suggestions for improvement are welcomefrom all.

References

[1] ftp://ftp.msa.microscopy.com/pub/4-MMSLib/XEDS/EMMFF/; ftp://www.amc.anl.gov/AMC-3/ANLSoftwareLibrary/2-EMMPDL/Xeds/EMMFF.
[2] many languages are registered with the XML repository at www.xml.org.

```
#FORMAT      : EMSA/MAS Spectral Data File
#VERSION     : 1.0
#TITLE       : NIO EELS OK SHELL
#DATE        : 01-OCT-1991
#TIME        : 12:00
#OWNER       : EMSA/MAS TASK FORCE
#NPOINTS     : 20.
#NCOLUMNS    : 1.
#XUNITS      : Energy Loss (eV)
#YUNITS      : Intensity
#DATATYPE    : XY
#XPERCHAN    : 3.1
#OFFSET      : 520.13
#CHOFFSET    : -168
#SIGNALTYPE  : ELS
#XLABEL      :  Energy
#YLABEL      :  Counts
#BEAMKV   -kV: 120.0
#EMISSION -uA: 5.5
#PROBECUR -nA: 12.345
```

```
<?xml version="1.0"?>
<spectrum xunits="Energy Loss (eV)"
yunits="Intensity" signaltype="ELS">
<title>NIO EELS OK SHELL</title>
<date>1991-10-01</date>
<time>12:00</time>
<owner>EMSA/MAS TASK FORCE</owner>
<xperchan>3.1</xperchan>
<offset>520.13</offset>
<choffset>-168</choffset>
<beamkv>120.0</beamkv>
<emission-uA>5.5</emission-uA>
<probecurrent-nA>12.345</probecurrent-nA>
```

Figure 1. Example file excerpts from the headers of the existing EMSA/MAS Spectrum File Format 1.0 (left), and Microscopy Markup Language (MML), a proposed XML-based enhanced format (right). Both formats are designed to be human-readable/editable and intuitive instead of efficient and parsimonious. The XML-based format has the added advantages of being extensible, web-aware, more easily validated, and more tolerant of vendor customization. It is also backed by a large, professionally-developed codebase, a suite of tools, and plentiful documentation on XML itself.

https://doi.org/10.1017/S1431927602105782 Published online by Cambridge University Press