NISTIR 6821

SpectroML
An Extensible Markup Language for the Interchange of Molecular Spectrometry Data

M. Alexander Rühl
Reinhold Schäfer
Gary W. Kramer

NIST
National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

NISTIR 6821

SpectroML
An Extensible Markup Language for the Interchange of Molecular Spectrometry Data

M. Alexander Rühl
Gary W. Kramer
Analytical Chemistry Division
Chemical Science and Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8394

Reinhold Schäfer
Wiesbaden Computer Integrated Laboratory
Computer Science Department
University of Applied Sciences
65197 Wiesbaden / Germany

April 2002

U.S. Department of Commerce
Donald L. Evans, Secretary

National Institute of Standards and Technology
Arden L. Bement, Director

Disclaimer:
This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any of its employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, product, or process disclosed, or represents that its use would not infringe upon privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government and shall not be used for advertising or product endorsement purposes.

Acknowledgments:
This project is funded [in part] by NIST's Systems Integration for Manufacturing Applications (SIMA) Program. Initiated in 1994 under the federal government's High Performance Computing and Communications effort, SIMA is addressing manufacturing systems integration problems through applications of information technologies and development of standards-based solutions. With technical activities in all of NIST's laboratories covering a broad spectrum of engineering and manufacturing domains, SIMA is making information interpretable among systems and people within and across networked enterprises.

We would like to acknowledge XML.org for hosting the SpectroML files on their web site. We thank J.C. Travis and P.C. DeRose for helpful discussions during the development of SpectroML and for reviewing this manuscript.

Availability:
Paper copies of this document are available from the National Institute of Standards and Technology. This document is available electronically on the Internet from the Molecular Spectrometry and Microfluidic Methods Group homepage (http://www.cstl.nist.gov/nist839/839.04/index.html) as an Adobe Portable Document Format (PDF).

Corrections:
Error reports, corrections and comments concerning this document are welcomed and should be directed to Gary W. Kramer, 100 Bureau Drive, Stop 8394, NIST, Gaithersburg, MD 20899-8394; email: gary.kramer@nist.gov; telephone: (301) 975-4132; telefax: (301) 977-0587.

Keywords:
data interchange, markup language, molecular spectrometry data, SpectroML, uv/vis, XML, spectral data interchange, chemical data

Table of Contents

Introduction

Ever since spectrometric instruments have been coupled with computers, there has been a need to interchange the data produced by the instrument so that they can be used in other applications.

Each instrument's software usually has its own native data format, which is often incompatible with others. Because of this, a number of data interchange formats have been developed over the years so that different applications can share data, as long as each application supports the import and export functions of the interchange format.

This approach is functional, but it has several disadvantages:

- The current interchange formats are fixed, which means it is not possible to add new data elements easily.
- The interchange data structure is fixed, which forces the data elements to be maintained in a precise order.
- The various instrument software and interchange formats do not all convey the same information, so data can be lost in the interconversion.
- Some of the interchange formats encompass a wide variety of instrument data resulting in a huge number of data elements, many of which are not needed for a given application.
- Many applications do not support all formats.
- Result metadata (descriptive elements concerning the data) and information about the sample and the measurement process are often omitted.
- Current interchange mechanisms are not compatible with modern computer network technologies.

Beyond these difficulties, interchange developers, instrument manufacturers, and end users have often worked against each other in developing consistent standards for data interchange. Consequently, today there is no single standard way to exchange or visualize scientific instrument data.

Use of an extensible markup language for data interchange can solve most of these difficulties. The concept of an XML is to enclose data elements between tags that identify the data element by name and attributes. The most famous of the XML languages is HTML (hypertext markup language), the *lingua franca* of the Internet. Together with its type definition, a marked-up document can be easily interchanged, processed, stored, and visualized by numerous applications—many of them already developed for Internet use. XML documents are thus free of ties to specific systems or manufacturers.

To demonstrate the utility of an XML approach for instrument data interchange, *SpectroML* was created - a markup language for molecular spectroscopy data. At present *SpectroML* is being developed solely for UV/Vis data to keep the scope of the project manageable. This document describes this markup language and its environment, shows its structure and elements, and gives examples and an outlook on applications.

UV/Vis Formats and the Way to *SpectroML*

To develop the structure and an initial vocabulary for *SpectroML*, three existing data interchange formats (GRAMS [*1*], JCAMP-DX [*2*], and ANDI [*3*]) were compared, relevant ASTM definitions [4] were consulted, and all items related to UV/Vis spectroscopy were extracted. [*Appendix A documents this effort.*]

The structure developed while analyzing these formats provided a good base for the XML vocabulary. Each of the existing formats views the data differently and emphasizes different elements. Selecting those elements that suit most applications in UV/Vis spectroscopy provides a good initial vocabulary. Some of the elements can be sub-divided to avoid having more than one piece of information per element. By using XML, the vocabulary can be extended fairly easily, so that one can have, for example, an extended *SpectroML* containing elements for his/her own usage together with the core that is general for all applications. [*See Appendix B for a short introduction to XML.*]

With the initial vocabulary in hand, a DTD (document type definition) can be developed. This allows checking an XML file to determine its correctness. An XML schema can be developed to provide a more powerful way of validating XML documents.

There are different possibilities for visualizing the data. For example, using XSL (Extended Stylesheet Language) stylesheets, the information of the file can be displayed in various ways, and users can easily adapt this to their needs.

Moreover, applications or plug-ins for applications can be developed to use *SpectroML* in multiple ways. The Java platform is attractive for developing these, because it provides platform independence.

A well-structured XML file provides a flexible, powerful, complete, and platform-independent way to store UV/Vis data and exchange them over the Internet.

Figure 1 - The *SpectroML* logo

Structure of *SpectroML*

Based on our analysis of existing data interchange formats for molecular spectrometry data, we developed an initial vocabulary and organized it to develop a logical and regular structure, and extended it to provide a linking mechanism. Like all XML documents, the structure is arranged hierarchically like a tree, starting from a root and with increasingly detailed sub-elements ending in the leaves as shown in Figure 2:

- The root element (the ground in Figure 2) contains one or more experiments. The individual experiments are implicitly related by being grouped into one document; however, they can be explicitly related via linking references.
- Each experiment (the tree trunk in Figure 2) contains five groups. The file group is a header group that describes all the datasets within an experiment. Each of the four groups—instrument, sample, measurement, and data—describes a different aspect of the dataset and contains the data values themselves.
- Each of these four groups (a main branch in Figure 2) contains two different blocks. Generally speaking, the blocks divide the group data into a fixed part and a variable part. Each of these two blocks can appear several times. Its ID (identification string) affords the possibility of reusing one block for different datasets within an experiment. For example, one instrument can be used with several samples, without repeating it for each dataset.
- Each block (a smaller branch in Figure 2), except for the core data, contains sections (a smaller branch in the figure). A section divides a block into different sub-aspects. In this specification, each of these blocks has two sections; however, this is not mandatory and can be expanded in future versions.
- Each section (a twig in Figure 2) contains data elements to hold the data and metadata.
- Each element (a leaf in Figure 2) may contain sub-elements. This allows storing structured data in an element. Each element can also have an attribute, such as a format description for the data contained.

The spectroscopy method is an attribute of an experiment, which means several methods can be combined within one *SpectroML* file. The current elements focus on UV/Vis, but the required metadata for other methods can be added in future since the structure that holds the data values was designed for a broad range of data structures.

It is important to realize that even though XML files are human-readable, they are created to be processed by a computer. The hierarchy, its structure, its depth, and complexity are designed to make the XML file "parsable" by a computer and foster flexibility and extensibility.

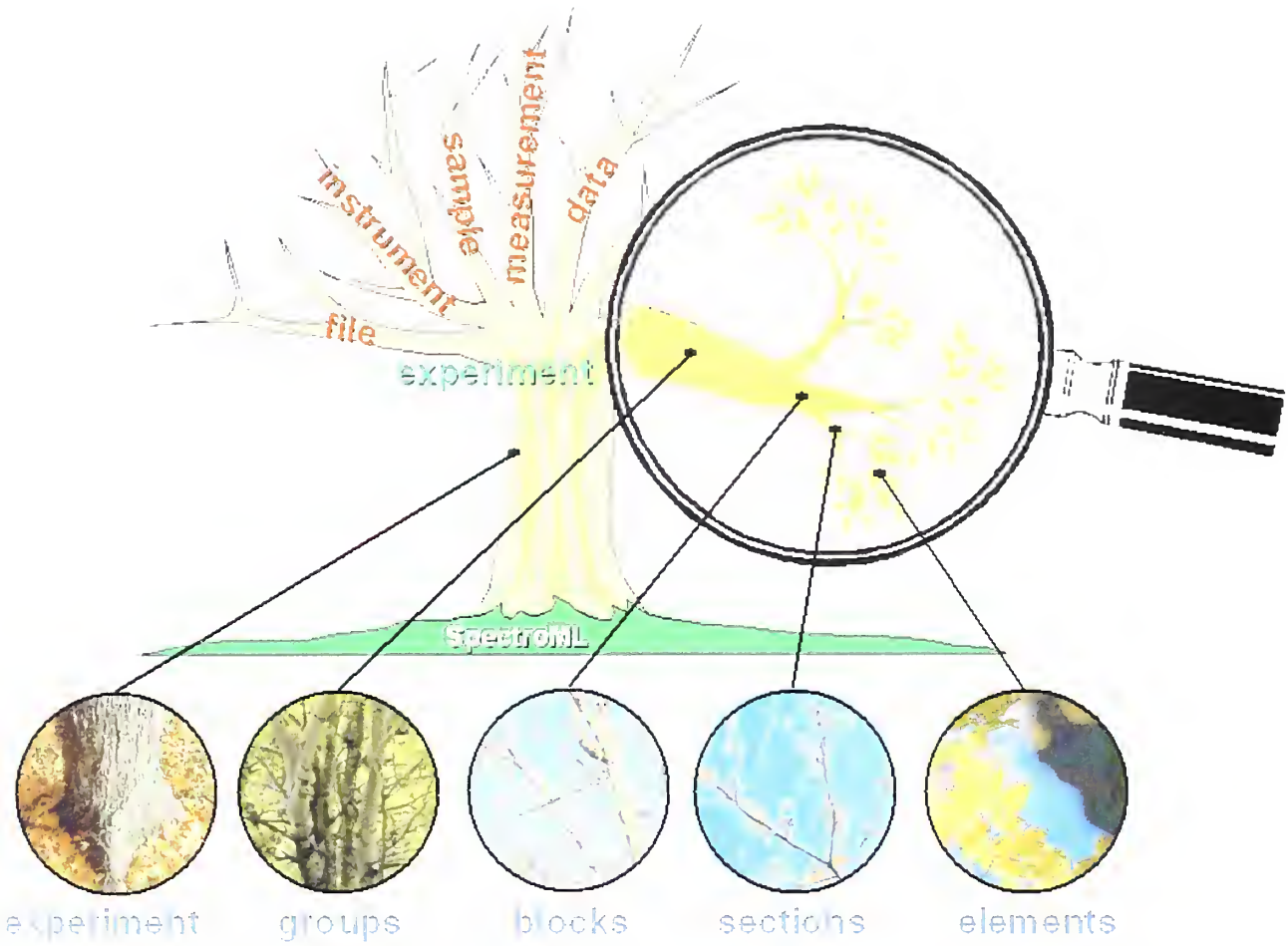

Figure 2 – The hierarchical structure of SpectroML

Illustration of the Structure of SpectroML

- *SpectroML* (the ground in Figure 2) supports the whole structure.
- An *experiment* (the tree trunk in Figure 2) holds information about the whole experiment and contains groups.
- A *group* (a main branch in Figure 2) pertains to a specific data topic and contains blocks.
- A *block* (a smaller branch in Figure 2) separates groups into different units and contains sections.
- A *section* (a twig in Figure 2) divides a block into smaller units of related data and contains elements.
- An *element* (a leaf in Figure 2) holds a metadata or data value.

Datasets–Paths Through the Experiment

A dataset is a path or linkage through the experiment blocks. Datasets are stored in the file group and connect all eight blocks (two of each remaining group) together. If a given block is needed in a number of datasets, it can be reused multiple times with different collections of other blocks without the need for maintaining copies of it. *Figure 3* illustrates the concept of experiment paths within *SpectroML*:

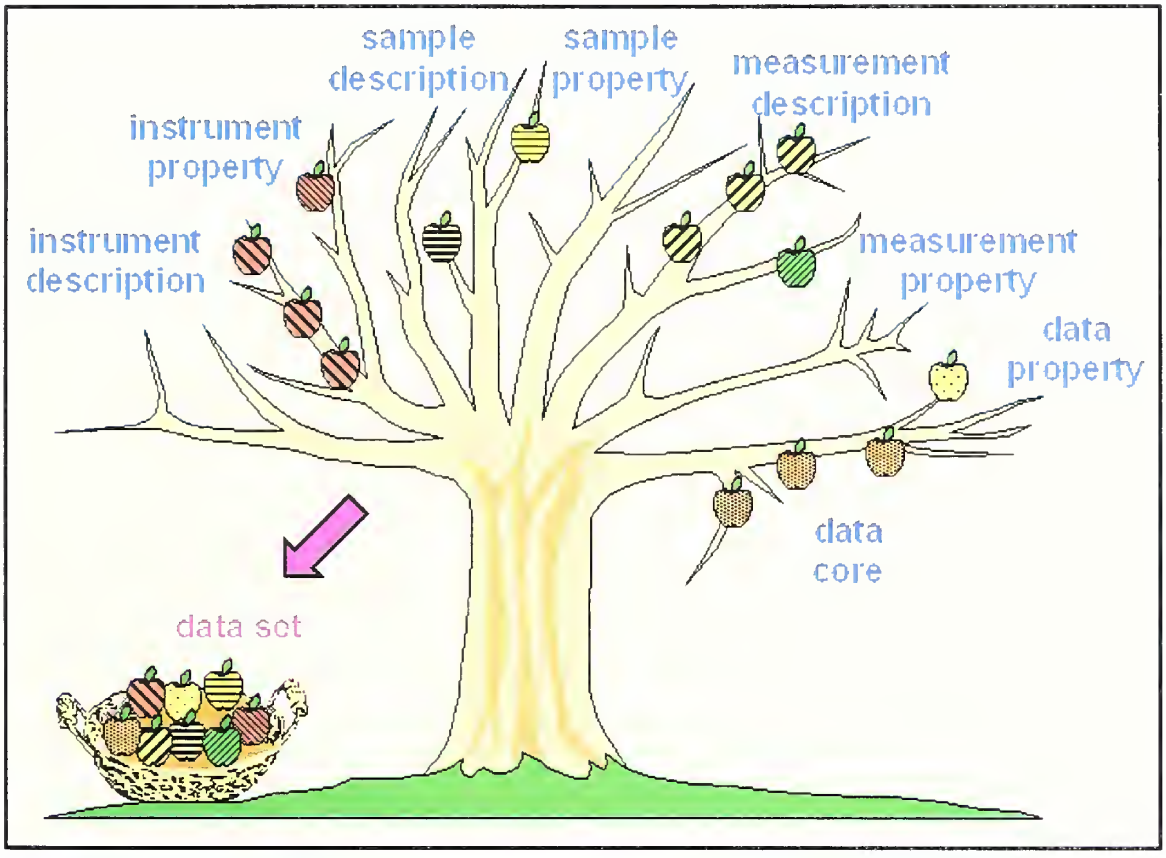

Figure 3 - Experiment paths in *SpectroML*

The eight different colors (or patterns, respectively) represent the eight different block types. Each block type can appear multiple times as a discrete block (an apple in the figure) in the experiment and must have a unique ID (each apple in the figure would need to have a unique ID, for example "id1," "id2," "id3" for each of the three "instrument description" blocks).

A collection of exactly one block of each block type is a *dataset* (the basket of eight apples in the figure). A *path* is the list of the elements of this set consisting of the eight different IDs of the blocks.

Taking Figure 3 as a universe of possible UV/Vis experiments would mean that there were three available instruments each having the same properties; there was one sample with a single set of properties; three possible measurements all with the same properties; and three result data packages all with the same properties.

Data Handling

SpectroML is capable of storing multiple datatypes:

- single data points
- a single spectrum
- multiple spectra
- multi-dimensional data.

Using the typical XML mechanism, data values can be stored in a structure as illustrated in the following example showing three two-dimensional data points:

```
<point>
  <x>1</x>
  <y>2</y>
</point>
<point>
  <x>2</x>
  <y>4</y>
</point>
<point>
  <x>3</x>
  <y>8</y>
</point>
```

But since spectra often contain numerous data points, this simple approach, while functional, would be unwieldy because of its huge amount of overhead. To minimize the overhead, SpectroML can store values in a more compact form by using one tag for the values of one dimension while incorporating the data as a list of values separated by a whitespace character (e.g., space or tab):

```
<values dim="x">1 2 3</values>
<values dim="y">2 4 8</values>
```

The name of the dimension is not fixed in a tag, but is variable in an attribute; this allows as many dimensions as needed. The dimension attribute provides the link between the data and the related metadata elements (e.g., a minimum value or a start value):

```
<values dim="x">1 2 3</values>
<startValue dim="x">1</startValue>
```

In cases where the data values are mathematically related (such as evenly spaced x values), only the initial (starting) value is needed:

```
<values dim="x">1</values>
<values dim="y">2 4 8</values>
```

Of course, when this approach is used, one has to provide the information necessary for calculating the actual values in the corresponding metadata block, such as the increment value for simple accession data.

SpectroML Elements

XML tags are case sensitive. Tags in *SpectroML* are formed according to the following rules:

- Tags contain only letters from the English alphabet (ASCII characters 65-90 and 97-122).
- Tags within the root tag `<SpectroML>` begin with a lower case letter.
- Each new word in a tag starts with an upper case letter for better readability.
- Abbreviations are avoided in tag names as far as possible.
- Wherever a physical value occurs as element content, there must be an attribute for its unit.
- Wherever a data value or calculating property occurs, there must be an attribute for its dimension.

In the tables listed on the following pages, there are different types of elements:

- `elementA:elementB` refers to the tree structure and means that A is the parent of B.
- `element` is a child element of the block element in the header of the table.
- `∞element` is an element which can occur several times.
- `→element` is a sub-element of the element above it.
- `•element` is an attribute of the element above it.

If an element has sub-elements, it cannot contain data itself and has no attribute. It is structured to group information.

Each element that holds data and each attribute must have a datatype. An element always contains only character data, but it can represent a different datatype, e.g., a float value. The following types are used:

- `string` for character data.
- `language` for language setting of elements (ISO 639).
- `date, time` for dates and times (ISO 8601).
- `ID, IDREF, IDREFS` for identifier and references (XML DTD).
- `double, doubles` for float values (IEEE 754-1985) and space-separated doubles.
- `unsignedInt` for positive integer values.
- A dash (`-`) in the type field means that the element does not hold data and thus has no type.

The descriptions of most of the elements are taken from the ASTM definitions [*4*].

Complete Element List

In the following tables of this section, the elements of *SpectroML* are listed in the order of the hierarchy in which they appear. All elements within the four metadata groups are optional, so that the user of *SpectroML* can decide which of these are important for his/her usage.

SpectroML		
Element	**Type**	**Description**
•version	string	The version of SpectroML. For the initial SpectroML only '1.0' exists.
experiment	-	An experiment is one dataset. This provides for the possibility of collecting several experiments in one file.

SpectroML:experiment		
Element	**Type**	**Description**
•type	string	The type of the analytical data in the dataset. For the initial SpectroML only 'UV/Vis' exists.
•language	string	The language of the data elements containing text.
•experimentId	ID	A unique ID for the complete experiment. This allows other experiments to refer to it.
file	-	The file group contains information related to the complete experiment.
instrument	-	The instrument group contains all information related to the instrument used.
sample	-	The sample group contains all information related to the sample used.
measurement	-	The measurement group contains all information related to the measuring process.
data	-	The data group contains all information related to the data and the data themselves.

SpectroML:experiment:file		
Element	**Type**	**Description**
•experimentLinks	IDREFS	A list of references to other experiments within the file.
•externalLinks	string	A list of references to any external data.
title	string	The title of the experiment should be a common name or a short description of the experiment as it would appear on a document or graph. This title is used for all datasets within one experiment.
timeStamp	-	The date and time when the experiment was last modified.
→date	date	The date of the timestamp.
→time	time	The day of the timestamp.
∞path	-	A path connects the eight blocks of one dataset by listing their IDs.
•pathId	ID	A unique ID for the dataset path. This allows paths to be distinguished.
•instrumentDescriptionLink	IDREF	The ID of the instrument description block.
•instrumentPropertyLink	IDREF	The ID of the instrument property block.
•sampleDescriptionLink	IDREF	The ID of the sample description block.
•samplePropertyLink	IDREF	The ID of the sample property block.
•measurementDescriptionLink	IDREF	The ID of the measurement description block.
•measurementPropertyLink	IDREF	The ID of the measurement property block.
•dataPropertyLink	IDREF	The ID of the data property block.
•dataCoreLink	IDREF	The ID of the data core block.
comment	string	A comment provides the opportunity to include additional human-readable information about this block.

SpectroML:experiment:instrument		
Element	**Type**	**Description**
∞instrumentDescription	-	The description block contains all information that describes the instrument and its environment.
∞instrumentProperty	-	The property block contains all instrument settings that can be adjusted.

SpectroML:experiment:sample		
Element	**Type**	**Description**
∞sampleDescription	-	The description block contains all information that describes the sample and its handling.
∞sampleProperty	-	The property block contains all characteristics of the sample.

SpectroML:experiment:measurement		
Element	**Type**	**Description**
∞measurementDescription	-	The description block contains all information that is useful for recording a measurement.
∞measurementProperty	-	The property block contains measurement settings adjusted by the user.

SpectroML:experiment:data		
Element	**Type**	**Description**
∞dataProperty	-	The property block contains information for proper visualization of the raw data.
∞dataCore	-	The core block contains the raw data themselves.

SpectroML:experiment:instrument:instrumentDescription		
Element	**Type**	**Description**
•instrumentDescriptionId	ID	A unique ID for the instrument description block.
instrumentDesignation	-	The designation section contains the instrument's designation, owner, and location.
instrumentApplication	-	The application section contains the instrument's software environment and the operator's name.

SpectroML:experiment:instrument:instrumentProperty		
Element	**Type**	**Description**
•instrumentPropertyId	ID	A unique ID for the instrument property block.
instrumentSetting	-	The setting section contains the instrument's inherent parameters.
instrumentParameter	-	The parameter section contains the instrument's adjustable parameters.

SpectroML:experiment:sample:sampleDescription		
Element	**Type**	**Description**
•sampleDescriptionId	ID	A unique ID for the sample description block.
sampleDesignation	-	The designation section contains the sample's designation, owner location, and handling method.
samplePreparation	-	The preparation section contains the sample's preparation method or source, operator, and date.

SpectroML:experiment:sample:sampleProperty		
Element	**Type**	**Description**
•samplePropertyId	ID	A unique ID for the instrument property block.
sampleAttribute	-	The attribute section contains the sample's inherent properties.
sampleParameter	-	The parameter section contains the sample's adjustable properties.

SpectroML:experiment:measurement:measurementDescription		
Element	**Type**	**Description**
•measurementDescriptionId	ID	A unique ID for the measurement description block.
measurementDesignation	-	The designation section contains the measurement's designation, reference, and owner.
measurementExecution	-	The execution section contains the measurement's project, operator, and date.

SpectroML:experiment:measurement:measurementProperty		
Element	**Type**	**Description**
•measurementPropertyId	ID	A unique ID for the measurement property block.
measurementParameter	-	The parameter section contains the measurement's adjustable parameters.
measurementCorrection	-	The correction section contains the measurement's correction procedures.

SpectroML:experiment:data:dataProperty		
Element	**Type**	**Description**
•dataPropertyId	ID	A unique ID for the data property block.
dataParameter	-	The parameter section contains the data's parameters for proper visualization.
dataCalculation	-	The calculation section contains the data's parameters for calculating the actual result values.

SpectroML:experiment:data:dataCore		
Element	**Type**	**Description**
•dataCoreId	ID	A unique ID for the data core block.
∞values	doubles	Holds a list of data values of one dimension separated by a whitespace.
•dim	string	The name of the dimension.

SpectroML:instrument:instrumentDescription:instrumentDesignation		
Element	**Type**	**Description**
identifier	string	A unique character string or number to identify the instrument within the owner's organization. If available, it should be the barcode of the serial number on the instrument.
manufacturer	string	The name of the manufacturer of the instrument. In the case of an instrument built by the owner's organization itself, this could be the name of the responsible group or person.
model	string	The model name of the instrument as it appears on the instrument or in its manual. In case this is a special version or has special equipment, this could be listed after the name.
owner	-	The owner is the public agency or authority, group, corporation, partnership, or individual, who owns the instrument.
→name	string	Full name of the owner as it would appear on a written document.
→contact	string	Eligible contact information, such as phone number, mail address, or email address.
location	-	The physical location of the instrument within the owner's organization.
→name	string	Complete name of the location as it appears on the room sign.
→contact	string	Eligible contact information for the room or the person who is responsible for the room, such as phone number, mail address, or email address.
comment	string	A comment provides the opportunity to include additional human-readable information about this block.

SpectroML:instrument:instrumentDescription:instrumentApplication		
Element	**Type**	**Description**
software	string	The name of the software used to control the instrument and collect the data.
version	string	The version of the controlling software, including add-ins or service-packs.
operatingSystem	string	The name and version of the operating system on which the instrument's software runs, including add-ins and service-packs.
firmware	string	The revision level of the software in the instrument itself, e.g., its BIOS revision.
operator	-	The operator of the application is the person whose computer account is used for running the software.
→name	string	Full name of the operator as it would appear on a written document.
→contact	string	Eligible contact information, such as phone number, mail address, or email address.
comment	string	A comment provides the opportunity to include additional human-readable information about this block.

SpectroML:instrument:instrumentProperty:instrumentSetting		
Element	**Type**	**Description**
resolution	double	The minimum increment available for the independent variable or fineness of detail reported for the dependent variable.
•unit	string	The unit of the resolution, e.g., 'nm'.
linearDispersion	double	The linear distance that light is dispersed in the plane of the exit slit per unit wavelength.
•unit	string	The unit of the dispersion, e.g., 'mm/nm'.
spectralBandWidthRange	-	The wavelength interval of radiant energy leaving the exit slit measured at half the peak detected power.
→min	double	The shortest wavelength of bandwidth range.
•unit	string	The unit of the minimum, e.g., 'nm'.
→max	double	The longest wavelength of bandwidth range.
•unit	string	The unit of the maximum, e.g., 'nm'.
wavelengthRange	-	The range of wavelength coverage of which an instrument is capable.
→min	double	The shortest wavelength of the range.
•unit	string	The unit of the minimum, e.g., 'nm'.
→max	double	The longest wavelength of the range.
•unit	string	The unit of the maximum, e.g., 'nm'.
absorbanceRange	-	The range of absorbance coverage of which an instrument is capable.
→min	double	The smallest absorbance of the range.
•unit	string	The unit of the minimum, e.g., 'nm'.
→max	double	The largest absorbance of the range.
•unit	string	The unit of the maximum, e.g., 'nm'.
detectorTypes	string	The types or names of the detector used for measuring.
sourceTypes	string	The types or names of the light source.
comment	string	A comment provides the opportunity to include additional human-readable information about this block.

SpectroML:instrument:instrumentProperty:instrumentParameter		
Element	**Type**	**Description**
slitWidth	double	The physical width of the slit of the wavelength selection device.
•unit	string	The unit of the spectral slitwidth, e.g., 'mm'.
spectralSlitWidth	double	The effective spectral bandwidth of the wavelength selection device as defined by the physical slit-width divided by the linear dispersion.
•unit	string	The unit of the spectral slitwidth, e.g., 'nm'.
beamChannel	string	The beam channel used in the instrument.
sampleHolder	string	The type or name of the sample holder unit.
samplePosition	string	The position of the sample within the sample holder unit.
scanSpeed	double	The speed of the scan per time interval or for the whole scan.
•unit	string	The unit of the speed, e.g., 'nm/s'.
integrationTime	double	The amount of time used to measure at one specific wavelength.
•unit	string	The unit of the speed, e.g., 'ms'.
pointSeparation	double	The spacing between two wavelength values in case a whole spectrum is measured.
•unit	string	The unit of the separation, e.g., 'nm'.
comment	string	A comment provides the opportunity to include additional human-readable information about this block.

SpectroML:sample:sampleDescription:sampleDesignation		
Element	**Type**	**Description**
identifier	string	A unique character string or number to identify the sample within the owner's organization. If available, it should be the bar code on the sample.
name	string	A common or trade name for the sample or a name according to a sample index.
owner	-	The public agency or authority, group, corporation, partnership, or individual who owns the sample.
→name	string	Full name of the owner as it would appear on a written document.
→contact	string	Eligible contact information, such as phone number, mail address, or email address.
location	-	The physical location of the sample within the owner's organization.
→name	string	Complete name of the location as it appears on the room sign.
→contact	string	Eligible contact information for the room or the person who is responsible for the room, such as phone number, mail address, or email.
casNumber	string	The registry number of the sample compound according to the Chemical Abstracts Service Index.
formula	string	Molecular formula of the compound in the common notation. Elemental symbols should be arranged with carbon first, followed by hydrogen, and then remaining element symbols in alphabetic order.
storageMethod	string	The name or description of the method to store the sample.
disposalMethod	string	The name or description of the method to dispose of the sample.
comment	string	A comment provides the opportunity to include additional human-readable information about this block.

SpectroML:sample:sampleProperty:samplePreparation		
Element	**Type**	**Description**
procedureMethod	string	The name or short description of the procedure used to prepare the sample.
timeStamp	-	The date and time when the sample was prepared or purchased.
→date	date	The date of the timestamp.
→time	time	The day of the timestamp.
operator	-	The person who was responsible for the sample's preparation.
→name	string	Full name of the operator as it would appear on a written document.
→contact	string	Eligible contact information, such as phone number, mail address, or email address.
supplier	-	The organization, from which the sample was acquired.
→name	string	Full name of the supplier as it would appear on a written document.
→contact	string	Eligible contact information, such as phone number, mail address, or email address.
procedureDescription	string	A detailed description of the procedure method used to prepare the sample.
comment	string	A comment provides the opportunity to include additional human-readable information about this block.

SpectroML:sample:sampleDescription:sampleAttribute		
Element	**Type**	**Description**
molecularWeight	double	The molecular weight of the sample.
•unit	string	The unit of the molecular weight, e.g., 'atomic mass unit'.
meltingPoint	double	The melting point temperature of the sample.
•unit	string	The unit of the temperature, e.g., '°C'.
boilingPoint	double	The boiling point temperature of the sample.
•unit	string	The unit of the temperature, e.g., '°C'.
density	double	The density of the sample.
•unit	string	The unit of the density, e.g., 'g/mL'.
refractiveIndex	double	The refractive index of the sample.
•unit	string	The unit of the refractive index, usually a ratio relative to air at a specific temperature and wavelength.
comment	string	A comment provides the opportunity to include additional human-readable information about this block.

SpectroML:sample:sampleProperty:sampleParameter		
Element	**Type**	**Description**
state	string	The state of the sample at the time of measuring. Usually that is 'gas', 'solid', or 'liquid', but it is also possible to specify other states.
pathLength	double	The distance traveled by the light beam through the sample, usually the internal width of the sample holder.
•unit	string	The unit of the distance, e.g., 'cm'.
amount	double	The amount of the sample. This could be a volume or a mass.
•unit	string	The unit of the amount, e.g., 'mL'.
pressure	double	The pressure of the sample at the time of measuring.
•unit	string	The unit of the pressure, e.g., 'Pa'.
temperature	double	The current temperature of the sample at the time of measuring.
•unit	string	The unit of the temperature, e.g., '°C'.
humidity	double	The current humidity of the sample at the time of measuring.
•unit	string	The unit of the humidity, e.g., '%'.
comment	string	A comment provides the opportunity to include additional human-readable information about this block.

SpectroML:measurement:measurementDescription:measurementDesignation		
Element	**Type**	**Description**
identifier	string	A unique character string or number to identify the measurement within the owner's organization. It can be a barcode or an entry number in a laboratory notebook.
title	string	The title of the experiment part as it would appear on a written document.
owner	-	The public agency or authority, group, corporation, partnership, or individual who owns the measurement results.
→name	string	Full name of the owner as it would appear on a written document.
→contact	string	Eligible contact information, such as phone number, mail address, or email address.
laboratoryReference	string	A reference to any other documentation method of the measurement, e.g., a laboratory notebook.
comment	string	A comment provides the opportunity to include additional human-readable information about this block.

SpectroML:measurement:measurementDescription:measurementExecution		
Element	**Type**	**Description**
project	string	The name of the project within the owner's organization to which the measurement belongs.
timeStamp	-	The date and time when the measurement was performed.
→date	date	The date of the timestamp.
→time	time	The day of the timestamp.
operator	-	The person who performed the experiment.
→name	string	Full name of the operator as it would appear on a written document.
→contact	string	Eligible contact information, such as phone number, mail address, or email address.
comment	string	A comment provides the opportunity to include additional human-readable information about this block.

SpectroML:measurement:measurementProperty:measurementParameter		
Element	**Type**	**Description**
measurementType	string	The type of the measurement, e.g., a sample measurement or blank measurement.
scanMode	string	The name of the experiment mode, e.g., measuring discrete wavelengths or measuring a spectrum.
referenceSample	string	The reference sample used in the measurement.
•sampleDescriptionLink	string	The ID of a sample description block of the reference sample or any other reference which links to it.
filter	string	The name or type of the filter used in the measurement to exclude certain wavelengths.
signalNoise	string	The name or type of the signal-to-noise processing used for correction.
scanNumbers	unsignedInt	The number of scans used to average the final value.
scanDuration	double	The total amount of time to collect all data for this measurement.
•unit	string	The unit of the time, e.g., 's'.
comment	string	A comment provides the opportunity to include additional human-readable information about this block.

SpectroML:measurement:measurementProperty:measurementCorrection		
Element	**Type**	**Description**
qualificationTimeStamp	-	The date and time when the instrument was qualified, usually after purchase or major upgrade.
→date	date	The date of the timestamp.
→time	time	The day of the timestamp.
qualificationReference	string	A reference to a file or other document which contains the data for the qualification test.
proficiencyTimeStamp	-	The date and time when a measurement against a standards institution (e.g., NIST) was performed.
→date	date	The date of the timestamp.
→time	time	The day of the timestamp.
proficiencyReference	string	A reference to a file or other document which contains the data for the proficiency test.
transmittanceTimeStamp	-	The date and time when the transmittance linearity was checked.
→date	date	The date of the timestamp.
→time	time	The day of the timestamp.
transmittanceReference	string	A reference to a file or other document which contains the data for the transmittance linearity test.
wavelengthTimeStamp	-	The date and time when the wavelength calibration was performed.
→date	date	The date of the timestamp.
→time	time	The day of the timestamp.
wavelengthReference	string	A reference to a file or other document which contains the data for the wavelength calibration.
comment	string	A comment provides the opportunity to include additional human-readable information about this block.

SpectroML:data:dataProperty:dataParameter		
Element	**Type**	**Description**
axisLabel	-	Contains the labels for each axis, as they would appear in a graph.
→∞axis	string	The name of the label for one axis.
•dim	string	The name of the dimension of the axis.
axisUnit	-	Contains the units of the data values for each axis.
→∞axis	string	The name of the unit for one axis.
•dim	string	The name of the dimension of the axis.
minimumValue	-	Contains the minimum values for each axis, as set by the operator or measured by the instrument. This can be used during the visualization process to adjust the size of the graph.
→∞value	double	The minimum value for one axis.
•dim	string	The name of the dimension of the axis.
maximumValue	-	Contains the maximum values for each axis, as set by the operator or measured by the instrument. This can be used during the visualization process to adjust the size of the graph.
→∞value	double	The maximum value for one axis.
•dim	string	The name of the dimension of the axis.
comment	string	A comment provides the opportunity to include additional human-readable information about this block.

SpectroML:data:dataProperty:dataCalculation		
Element	**Type**	**Description**
scaleFactor	-	Contains the factors, by which the dataset values must be multiplied to get the actual values. This is used to have convenient numbers in the data core.
→∞value	double	The scale factor for one axis.
•dim	string	The name of the dimension of the axis.
numberPoints	unsignedInt	The total number of points in the dataset. A point is a set of numbers belonging to one measured value.
pointIncrement	-	Contains the fixed increment along each axis from one point to the next point.
→∞value	double	The point increment for one axis.
•dim	string	The name of the dimension of the axis.
startValue	-	Contains the start value along each axis. When values are evenly spaced, they can be calculated by using a fixed increment value and a start value, so they do not have to appear in the data core section.
→∞value	double	The start value for one axis.
•dim	string	The name of the dimension of the axis.
comment	string	A comment provides the opportunity to include additional human-readable information about this block.

SpectroML Files

The structure and its elements were transformed into an XML structure and its related documents. They are fully contained within this document (and can be downloaded from the project page http://www.cstl.nist.gov/nist839/839.04/index.html) and are ready to use. They can be viewed and edited with any common text editor; however, to demonstrate their functionality, some tools are required:

- a browser that can process a DTD
  (e.g., MS Internet Explorer 5.5, http://www.microsoft.com/windows/ie/default.htm)
- a parser that can process XML Schemas
  (e.g., Apache Xerces, http://xml.apache.org)
- a browser that can process an XML Stylesheet
  (e.g., MS Internet Explorer 6.0, http://www.microsoft.com/windows/ie/default.htm)
- a programming package to handle XML documents in applications
  (e.g., Hunter/McLaughlin's JDOM, http://www.jdom.org)

The *SpectroML* sample file, DTD, Schema, and Stylesheet are accessible on the Internet at the site hosted by XML.org (http://www.xml.org).

Sample file

A sample file was created that uses all the elements of *SpectroML*. It is based on a real measurement of a sample of tap water using an HP 8453 diode array spectrophotometer at three specific wavelengths. The file was then completed with information from the laboratory and the literature, and, where necessary, populated with some arbitrary values.

[*see Appendix C for the code listing*]

The current sample file is available at: http://www.xml.org/xml/schema/2c09ac55/*SpectroML*.xml

Document Type Definition

A document type definition (DTD) makes it possible to define precisely the structure of an XML document and to prove whether or not it belongs to a certain type. The DTD is part of the XML specification (http://www.w3.org/TR/2000/REC-xml-20001006).

A DTD for *SpectroML* has been defined, and a *SpectroML* document using it must refer to it by adding the following line (which must be placed below the XML header line):

```
<!DOCTYPE SpectroML SYSTEM "http://www.xml.org/xml/schema/2c09ac55/SpectroML.dtd">
```

The URL (Universal Resource Locator) refers to the *SpectroML* DTD located in the repository at the XML.org site.

The *SpectroML*-DTD is structured as follows:

- The elements must be defined in the order of their hierarchy level.
- Within each structure level, elements are listed alphabetically.
- The elements have a defined order.
- All elements are set as optional, except for the header information in the file group.
- All attributes of elements are set as mandatory.
- The blocks are identified with IDs and referred to via IDREFs.

To validate the *SpectroML* document against its DTD, a parser must read the document in validation mode. Those browsers that can display XML files are capable of doing this.

The DTD mechanism has some drawbacks: for example, it does not permit an arbitrary element order without explicit specification of each of the different possibilities. This is, in most cases, not practical due to the huge number of possible permutations. Furthermore, the DTD concept does not support different datatypes–the only data element type is character data. But since the DTD remains the standard mechanism to define document types for XML, a DTD for *SpectroML* is maintained as well.

XML Schema is another way to define XML document types, and its approach is considerably more flexible. Accordingly, we developed a schema for *SpectroML*. It is important to realize that a document valid against a schema may not necessarily be valid against the corresponding DTD.

The following file contains the document type definition of *SpectroML* as described above:

[*see Appendix C for the code listing*]

The current *SpectroML* DTD is also available at:

http://www.xml.org/xml/schema/2c09ac55/*SpectroML*.dtd

Schema

The XML Schema specification was recently released by the W3C (World Wide Web Consortium) (http://www.w3.org/TR/xmlschema-0) and may replace the DTD concept in future. However, DTD is part of the XML specification, whereas XML Schema is not. Therefore a schema must be specified in the root element via the namespace mechanism (http://www.w3.org/TR/REC-xml-names/). To validate the document against the schema, a parser must read the document in validating mode and be capable of dealing with XML schema. In the future, most browser versions should support this.

The root element of *SpectroML* looks like this:

```
<SpectroML xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://www.xml.org/xml/schema/2c09ac55/SpectroML.xsd">
```

This declares that the instance of the *SpectroML* document type is specified without its own namespace and uses the 2001 XML Schema version. The URL refers to the *SpectroML* schema located in the repository at the XML.org site.

Using a schema provides several advantages over the DTD mechanism:

- It is a regular XML file itself, using a set of tags to describe the document type without special mechanisms like a DTD; this makes it parseable like a *SpectroML* file.
- It permits the specification of different datatypes, for example measured values can be of type float instead of a general CDatatype; typing allows some checking of element content.
- It permits the specification of elements that may occur in any order; for example, elements within a section do not need to appear in a strict order, which is sensible, since they are already tagged and contained within a defined structure.

However, due to the powerful type structure, schema files are more verbose and physically much larger than DTDs.

The *SpectroML* schema was based on the *SpectroML* DTD and uses the same structure. The following new features were added:

- Sections within a block can appear in any order.
- Elements within a section can appear in any order.
- Datatypes were applied to the elements.
- The data values for each dimension have a list type.

[*see Appendix C for the code listing*]

The current *SpectroML* schema is available at:

http://www.xml.org/xml/schema/2c09ac55/*SpectroML*.xsd

Stylesheet

Originally, stylesheets were used to format a document for different types of output media, so that one document could be used with several stylesheets, for example, a different one each for a web site, a handout, or a book to provide an output appropriate to the medium.

XML stylesheets have their own powerful language (XSLT) based on an XML tag set. XSL is a transformation language that takes a source XML document and transforms it using a set of rules into a target document. This is extensively used when transferring data between two systems. The XSLT specification is available at: http://www.w3.org/TR/xslt.html

A commonly used XSL application is the transformation of an XML document into an HTML document to display the XML content in a better way. A stylesheet was developed to view *SpectroML* documents in a browser or another XSLT processor, which adds a line in the XML file that refers to it:

```
<?xml-stylesheet type="text/xsl"
  href="http://www.xml.org/xml/schema/2c09ac55/SpectroML.xsl"?>
```

The URL refers to the SpectroML stylesheet located in the repository at the XML.org site.

Some recent browser versions are capable of processing an XML schema, and this might become a standard browser feature in future. For example, opening an XML document usually yields the display of the tree structure of the file; however, if a stylesheet is assigned to the document, its output could be displayed instead.

The stylesheet for *SpectroML* does the following:

- It lists all elements within the file.
- It groups them by experiment, group, block, and section.
- Attribute values are listed in round brackets.
- IDs of experiments, blocks, and paths are listed in square brackets.
- Datasets are not listed separately, but since the IDs and paths are displayed, one can see which blocks belong together.
- Data points are listed like regular elements as they appear in the file, so there is no special processing or visualization for them.

To enhance the stylesheet, mainly to visualize the data elements, further programming and a much more complicated stylesheet would be necessary. This will likely be done as part of the future application development for *SpectroML*.

[*see Appendix C for the code listing*]

The current *SpectroML* stylesheet is available at:

http://www.xml.org/xml/schema/2c09ac55/*SpectroML*.xsl

SpectroML Applications

There are manifold possibilities for useful applications (Figure 4) for *SpectroML* such as:

- a converter that transfers data from another format into *SpectroML* and vice versa;
- an editor that adds further information to a *SpectroML* file manually that cannot be put in automatically or that can construct *SpectroML* files easily when no automation is available;
- an enhanced stylesheet that displays both metadata and data in a convenient way;
- a virtual library that stores performed experiments and provides for queries through a web portal;
- a viewer that displays *SpectroML* with user-definable views;
- a plug-in for office software that assists in getting experiment data into spreadsheets, presentations, or paper documents;
- a database application that receives *SpectroML* files for storage and then retrieves and transmits them on request.

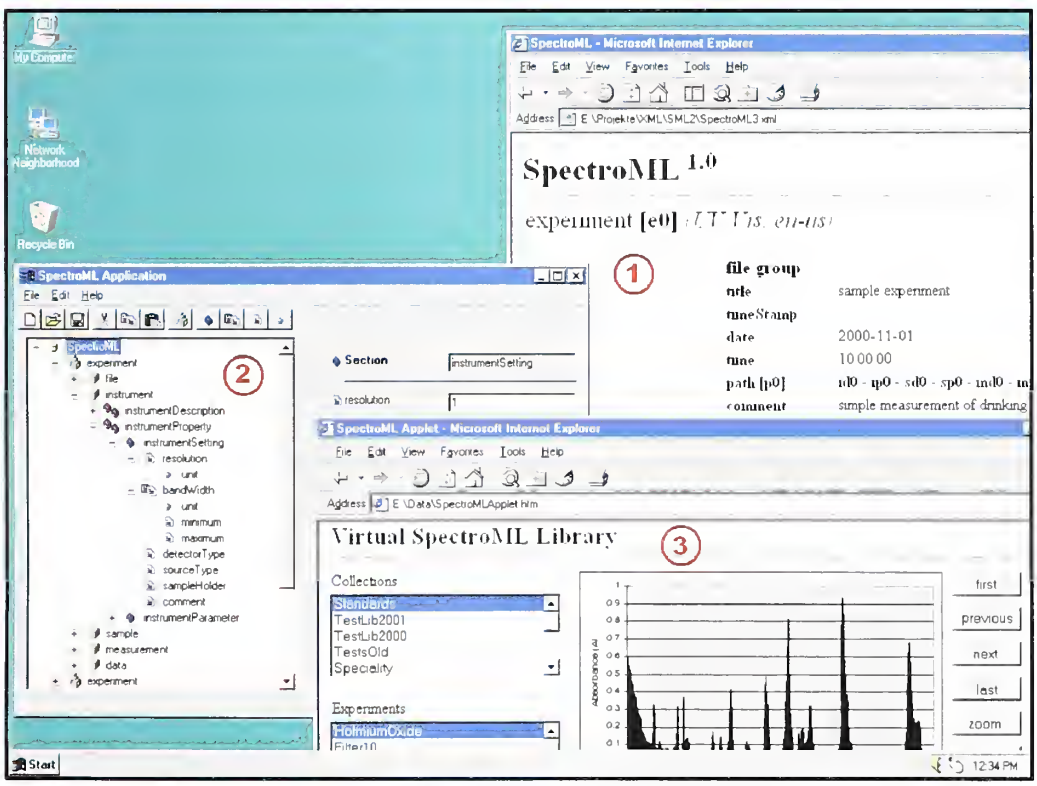

Figure 4 - Some *SpectroML* applications
(1: visualization stylesheet, 2: editor application, 3: visualization applet)

SpectroML API

SpectroML applications need to perform one or more of the following operations:

- read *SpectroML* data to display or process them
- edit *SpectroML* data to alter files or build new ones
- write *SpectroML* data to transfer or store them.

For example, to change an element in a *SpectroML* document, the following steps are necessary:

- read the XML file
- parse the file and validate it
- step through the XML document to find a certain location
- change the content of an element
- rebuild the XML document
- store the XML file.

This procedure requires an extensive amount of programming code each time one deals with a *SpectroML* file. To minimize the programming effort for each new application, it would be very convenient to have a toolkit that provides abstract functions:

- open ("file.xml")
- change (element, content)
- store ("new xml file").

We are building a toolkit in the form of an API (application program interface) that provides a number of functions to work with *SpectroML* files, their metadata, and data.

Digital signatures

In many laboratory environments, it is essential to demonstrate the integrity of experimental data to ensure that any manipulation or tampering can be detected. SpectroML files are regular ASCII (American Standard Code for Information Interchange) text, and therefore it is easy to alter the data, either intentionally or not. Even when there is no need for completely secure data, there often is a need to establish the origin of the data and its subsequent history.

In a laboratory notebook, one certifies a dataset with a written signature. In a similar fashion, computerized datasets can be "signed" by enclosing them with a digital signature. There are several mechanisms to do this, but basically, they all have an algorithm that calculates a unique byte sequence (a signature) based on the content of the file itself. This sequence is delivered together with the data file and a recipient can validate it, as long as he/she knows the algorithm. If an element in a signed file were changed after applying the signature, the subsequent validation would fail.

XML provides a mechanism for digital signatures. It is not yet officially released, but it is fully operable (http://www.w3.org/TR/xmldsig-core/). The following are its main features:

- A signature element contains all information about the validation process.
- The signature element can either become a part of the XML document that it signs, or it can be put into a separate file.
- The signature element refers to an XML document or object and specifies the methods of validation.
- A signature value contains the calculated digital signature.

SpectroML has no built-in tags for signatures, but it can be signed via the mechanism provided by the XML Signature routine.

Conclusion

SpectroML is still largely a proposal. Even though *SpectroML* is ready for use, it has yet to be tested in practice. We are now beginning to write applications to use *SpectroML*. Therefore, reviewer comments and/or suggestions are solicited and will be highly appreciated [gary.kramer@nist.gov].

Now that the *SpectroML* structure and its elements together with its DTD/Schema are in hand, anyone can use *SpectroML*. All that is needed is a text editor and some of the many free tools available on the Internet. To utilize *SpectroML* in an application, one can use an XML API or the soon to be available *SpectroML* API.

At present, *SpectroML* is focused on UV/Vis spectroscopy. But its structure and its flexible data model should make it easily adaptable to other fields of spectroscopy. Our ultimate goal is to build *SpectroML* into a standard that will benefit everyone who deals with spectrometric data.

References

[1] http://www.galactic.com/instruments/spc.htm.

[2] McDonald, R.S.; Wilks, P.A. *Applied Spectroscopy* **1988**, *42*, 151-162; Davies, A.N.; Lampen, P. *Applied Spectroscopy* **1993**, *47*, 1093-1099; Lampen, P.; Hillig, H.; Davies, A.N.; Linscheid, M. *Applied Spectroscopy* **1994**, *48*, 15451552; http://jcamp.isas.dortmund.de; http://members.aol.com/rmcdjcamp/index.htm.

[3] *Standard Specification for Analytical Data Interchange Protocol for Chromatographic Data*, ASTM E 1947-98; *Standard Guide for Analytical Data Interchange Protocol for Chromatographic Data*, ASTM E 1948-98; *Standard Specification for Analytical Data Interchange Protocol for Mass Spectrometry Data*, ASTM E 2077-00; *Standard Guide for Analytical Data Interchange Protocol for Mass Spectrometry Data*, ASTM E 2078-00; http://www.astm.org.

[4] Compilation of ASTM Standard Definitions, 8th Edition, ASTM, Philadelphia, PA, 1994; http://www.astm.org.

Appendix A: Analysis of Existing UV/Vis Data Formats

Introduction

Our decision to create SpectroML did not mean starting from scratch. There was simply no need to do this. We believed that the terminology, data dictionaries, and concepts embodied in existing standards, instrument software, and data interchange formats could be leveraged to facilitate the development of SpectroML. We wanted to take advantage of the large body of work that has been done in the field of spectrometric data interchange rather than re-inventing it. With this concept of reuse firmly in hand, we studied terminology definitions in normative standards [4], spectrometer operation and software manuals, and existing native and interchange formats in hopes of extracting the most useful parts of each.

There are many different data formats for analytical data interchange. Each of them has a different viewpoint and emphasis, and each concentrates on different elements. Therefore, the first step in developing a markup vocabulary for UV/Vis spectroscopy data interchange was to analyze these formats and to extract their most useful parts.

For this document three interchange formats were selected for study:

- GRAMS SPC (Galactic Industries Corp., 9/97) [1]
  (http://www.galactic.com/instruments/spc.htm)
- JCAMP-DX (Joint Committee on Atomic and Molecular Physics Data Exchange, 9/87) [2]
  (http://www.isas-dortmund.de/projects/jcamp/protocol.html)
- ANDI/NetCDF (ASTM E2077, E2078, 3/00) [3]
  (http://enterprise.astm.org/PAGES/E2077.htm)

The documents were examined, and all UV/Vis related items were extracted and compared. This revealed similarities and differences in the approaches used to store the data (data values) and metadata (descriptive elements concerning the data). Combining the best from each format provided a good starting vocabulary and structure for the development of *SpectroML*.

GRAMS SPC

SPC is a format developed by Galactic Industries Corp. that is used internally in their products and as an exchange format between applications. It is designed for a variety of different types of data taken by laboratory analytical instrumentation. SPC consists of a header describing the content of the file, followed by a binary storage area for the instrumentation data, and an optional block for storing additional information.

The header and the data area are in binary format; this means that all the elements are in a defined order and each element has a fixed length. The log block is a text block, where the elements are defined as keys and their values. This section has a number of predefined keys for different types of data files, but is open for user-defined keys, as well.

The structure of the data area depends on the type of data. SPC distinguishes the following types:

- single evenly (one spectrum, evenly spaced X values)
  ⇨ X values calculated, one block for the Y values
- multi evenly (multiple spectra, evenly spaced X values)
  ⇨ X values calculated, multiple blocks for Y values
- single unevenly (one spectrum, unevenly spaced X values)
  ⇨ one block for each X and Y values
- multi unevenly common (multiple spectra, unevenly spaced X values, same in all spectra)
  ⇨ one block for X values, multiple blocks for Y values
- multi unevenly unique (multiple spectra, unevenly spaced X, different for each spectrum)
  ⇨ alternating blocks for X and Y values for each spectrum.

JCAMP-DX

The JCAMP-DX format was developed in the late 1980s and continues to be used for data exchange. It is a character-based format consisting of text lines, each containing a defined keyword and its value. It has a number of required items (core data), followed by optional information and parameters, and a data block in various formats.

Because of its text-based format, each key value contains a number of characters, but its type can be integer or float. The elements do not have to be in a defined order. JCAMP allows building blocks of sub-files for more than one spectrum.

The data block can be organized as:

- a point list (XY..XY or XYZ...XYZ)
  ⇨ data points come in pairs or triplets; this is used for unevenly spaced X values or for better human readability
- an ordinate list (X++(Y..Y))
  ⇨ a line starts with an X value and is followed by a number of Y values; this is used for evenly spaced X values.

The data themselves can appear in the following formats:

- fixed form
  ⇨ each number has a fixed number of characters
- packed form
  ⇨ adjacent values are separated by space or sign
- squeezed form
  ⇨ delimiter, leading digit, and sign are replaced by a pseudo-digit
- difference form
  ⇨ delimiter, leading digit, and sign of the difference between adjacent values are represented by a pseudo-digit
- difference duplicate form
  ⇨ in addition to difference form duplicate values are replaced by the value and the number of its appearance.

ANDI/NetCDF

ANDI (Analytical Data Interchange) is a subset of the NetCDF (Network Compound Document Format) format and is specified for both mass spectrometry and chromatography data interchange. Unlike the SPC and JCAMP-DX formats, each ANDI method has its own protocol and contains very technique-specific information. There is no official protocol for UV/Vis data (although draft versions for infrared and diode array UV/Vis spectroscopy have been circulated), so our analysis is based primarily on the protocol for mass spectrometric data. Much of it proved relevant for spectrophotometric data.

The whole format is defined in a C-language-like structure, where each element has a fixed type and length. When this structure is stored in a file, it becomes one binary block. The elements are divided into categories, and some of them are required for data completeness.

The data structure can contain a number of spectra. Each axis has an array for its values. The data can be organized as:

- pairs or triplets
  ⇨ values at the same position in each array belong together; this is used for unevenly spaced values
- single array
  ⇨ only the Y values are stored in the array, the others are calculated; this is used for evenly spaced values.

UV/Vis Elements

All the UV/Vis-related items gleaned from the formats we examined were extracted and organized into five groups:

- File (header information)
- Instrument (information about the instruments used)
- Sample (information about the processed samples)
- Measurement (information about the measurement process)
- Data (data values and information about its structure)

Each group was divided into sub-groups and the elements were listed together with their datatypes and a description. The datatypes include:

- values (integer and floating point)
- strings (parseable strings and free text)
- items (values or strings out of a defined list)
- arrays (of the previous types).

This is the collection of terms that formed the basis for the development of the *SpectroML*-UV/Vis vocabulary and structure.

File group

The file group contains:

- description
  ⇨ information about the complete dataset
- user
  ⇨ information about the creators of the dataset

File description		
GRAMS		
Name	**Type**	**Description**
file description	`text`	memo or comment that describes the data in the file
file format version	`item`	file format version of Galactic SPC files [old, new, new2]
name	`string`	data file name, optional path and extension
JCAMP		
Name	**Type**	**Description**
jcampdx	`string`	version of JCAMP-DX
title	`text`	concise description of the spectrum
ANDI		
Name	**Type**	**Description**
admin comments	`text`	comments about the dataset identification of the experiment
dataset time stamp	`string`	date and time at which the source file was created (relative to GMT)
experiment title	`text`	meaningful name of the experiment
languages	`string array`	list of human and programming languages delineated for processing
netcdf revision	`string`	revision level of NetCDF data interchange system
source file reference	`string`	adequate information to locate the original dataset
source file version	`string`	version of the data file format

File user		
GRAMS		
Name	**Type**	**Description**
user	text	user's or analyst's name
JCAMP		
Name	**Type**	**Description**
origin	text	name of organization, including address, phone, individual contributors
owner	text	name of owner of a proprietary spectrum, including copyright
ANDI		
Name	**Type**	**Description**
dataset origin	text	name of organization, including address, phone, individual contributors
dataset owner	text	name of owner of a proprietary dataset, including copyright
operator name	text	name of person who ran the equipment that acquired the dataset

Instrument group

The instrument group contains:

- description
  ⇨ general information about the instrument and its manufacturer
- properties
  ⇨ instrument settings that are inherent and information about the instrument environment
- parameters
  ⇨ instrument settings that can be set by the user.

Instrument description		
GRAMS		
Name	**Type**	**Description**
source description	`string`	unique instrument and model name
JCAMP		
Name	**Type**	**Description**
spectro system	`text`	manufacturer's name, model, software system, release number
ANDI		
Name	**Type**	**Description**
comments	`text`	comments about instrument
id	`string`	laboratory's identification code
manufacturer	`string`	name of manufacturer
model number	`string`	model number or name
name	`string`	generic descriptive name
serial number	`string`	manufacturer's serial number

Instrument properties		
GRAMS		
Name	**Type**	**Description**
bdelay	`float`	begin delay (in s)
det	`string`	detector type name
gain	`float`	detector gain factor
pmt	`float`	photomultiplier tube voltage
resolution	`string`	resolution for data collection, including unit of resolution
sdelay	`float`	scan delay (in s)
src	`string`	source type name
swppm	`float`	spectral bandwidth
JCAMP		
Name	**Type**	**Description**
resolution	`float`	nominal resolution (in abscissa units)
ANDI		
Name	**Type**	**Description**
application software	`string`	name and revision level software module
calibration history	`string array`	audit trail of datasets that records calibration history
detector max value	`float`	maximum output value of the detector (in detector units)
detector min value	`float`	minimum output value of the detector (in detector units)
detector potential	`float`	potential of detector (in V)
detector unit	`string`	name of unit of raw data
firmware version	`string`	revision level of instrument firmware, applies to non-data components
operation system	`string`	name and revision level of data system's operating system
resolution	`float`	spectrometer resolution
software version	`string`	revision level of instrument software, applies to non-data components

Instrument parameters		
GRAMS		
Name	**Type**	**Description**
channel	`integer`	detector or beam channel used
detcor	`items`	detector correction mode [on, off]
slit	`float`	slit aperture width
speed	`item`	scan speed or velocity description
JCAMP		
Name	**Type**	**Description**
deltax	`float`	nominal spacing between points
parameters	`text`	list of essential instrumental settings
ANDI		
Name	**Type**	**Description**
point separation	`float`	separation of spectral data points

Sample group

The sample group contains:

- description
  ⇨ description of the sample and its classification
- properties
  ⇨ state and property of the sample
- environment
  ⇨ information about the sample environment, its preparation, and handling.

Sample description		
GRAMS		
Name	**Type**	**Description**
id	`integer`	identification
name	`string`	name
JCAMP		
Name	**Type**	**Description**
beilstein number	`string`	structural formula code according to the Beilstein system
cas name	`string`	name according to conventions as described in CAS Index Guide
cas registry number	`string`	registry number according to Chemical Abstract Service indices, Merck Index or CAS Online
description	`text`	description for compounds, including composition, origin, appearance, interpretations
molform	`string`	molecular formula
names	`string array`	list of common, trade, or other names
wiswesser	`string`	structural formula according to Wiswesser notation

ANDI		
Name	**Type**	**Description**
cas name	string	name according to Chemical Abstract Service
cas number	integer	registry number according to Chemical Abstract Service
chemical formula	string	chemical formula
comments	text	comments about sample
external id	string	number or code assigned by submitter
internal id	string	number or code assigned within the laboratory or LIMS
other names	string array	list of additional names
owner	text	name of sample owner or submitter
receipt time stamp	string	date and time the sample was received or submitted for analysis (relative to GMT)
smiles notation	string	SMILES notation
type	items	type of sample [standard, unknown, control, blank]
wiswesser notation	string	Wiswesser notation

Sample properties		
GRAMS		
Name	**Type**	**Description**
amount	`float`	sample volume or amount
JCAMP		
Name	**Type**	**Description**
bp	`float`	boiling point (in °C)
density	`float`	density (in g/mL³)
mp	`float`	melting point (in °C)
mw	`float`	molecular weight
refractive index	`float`	refractive index (relative to air at 20 °C)
ANDI		
Name	**Type**	**Description**
boiling point	`float`	boiling point (in °C)
chemical mass	`float`	formula chemical mass, computed using average atomic masses for each element
melting point	`float`	melting point (in °C)

Sample environment		
GRAMS		
Name	**Type**	**Description**
solvent	string	solvent used
JCAMP		
Name	**Type**	**Description**
concentration	string array	list of known components and impurities and their concentration and units of concentration
path length	float	length of the path through sample (in cm)
pressure	string	pressure and unit of pressure
state	string	state (gas, liquid, solid...)
temperature	float	temperature (in °C)
ANDI		
Name	**Type**	**Description**
comments	text	comments concerning preparation
disposal information	text	description of disposal procedure
history	text	description of the history of the particular sample, including special handling, treatments
injection time stamp	string	date and time the sample was injected (relative to GMT)
matrix	text	description of natural matrix from which the sample was selected
precautions	text	safety issues when the sample is manually handled
procedure	text	description of procedure used to prepare sample for analysis
procedure name	string	procedure used to select a sample from its natural bulk matrix
sample thickness	float	thickness of the sample (in cm)
state	items	state [solid, liquid, gas, supercritical fluid, plasma, other state]
storage information	text	description of storage location and conditions

Measurement group

The measurement group contains:

- description
  ⇨ general information about the measurement
- parameters
  ⇨ measurement settings influenced by the user.

Measurement description		
GRAMS		
Name	**Type**	**Description**
collection timestamp	`string`	date and time the data were collected
comment	`text`	description of measurement
scanmode	`items`	scan mode [spectrum, time scan, multi wavelength time scan]
scantype	`items`	scan type [sample, zero line, baseline]
JCAMP		
Name	**Type**	**Description**
cross reference	`string array`	cross references to additional spectra of the same sample
date	`string`	date when spectrum was acquired
sampling procedure	`text`	description of mode of observation, including additional information
source reference	`string`	adequate identification to locate original spectrum
time	`string`	time when spectrum was acquired
ANDI		
Name	**Type**	**Description**
experiment type	`string`	type of experiment
sampling technique	`items`	sampling technique [transmission, reflectance, absorbance, diffuse reflectance, other]

Measurement parameters		
GRAMS		
Name	**Type**	**Description**
filter	`string`	optical filter
JCAMP		
Name	**Type**	**Description**
class	`string`	class of spectrum according to Coblentz Class and IUPAC Class
ANDI		
Name	**Type**	**Description**
calibration times	`integer`	number of times the data were calibrated before yielding final results
processed times	`integer`	number of times the data were processed to yield final results
scan numbers	`integer`	number of scans
scan time	`float`	scan time (in s)

Data group

The data group contains:

- parameters
  ⇨ information belonging to the raw data
- processing
  ⇨ information about the processing of the raw data after acquisition
- values
  ⇨ the acquired data values as described above in each format explanation.

Data parameters		
GRAMS		
Name	**Type**	**Description**
first	`float`	X value corresponding to first Y data point
last	`float`	X value corresponding to last Y data point
npts	`integer`	number of data points
winc	`float`	W value increment for 4D data
wplanes	`integer`	number of planes for 4D data
wtype	`items`	allowed X axis type [list of types, e.g., °C]
xtype	`items`	allowed X axis type [list of types, e.g., nm]
yscaling	`integer`	scaling exponent for Y data values
ytype	`items`	allowed Y axis type [list of types, e.g., AU]
zinc	`float`	Z value increment for evenly spaced Z axes
ztype	`items`	allowed X axis type [list of types, e.g., s]

JCAMP		
Name	**Type**	**Description**
firstx	float	first actual abscissa value
firsty	float	actual Y value corresponding to first X value
lastx	float	last actual abscissa value
maxx	float	largest actual X value in the spectrum
maxy	float	largest actual Y value in the spectrum
minx	float	smallest actual X value in the spectrum
miny	float	smallest actual Y value in the spectrum
npoints	integer	number of data points
xfactor	float	factor by which X values are multiplied to obtain actual values
xlabel	string	label for X axis
xunits	items	abscissa units [1/cm, µm, nm, s]
yfactor	float	factor by which Y values are multiplied to obtain actual values
ylabel	string	label for Y axis
yunits	items	ordinate units [transmittance, reflectance, absorbance, arbitrary]
ANDI		
Name	**Type**	**Description**
data points number	integer	number of actual data points
raw data comments	text	comments relevant to the raw data
starting point	float	value of first spectral data point
xaxis label	string	label for X axis
xaxis range	float array	maximum range of X axis, minimum and maximum value
xaxis scale	float	scaling factor applied to the X axis
xaxis unit	items	units for X axis [list of types, e.g., nm]
yaxis label	string	label for Y axis
yaxis range	float array	maximum range of Y axis, minimum and maximum value
yaxis scale	float	scaling factor applied to the Y axis
yaxis unit	items	units for X axis [list of types, e.g., AU]
zaxis label	string	label for Z axis
zaxis range	float array	maximum range of Z axis, minimum and maximum value
zaxis scale	float	scaling factor applied to the Z axis
zaxis unit	items	units for X axis [list of types, e.g., s]

Data processing		
GRAMS		
Name	**Type**	**Description**
correction	items	sample baseline correction mode [on, off]
corrtype	items	type of correction [normal, zero line, zero sra, zero stdref]
namebg	string	background or baseline spectrum filename
namestd	string	standard reference correction filename
namnezr	string	zero reference correction filename
signoise	items	signal/noise processing mode [on, off]
snlevel	float	acceptable signal/noise ratio
JCAMP		
Name	**Type**	**Description**
data processing	text	description of data processing, e.g., including correction, smoothing, subtraction
ANDI		
Name	**Type**	**Description**
error log	string	information about failures of any type
post experiment	string	names of programs used to process raw data after acquisition
pre experiment	string	names of programs run prior to the start of acquisition

Conclusion

The data elements from the three data formats show different levels of detail in each group; often one format concentrates more on a specific group than the others do. Not all elements are of the same importance. It is difficult to decide which elements are necessary and which are not; that can depend on a specific application. But after perusing the list, we found that some elements were not directly part of the experiment, and so we felt that they should be stored separately elsewhere and designated, where necessary, with a link.

However, the extracted group structure provides a good start for designing the corresponding XML structure. There is one root element, which contains five groups and their blocks. The blocks store the collection of data elements. They can then also be divided, depending on their datatype. Taken together these data elements are useful as an initial vocabulary for *SpectroML*, which covers those elements used in most cases for UV/Vis spectroscopy. XML affords access to all its inherent advantages and to the tools that come with it to create a data format more powerful, more flexible, more extensible, and easier to use than any of those currently available.

Appendix B: A Short Introduction to XML

Since some familiarity with XML ("Extensible Markup Language") and its related concepts is essential to understanding *SpectroML*, we provide here a brief, basic background. The purpose to provide the fundamental background needed to understand the structure and elements of *SpectroML*.

However, there are a number of very good tutorials and a huge pool of resources on that topic on the Internet. Here are some recommended starting points:

- The World Wide Web Consortium (www.w3c.org) contains all specifications to XML and its related technologies, together with references to tools and more.
- The XML Industry Portal (www.xml.org) maintains a repository for XML files and has many references to tutorials and more.
- Other related sites include: www.xml.com and www.xml101.com.

Markup

The concept of markup is much older than its use with computers, but it became popular with the development of HTML, the markup language for documents on the Internet. The basic principle of markup is tagging–enclosing parts of a document between a start tag and an end tag:

```
<title>This is a title.</title>
```

Tags can be structured hierarchically to encapsulate or structure related data:

```
<sample>
  <id>1001</id>
  <name>water</name>
</sample>
```

Tags can contain attributes that contain data:

```
<sample id="1001">water</sample>
```

IDs are special attributes. They permit a unique identification of elements and are used to differentiate one element from another.

An XML file is a fully tagged text file; this means that it starts and ends with one root tag, that it contains an arbitrary number of subtags, and that all content is enclosed in tags. An XML file is human-readable, but designed to be processed by computers.

Document Type Definition (DTD)

To ensure that an XML document is valid and well-formed, its document type must be defined. The standard way to do that is to write a DTD and refer to the DTD in the header of the XML file.

The DTD specifies the names of the elements and attributes and their order of appearance. This allows a parser to check a document and initiate further processing, for example, to extract or to change data.

The DTD mechanism has some drawbacks: for example, the datatypes are basically all text types, there is no way to assign datatypes as with a programming language; and a DTD forces elements to appear in predefined order. Despite its shortcomings, it is still the standard way to define XML documents.

XML Schema

XML Schema uses XML tags themselves to define a document type instead of having a unique syntax as DTD. A schema is much more powerful than a DTD; for example, it provides for a variety of datatypes and allows an arbitrary ordering of elements.

XML Schemas have been used for some time, but only recently was the approved specification officially released by the W3C. The schema mechanism will likely replace DTDs in future.

Namespace

Defining tag vocabularies in document types raises the problem of name collision (multiple usage of the same name tag for different entities). The concept of namespaces introduces a unique prefix for each tag, so that multiply defined tags can be distinguished or even used within the same document:
<person1:name>...
<person2:name>...

To declare a namespace, a URL (Universal Resource Locator) is assigned to each prefix. This requires that valid locations for namespace definitions be maintained; otherwise applications that use the namespace may be broken.

Transformation and Stylesheets

A transformation language, XSLT, is used with XML to transform one class of XML documents into another. A common case is transforming an XML document into a HTML (HyperText Markup Language) document to display its data with a network browser. The mapping information for such transformations is contained in a stylesheet. Stylesheets contain rules that define patterns in the XML document and linkages to corresponding output elements

Appendix C: *SpectroML* Code

Sample file

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!-- SpectroML sample file, 6/5/01 -->
<SpectroML version="1.0">
  <experiment type="UV/Vis" language="en-us" experimentId="e0">
    <file experimentLinks="e0" externalLinks="">
      <title>sample experiment</title>
      <timeStamp>
        <date>2000-11-01</date>
        <time>10:00:00</time>
      </timeStamp>
      <path pathId="p0" instrumentDescriptionLink="id0" instrumentPropertyLink="ip0"
        sampleDescriptionLink="sd0" samplePropertyLink="sp0" measurementDescriptionLink="md0"
        measurementPropertyLink="mp0" dataPropertyLink="dp0" dataCoreLink="dc0"/>
      <comment>simple measurement of drinking water</comment>
    </file>
    <instrument>
      <instrumentDescription instrumentDescriptionId="id0">
        <instrumentDesignation>
          <identifier>25547UV-247</identifier>
          <manufacturer>Hewlett Packard</manufacturer>
          <model>HP 8453</model>
          <owner>
            <name>NIST, ACSL</name>
            <contact>gary.kramer@nist.gov</contact>
          </owner>
          <location>
            <name>Filter Lab</name>
            <contact>john.travis@nist.gov</contact>
          </location>
          <comment>UV/Vis diode array spectrometer</comment>
        </instrumentDesignation>
        <instrumentApplication>
          <software>Hewlett-Packard ChemStation</software>
          <version>Win system 1.0</version>
          <operatingSystem>Windows NT 4.0 SP 5</operatingSystem>
          <firmware>1.0</firmware>
          <operator>
            <name>Paul DeRose</name>
            <contact>paul.derose@nist.gov</contact>
          </operator>
          <comment>standard installation, advanced mode</comment>
        </instrumentApplication>
      </instrumentDescription>
      <instrumentProperty instrumentPropertyId="ip0">
        <instrumentSetting>
          <resolution unit="nm">1</resolution>
          <linearDispersion unit="mm/nn">1</linearDispersion>
          <spectralBandWidthRange>
            <minimum unit="nm">1.5</minimum>
            <maximum unit="nm">1.5</maximum>
          </spectralBandWidthRange>
          <wavelengthRange>
            <minimum unit="nm">1.5</minimum>
            <maximum unit="nm">1.5</maximum>
          </wavelengthRange>
          <absorbanceRange>
            <minimum unit="AU">0</minimum>
            <maximum unit="AU">4</maximum>
          </absorbanceRange>
          <detectorTypes>diode array</detectorTypes>
          <sourceTypes>tungsten + deuterium lamp</sourceTypes>
          <comment>standard properties</comment>
```

```
      </instrumentSetting>
      <instrumentParameter>
        <slitWidth unit="mm">1</slitWidth>
        <spectralSlitWidth unit="nm">1</spectralSlitWidth>
        <beamChannel>1</beamChannel>
        <sampleHolder>multi sample unit</sampleHolder>
        <samplePosition>1</samplePosition>
        <scanSpeed unit="ms">500</scanSpeed>
        <pointSeparation unit="nm">1</pointSeparation>
        <comment>standard parameters</comment>
      </instrumentParameter>
    </instrumentProperty>
  </instrument>
  <sample>
    <sampleDescription sampleDescriptionId="sd0">
      <sampleDesignation>
        <identifier>1063546374</identifier>
        <name>water</name>
        <owner>
          <name>NIST, ACSL</name>
          <contact>alexander.ruehl@nist.gov</contact>
        </owner>
        <location>
          <name>Filter Lab</name>
          <contact>john.travis@nist.gov</contact>
        </location>
        <casNumber>7732-18-5</casNumber>
        <formula>H2O</formula>
        <storageMethod>no storage</storageMethod>
        <disposalMethod>water sink</disposalMethod>
        <comment>one time use</comment>
      </sampleDesignation>
      <samplePreparation>
        <procedureMethod>fill</procedureMethod>
        <timeStamp>
          <date>2001-11-01</date>
          <time>09:30:00</time>
        </timeStamp>
        <operator>
          <name>Alexander Ruehl</name>
          <contact>alexander.ruehl@nist.gov</contact>
        </operator>
        <supplier>
          <name>Filter Lab</name>
          <contact>john.travis@nist.gov</contact>
        </supplier>
        <preparationDescription>out of crane</preparationDescription>
        <comment>regular drinking water</comment>
      </samplePreparation>
    </sampleDescription>
    <sampleProperty samplePropertyId="sp0">
      <sampleAttribute>
        <molecularWeight unit="AMU">18.02</molecularWeight>
        <meltingPoint unit="C">0</meltingPoint>
        <boilingPoint unit="C">100</boilingPoint>
        <density unit="g/cc">0.995</density>
        <refractiveIndex unit="rel. air, 20C, 434 nm">1.3404</refractiveIndex>
        <comment>ususal properties</comment>
      </sampleAttribute>
      <sampleParameter>
        <state>liquid</state>
        <pathLength unit="mm">10</pathLength>
        <amount unit="ml">5</amount>
        <pressure unit="torr">760</pressure>
        <temperature unit="K">293</temperature>
        <comment>filled cuvette</comment>
      </sampleParameter>
    </sampleProperty>
  </sample>
  <measurement>
```

```
    <measurementDescription measurementDescriptionId="md0">
      <measurementDesignation>
        <identifier>M876-UV</identifier>
        <title>water analysis</title>
        <owner>
          <name>NIST, Analytical Chemistry Division</name>
          <contact>(301)975-4645</contact>
        </owner>
        <laboratoryReference>printout 11/01/00-1</laboratoryReference>
        <comment>single quick measurement</comment>
      </measurementDesignation>
      <measurementExecution>
        <project>SpectroML</project>
        <timeStamp>
          <date>2001-11-01</date>
          <time>09:30:00</time>
        </timeStamp>
        <operator>
          <name>Alexander Ruehl</name>
          <contact>alexander.ruehl@nist.gov</contact>
        </operator>
        <comment>test</comment>
      </measurementExecution>
    </measurementDescription>
    <measurementProperty measurementPropertyId="mp0">
      <measurementParameter>
        <measurementType>sample</measurementType>
        <scanMode>discrete wavelengths</scanMode>
        <referenceSample sampleDescriptionLink="">empty cuvette</referenceSample>
        <filter>none</filter>
        <signalNoise>none</signalNoise>
        <scanNumbers>1</scanNumbers>
        <scanDuration unit="s">5</scanDuration>
        <comment>no averaging</comment>
      </measurementParameter>
      <measurementCorrection>
        <qualificationTimeStamp>
          <date>2000-07-19</date>
          <time>11:51:00</time>
        </qualificationTimeStamp>
        <qualificationReference>qual.csv</qualificationReference>
        <proficiencyTimeStamp>
          <date>2000-07-19</date>
          <time>14:00:00</time>
        </proficiencyTimeStamp>
        <proficiencyReference>prof.csv</proficiencyReference>
        <transmittanceTimeStamp>
          <date>2000-09-29</date>
          <time>10:05:00</time>
        </transmittanceTimeStamp>
        <transmittanceReference>trans.csv</transmittanceReference>
        <wavelengthTimeStamp>
          <date>2000-09-29</date>
          <time>15:12:00</time>
        </wavelengthTimeStamp>
        <wavelengthReference>wave.csv</wavelengthReference>
        <comment>NTRM correction infos</comment>
      </measurementCorrection>
    </measurementProperty>
  </measurement>
  <data>
    <dataProperty dataPropertyId="dp0">
      <dataParameter>
        <axisLabel>
          <axis dim="x">Wavelength</axis>
          <axis dim="y">Transmittance</axis>
        </axisLabel>
        <axisUnit>
          <axis dim="x">nm</axis>
          <axis dim="y">%T</axis>
```

```
        </axisUnit>
        <minimumValue>
          <value dim="x">270</value>
          <value dim="y">0.029019</value>
        </minimumValue>
        <maximumValue>
          <value dim="x">976</value>
          <value dim="y">0.10616</value>
        </maximumValue>
        <comment>lines at 270 nm, 655nm, 976 nm</comment>
      </dataParameter>
      <dataCalculation>
        <scaleFactor>
          <value dim="x">1</value>
          <value dim="y">1</value>
        </scaleFactor>
        <numberPoints>3</numberPoints>
        <pointIncrement>
          <value dim="x">0</value>
          <value dim="y">0</value>
        </pointIncrement>
        <startValue>
          <value dim="x">270</value>
          <value dim="y">0.029019</value>
        </startValue>
        <comment>discrete points</comment>
      </dataCalculation>
    </dataProperty>
    <dataCore dataCoreId="dc0">
      <values dim = "x">270 655 976</values>
      <values dim = "y">0.10616 0.029019 0.23453</values>
    </dataCore>
  </data>
 </experiment>
</SpectroML>
```

Document type definition

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!-- DTD for SpectroML, 6/5/01 -->
<!ELEMENT SpectroML (experiment+)>
<!ATTLIST SpectroML
  version CDATA #FIXED "1.0">

<!ELEMENT experiment (file, instrument, sample, measurement, data)>
<!ATTLIST experiment
  type CDATA "UV/Vis"
  language CDATA #IMPLIED
  experimentId ID #IMPLIED>
<!ELEMENT file (title, timeStamp, path*, comment?)>
<!ATTLIST file
  experimentLinks IDREFS #IMPLIED
  externalLinks CDATA #IMPLIED>
<!ELEMENT data (dataProperty+, dataCore+)>
<!ELEMENT instrument (instrumentDescription+, instrumentProperty+)>
<!ELEMENT measurement (measurementDescription+, measurementProperty+)>
<!ELEMENT sample (sampleDescription+, sampleProperty+)>

<!ELEMENT dataCore (values*)>
<!ATTLIST dataCore
  dataCoreId ID #IMPLIED>
<!ELEMENT dataProperty (dataParameter?, dataCalculation?)>
<!ATTLIST dataProperty
  dataPropertyId ID #IMPLIED>
<!ELEMENT instrumentDescription (instrumentDesignation?, instrumentApplication?)>
<!ATTLIST instrumentDescription
  instrumentDescriptionId ID #IMPLIED>
<!ELEMENT instrumentProperty (instrumentSetting?, instrumentParameter?)>
<!ATTLIST instrumentProperty
  instrumentPropertyId ID #IMPLIED>
<!ELEMENT measurementDescription (measurementDesignation?, measurementExecution?)>
<!ATTLIST measurementDescription
  measurementDescriptionId ID #IMPLIED>
<!ELEMENT measurementProperty (measurementParameter?, measurementCorrection?)>
<!ATTLIST measurementProperty
  measurementPropertyId ID #IMPLIED>
<!ELEMENT sampleDescription (sampleDesignation?, samplePreparation?)>
<!ATTLIST sampleDescription
  sampleDescriptionId ID #IMPLIED>
<!ELEMENT sampleProperty (sampleAttribute?, sampleParameter?)>
<!ATTLIST sampleProperty
  samplePropertyId ID #IMPLIED>

<!ELEMENT dataCalculation (scaleFactor?, numberPoints?, pointIncrement?, startValue?, comment?)>
<!ELEMENT dataParameter (axisLabel?, axisUnit?, minimumValue?, maximumValue?, comment?)>
<!ELEMENT instrumentApplication (software?, version?, operatingSystem?, firmware?, operator?, comment?)>
<!ELEMENT instrumentDesignation (identifier?, manufacturer?, model?, owner?, location?, comment?)>
<!ELEMENT instrumentParameter (slitWidth?, spectralSlitWidth?, beamChannel?, sampleHolder?,
  samplePosition?, scanSpeed?, pointSeparation?, comment?)>
<!ELEMENT instrumentSetting (resolution?, linearDispersion?, spectralBandWidthRange?, wavelengthRange?,
  absorbanceRange?, detectorTypes?, sourceTypes?, comment?)>
<!ELEMENT measurementCorrection (qualificationTimeStamp?, qualificationReference?,
  proficiencyTimeStamp?, proficiencyReference?, transmittanceTimeStamp?, transmittanceReference?,
  wavelengthTimeStamp?, wavelengthReference?, comment?)>
<!ELEMENT measurementDesignation (identifier?, title?, owner?, laboratoryReference?, comment?)>
<!ELEMENT measurementExecution (project?, timeStamp?, operator?, comment?)>
<!ELEMENT measurementParameter (measurementType?, scanMode?, referenceSample?, filter?, signalNoise?,
  scanNumbers?, scanDuration?, comment?)>
<!ELEMENT sampleAttribute (molecularWeight?, meltingPoint?, boilingPoint?, density?, refractiveIndex?,
  comment?)>
<!ELEMENT sampleDesignation (identifier?, name?, owner?, location?, casNumber?, formula?,
  storageMethod?, disposalMethod?, comment?)>
<!ELEMENT sampleParameter (state?, pathLength?, amount?, pressure?, temperature?, humidity?, comment?)>
<!ELEMENT samplePreparation (procedureMethod?, timeStamp?, operator?, supplier?,
```

```
  preparationDescription?, comment?)>

<!ELEMENT path EMPTY>
<!ATTLIST path
  pathId ID #REQUIRED
  instrumentDescriptionLink IDREF #REQUIRED
  instrumentPropertyLink IDREF #REQUIRED
  sampleDescriptionLink IDREF #REQUIRED
  samplePropertyLink IDREF #REQUIRED
  measurementDescriptionLink IDREF #REQUIRED
  measurementPropertyLink IDREF #REQUIRED
  dataPropertyLink IDREF #REQUIRED
  dataCoreLink IDREF #REQUIRED>

<!ELEMENT absorbanceRange (minimum, maximum)>
<!ELEMENT axisLabel (axis*)>
<!ELEMENT axisUnit (axis*)>
<!ELEMENT location (name, contact)>
<!ELEMENT maximumValue (value*)>
<!ELEMENT minimumValue (value*)>
<!ELEMENT operator (name, contact)>
<!ELEMENT owner (name, contact)>
<!ELEMENT pointIncrement (value*)>
<!ELEMENT proficiencyTimeStamp (date, time)>
<!ELEMENT qualificationTimeStamp (date, time)>
<!ELEMENT scaleFactor (value*)>
<!ELEMENT spectralBandWidthRange (minimum, maximum)>
<!ELEMENT supplier (name, contact)>
<!ELEMENT startValue (value*)>
<!ELEMENT timeStamp (date, time)>
<!ELEMENT transmittanceTimeStamp (date, time)>
<!ELEMENT wavelengthRange (minimum, maximum)>
<!ELEMENT wavelengthTimeStamp (date, time)>

<!ELEMENT amount (#PCDATA)>
<!ATTLIST amount
       unit CDATA #REQUIRED>
<!ELEMENT axis (#PCDATA)>
<!ATTLIST axis
       dim CDATA #REQUIRED>
<!ELEMENT beamChannel (#PCDATA)>
<!ELEMENT boilingPoint (#PCDATA)>
<!ATTLIST boilingPoint
       unit CDATA #REQUIRED>
<!ELEMENT casNumber (#PCDATA)>
<!ELEMENT comment (#PCDATA)>
<!ELEMENT contact (#PCDATA)>
<!ELEMENT date (#PCDATA)>
<!ELEMENT density (#PCDATA)>
<!ATTLIST density
       unit CDATA #REQUIRED>
<!ELEMENT detectorTypes (#PCDATA)>
<!ELEMENT disposalMethod (#PCDATA)>
<!ELEMENT filter (#PCDATA)>
<!ELEMENT firmware (#PCDATA)>
<!ELEMENT formula (#PCDATA)>
<!ELEMENT humidity (#PCDATA)>
<!ATTLIST humidity
       unit CDATA #REQUIRED>
<!ELEMENT identifier (#PCDATA)>
<!ELEMENT laboratoryReference (#PCDATA)>
<!ELEMENT linearDispersion (#PCDATA)>
<!ATTLIST linearDispersion
       unit CDATA #REQUIRED>
<!ELEMENT manufacturer (#PCDATA)>
<!ELEMENT maximum (#PCDATA)>
<!ATTLIST maximum
       unit CDATA #REQUIRED>
<!ELEMENT measurementType (#PCDATA)>
<!ELEMENT meltingPoint (#PCDATA)>
```

```
<!ATTLIST meltingPoint
        unit CDATA #REQUIRED>
<!ELEMENT minimum (#PCDATA)>
<!ATTLIST minimum
        unit CDATA #REQUIRED>
<!ELEMENT model (#PCDATA)>
<!ELEMENT molecularWeight (#PCDATA)>
<!ATTLIST molecularWeight
        unit CDATA #REQUIRED>
<!ELEMENT name (#PCDATA)>
<!ELEMENT numberPoints (#PCDATA)>
<!ELEMENT operatingSystem (#PCDATA)>
<!ELEMENT pathLength (#PCDATA)>
<!ATTLIST pathLength
        unit CDATA #REQUIRED>
<!ELEMENT pointSeparation (#PCDATA)>
<!ATTLIST pointSeparation
        unit CDATA #REQUIRED>
<!ELEMENT preparationDescription (#PCDATA)>
<!ELEMENT pressure (#PCDATA)>
<!ATTLIST pressure
        unit CDATA #REQUIRED>
<!ELEMENT procedureMethod (#PCDATA)>
<!ELEMENT proficiencyReference (#PCDATA)>
<!ELEMENT project (#PCDATA)>
<!ELEMENT qualificationReference (#PCDATA)>
<!ELEMENT referenceSample (#PCDATA)>
<!ATTLIST referenceSample
        sampleDescriptionLink CDATA #IMPLIED>
<!ELEMENT refractiveIndex (#PCDATA)>
<!ATTLIST refractiveIndex
        unit CDATA #REQUIRED>
<!ELEMENT resolution (#PCDATA)>
<!ATTLIST resolution
        unit CDATA #REQUIRED>
<!ELEMENT sampleHolder (#PCDATA)>
<!ELEMENT samplePosition (#PCDATA)>
<!ELEMENT scanDuration (#PCDATA)>
<!ATTLIST scanDuration
        unit CDATA #REQUIRED>
<!ELEMENT scanMode (#PCDATA)>
<!ELEMENT scanNumbers (#PCDATA)>
<!ELEMENT scanSpeed (#PCDATA)>
<!ATTLIST scanSpeed
        unit CDATA #REQUIRED>
<!ELEMENT signalNoise (#PCDATA)>
<!ELEMENT slitWidth (#PCDATA)>
<!ATTLIST slitWidth
        unit CDATA #REQUIRED>
<!ELEMENT software (#PCDATA)>
<!ELEMENT sourceTypes (#PCDATA)>
<!ELEMENT spectralSlitWidth (#PCDATA)>
<!ATTLIST spectralSlitWidth
        unit CDATA #REQUIRED>
<!ELEMENT state (#PCDATA)>
<!ELEMENT storageMethod (#PCDATA)>
<!ELEMENT temperature (#PCDATA)>
<!ATTLIST temperature
        unit CDATA #REQUIRED>
<!ELEMENT time (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT transmittanceReference (#PCDATA)>
<!ELEMENT value (#PCDATA)>
<!ATTLIST value
        dim CDATA #REQUIRED>
<!ELEMENT values (#PCDATA)>
<!ATTLIST values
        dim CDATA #REQUIRED>
<!ELEMENT version (#PCDATA)>
<!ELEMENT wavelengthReference (#PCDATA)>
```

Schema

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!-- Schema for SpectroML, 6/5/01 -->
<schema xmlns="http://www.w3.org/2001/XMLSchema">
  <element name="SpectroML">
    <complexType>
      <sequence>
        <element ref="experiment" maxOccurs="unbounded"/>
      </sequence>
      <attribute name="version" fixed="1.0" type="string"/>
    </complexType>
  </element>

  <element name="experiment">
    <complexType>
      <sequence>
        <element ref="file"/>
        <element ref="instrument"/>
        <element ref="sample"/>
        <element ref="measurement"/>
        <element ref="data"/>
      </sequence>
      <attribute name="type" type="string"/>
      <attribute name="language" type="language"/>
      <attribute name="experimentId" type="ID"/>
    </complexType>
  </element>

  <element name="file">
    <complexType>
      <sequence>
        <element ref="title"/>
        <element ref="timeStamp"/>
        <element ref="path" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="comment" minOccurs="0"/>
      </sequence>
      <attribute name="experimentLinks" type="IDREFS"/>
      <attribute name="externalLinks" type="string"/>
    </complexType>
  </element>

  <element name="data">
    <complexType>
      <sequence>
        <element ref="dataProperty" maxOccurs="unbounded"/>
        <element ref="dataCore" maxOccurs="unbounded"/>
      </sequence>
    </complexType>
  </element>
  <element name="instrument">
    <complexType>
      <sequence>
        <element ref="instrumentDescription" maxOccurs="unbounded"/>
        <element ref="instrumentProperty" maxOccurs="unbounded"/>
      </sequence>
    </complexType>
  </element>
  <element name="measurement">
    <complexType>
      <sequence>
        <element ref="measurementDescription" maxOccurs="unbounded"/>
        <element ref="measurementProperty" maxOccurs="unbounded"/>
      </sequence>
    </complexType>
  </element>
  <element name="sample">
    <complexType>
```

```
      <sequence>
        <element ref="sampleDescription" maxOccurs="unbounded"/>
        <element ref="sampleProperty" maxOccurs="unbounded"/>
      </sequence>
    </complexType>
  </element>

  <element name="dataCore">
    <complexType>
      <sequence>
        <element ref="values" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
      <attribute name="dataCoreId" type="ID"/>
    </complexType>
  </element>
  <element name="dataProperty">
    <complexType>
      <all>
        <element ref="dataParameter" minOccurs="0"/>
        <element ref="dataCalculation" minOccurs="0"/>
      </all>
      <attribute name="dataPropertyId" type="ID"/>
    </complexType>
  </element>
  <element name="instrumentDescription">
    <complexType>
      <all>
        <element ref="instrumentDesignation" minOccurs="0"/>
        <element ref="instrumentApplication" minOccurs="0"/>
      </all>
      <attribute name="instrumentDescriptionId" type="ID"/>
    </complexType>
  </element>
  <element name="instrumentProperty">
    <complexType>
      <all>
        <element ref="instrumentSetting" minOccurs="0"/>
        <element ref="instrumentParameter" minOccurs="0"/>
      </all>
      <attribute name="instrumentPropertyId" type="ID"/>
    </complexType>
  </element>
  <element name="measurementDescription">
    <complexType>
      <all>
        <element ref="measurementDesignation" minOccurs="0"/>
        <element ref="measurementExecution" minOccurs="0"/>
      </all>
      <attribute name="measurementDescriptionId" type="ID"/>
    </complexType>
  </element>
  <element name="measurementProperty">
    <complexType>
      <all>
        <element ref="measurementParameter" minOccurs="0"/>
        <element ref="measurementCorrection" minOccurs="0"/>
      </all>
      <attribute name="measurementPropertyId" type="ID"/>
    </complexType>
  </element>

  <element name="sampleDescription">
    <complexType>
      <all>
        <element ref="sampleDesignation" minOccurs="0"/>
        <element ref="samplePreparation" minOccurs="0"/>
      </all>
      <attribute name="sampleDescriptionId" type="ID"/>
    </complexType>
  </element>
```

```
<element name="sampleProperty">
  <complexType>
    <all>
      <element ref="sampleAttribute" minOccurs="0"/>
      <element ref="sampleParameter" minOccurs="0"/>
    </all>
    <attribute name="samplePropertyId" type="ID"/>
  </complexType>
</element>

<element name="dataCalculation">
  <complexType>
    <all>
      <element ref="scaleFactor" minOccurs="0"/>
      <element ref="numberPoints" minOccurs="0"/>
      <element ref="pointIncrement" minOccurs="0"/>
      <element ref="startValue" minOccurs="0"/>
      <element ref="comment" minOccurs="0"/>
    </all>
  </complexType>
</element>
<element name="dataParameter">
  <complexType>
    <all>
      <element ref="axisLabel" minOccurs="0"/>
      <element ref="axisUnit" minOccurs="0"/>
      <element ref="minimumValue" minOccurs="0"/>
      <element ref="maximumValue" minOccurs="0"/>
      <element ref="comment" minOccurs="0"/>
    </all>
  </complexType>
</element>
<element name="instrumentApplication">
  <complexType>
    <all>
      <element ref="software" minOccurs="0"/>
      <element ref="version" minOccurs="0"/>
      <element ref="operatingSystem" minOccurs="0"/>
      <element ref="firmware" minOccurs="0"/>
      <element ref="operator" minOccurs="0"/>
      <element ref="comment" minOccurs="0"/>
    </all>
  </complexType>
</element>
<element name="instrumentDesignation">
  <complexType>
    <all>
      <element ref="identifier" minOccurs="0"/>
      <element ref="manufacturer" minOccurs="0"/>
      <element ref="model" minOccurs="0"/>
      <element ref="owner" minOccurs="0"/>
      <element ref="location" minOccurs="0"/>
      <element ref="comment" minOccurs="0"/>
    </all>
  </complexType>
</element>
<element name="instrumentParameter">
  <complexType>
    <all>
      <element ref="slitWidth" minOccurs="0"/>
      <element ref="spectralSlitWidth" minOccurs="0"/>
      <element ref="beamChannel" minOccurs="0"/>
      <element ref="sampleHolder" minOccurs="0"/>
      <element ref="samplePosition" minOccurs="0"/>
      <element ref="scanSpeed" minOccurs="0"/>
      <element ref="pointSeparation" minOccurs="0"/>
      <element ref="comment" minOccurs="0"/>
    </all>
  </complexType>
</element>
```

```
<element name="instrumentSetting">
  <complexType>
    <all>
      <element ref="resolution" minOccurs="0"/>
      <element ref="linearDispersion" minOccurs="0"/>
      <element ref="spectralBandWidthRange" minOccurs="0"/>
      <element ref="wavelengthRange" minOccurs="0"/>
      <element ref="absorbanceRange" minOccurs="0"/>
      <element ref="detectorTypes" minOccurs="0"/>
      <element ref="sourceTypes" minOccurs="0"/>
      <element ref="comment" minOccurs="0"/>
    </all>
  </complexType>
</element>
<element name="measurementCorrection">
  <complexType>
    <all>
      <element ref="qualificationTimeStamp" minOccurs="0"/>
      <element ref="qualificationReference" minOccurs="0"/>
      <element ref="proficiencyTimeStamp" minOccurs="0"/>
      <element ref="proficiencyReference" minOccurs="0"/>
      <element ref="transmittanceTimeStamp" minOccurs="0"/>
      <element ref="transmittanceReference" minOccurs="0"/>
      <element ref="wavelengthTimeStamp" minOccurs="0"/>
      <element ref="wavelengthReference" minOccurs="0"/>
      <element ref="comment" minOccurs="0"/>
    </all>
  </complexType>
</element>
<element name="measurementDesignation">
  <complexType>
    <all>
      <element ref="identifier" minOccurs="0"/>
      <element ref="title" minOccurs="0"/>
      <element ref="owner" minOccurs="0"/>
      <element ref="laboratoryReference" minOccurs="0"/>
      <element ref="comment" minOccurs="0"/>
    </all>
  </complexType>
</element>
<element name="measurementExecution">
  <complexType>
    <all>
      <element ref="project" minOccurs="0"/>
      <element ref="timeStamp" minOccurs="0"/>
      <element ref="operator" minOccurs="0"/>
      <element ref="comment" minOccurs="0"/>
    </all>
  </complexType>
</element>
<element name="measurementParameter">
  <complexType>
    <all>
      <element ref="measurementType" minOccurs="0"/>
      <element ref="scanMode" minOccurs="0"/>
      <element ref="referenceSample" minOccurs="0"/>
      <element ref="filter" minOccurs="0"/>
      <element ref="signalNoise" minOccurs="0"/>
      <element ref="scanNumbers" minOccurs="0"/>
      <element ref="scanDuration" minOccurs="0"/>
      <element ref="comment" minOccurs="0"/>
    </all>
  </complexType>
</element>
<element name="sampleAttribute">
  <complexType>
    <all>
      <element ref="molecularWeight" minOccurs="0"/>
      <element ref="meltingPoint" minOccurs="0"/>
      <element ref="boilingPoint" minOccurs="0"/>
```

```
      <element ref="density" minOccurs="0" >
      <element ref="refractiveIndex" minOccurs="0"/>
      <element ref="comment" minOccurs="0"/>
    </all>
  </complexType>
</element>
<element name="sampleDesignation">
  <complexType>
    <all>
      <element ref="identifier" minOccurs="0"/>
      <element ref="name" minOccurs="0"/>
      <element ref="owner" minOccurs="0"/>
      <element ref="location" minOccurs="0"/>
      <element ref="casNumber" minOccurs="0"/>
      <element ref="formula" minOccurs="0"/>
      <element ref="storageMethod" minOccurs="0"/>
      <element ref="disposalMethod" minOccurs="0"/>
      <element ref="comment" minOccurs="0"/>
    </all>
  </complexType>
</element>
<element name="sampleParameter">
  <complexType>
    <all>
      <element ref="state" minOccurs="0"/>
      <element ref="pathLength" minOccurs="0"/>
      <element ref="amount" minOccurs="0"/>
      <element ref="pressure" minOccurs="0"/>
      <element ref="temperature" minOccurs="0"/>
      <element ref="humidity" minOccurs="0"/>
      <element ref="comment" minOccurs="0"/>
    </all>
  </complexType>
</element>
<element name="samplePreparation">
  <complexType>
    <all>
      <element ref="procedureMethod" minOccurs="0"/>
      <element ref="timeStamp" minOccurs="0"/>
      <element ref="operator" minOccurs="0"/>
      <element ref="supplier" minOccurs="0"/>
      <element ref="preparationDescription" minOccurs="0"/>
      <element ref="comment" minOccurs="0"/>
    </all>
  </complexType>
</element>

<element name="absorbanceRange">
  <complexType>
    <all>
      <element ref="minimum"/>
      <element ref="maximum"/>
    </all>
  </complexType>
</element>
<element name="axisLabel">
  <complexType>
    <sequence>
      <element ref="axis" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>
<element name="axisUnit">
  <complexType>
    <sequence>
      <element ref="axis" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
</element>
<element name="location">
```

```
    <complexType>
      <all>
        <element ref="name"/>
        <element ref="contact"/>
      </all>
    </complexType>
  </element>
  <element name="maximumValue">
    <complexType>
      <sequence>
        <element ref="value" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </complexType>
  </element>
  <element name="minimumValue">
    <complexType>
      <sequence>
        <element ref="value" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </complexType>
  </element>
  <element name="operator">
    <complexType>
      <all>
        <element ref="name"/>
        <element ref="contact"/>
      </all>
    </complexType>
  </element>
  <element name="owner">
    <complexType>
      <all>
        <element ref="name"/>
        <element ref="contact"/>
      </all>
    </complexType>
  </element>
  <element name="path">
    <complexType>
      <attribute name="pathId" use="required" type="ID"/>
      <attribute name="instrumentDescriptionLink" use="required" type="IDREF"/>
      <attribute name="instrumentPropertyLink" use="required" type="IDREF"/>
      <attribute name="sampleDescriptionLink" use="required" type="IDREF"/>
      <attribute name="samplePropertyLink" use="required" type="IDREF"/>
      <attribute name="measurementDescriptionLink" use="required" type="IDREF"/>
      <attribute name="measurementPropertyLink" use="required" type="IDREF"/>
      <attribute name="dataPropertyLink" use="required" type="IDREF"/>
      <attribute name="dataCoreLink" use="required" type="IDREF"/>
    </complexType>
  </element>
  <element name="pointIncrement">
    <complexType>
      <sequence>
        <element ref="value" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </complexType>
  </element>
  <element name="proficiencyTimeStamp">
    <complexType>
      <all>
        <element ref="date"/>
        <element ref="time"/>
      </all>
    </complexType>
  </element>
  <element name="qualificationTimeStamp">
    <complexType>
      <all>
        <element ref="date"/>
        <element ref="time"/>
```

```
      </all>
    </complexType>
  </element>
  <element name="scaleFactor">
    <complexType>
      <sequence>
        <element ref="value" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </complexType>
  </element>
  <element name="spectralBandWidthRange">
    <complexType>
      <all>
        <element ref="minimum"/>
        <element ref="maximum"/>
      </all>
    </complexType>
  </element>
  <element name="startValue">
    <complexType>
      <sequence>
        <element ref="value" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </complexType>
  </element>
  <element name="supplier">
    <complexType>
      <all>
        <element ref="name"/>
        <element ref="contact"/>
      </all>
    </complexType>
  </element>
  <element name="timeStamp">
    <complexType>
      <all>
        <element ref="date"/>
        <element ref="time"/>
      </all>
    </complexType>
  </element>
  <element name="transmittanceTimeStamp">
    <complexType>
      <all>
        <element ref="date"/>
        <element ref="time"/>
      </all>
    </complexType>
  </element>
  <element name="wavelengthRange">
    <complexType>
      <all>
        <element ref="minimum"/>
        <element ref="maximum"/>
      </all>
    </complexType>
  </element>
  <element name="wavelengthTimeStamp">
    <complexType>
      <all>
        <element ref="date"/>
        <element ref="time"/>
      </all>
    </complexType>
  </element>

  <element name="amount">
    <complexType>
      <simpleContent>
        <extension base="double">
```

```
        <attribute name="unit" use="required" type="string"/>
      </extension>
    </simpleContent>
  </complexType>
</element>
<element name="axis">
  <complexType>
    <simpleContent>
      <extension base="string">
        <attribute name="dim" use="required" type="string"/>
      </extension>
    </simpleContent>
  </complexType>
</element>
<element name="beamChannel" type="string"/>
<element name="boilingPoint">
  <complexType>
    <simpleContent>
      <extension base="double">
        <attribute name="unit" use="required" type="string"/>
      </extension>
    </simpleContent>
  </complexType>
</element>
<element name="casNumber" type="string"/>
<element name="comment" type="string"/>
<element name="contact" type="string"/>
<element name="date" type="date"/>
<element name="density">
  <complexType>
    <simpleContent>
      <extension base="double">
        <attribute name="unit" use="required" type="string"/>
      </extension>
    </simpleContent>
  </complexType>
</element>
<element name="detectorTypes" type="string"/>
<element name="disposalMethod" type="string"/>
<element name="filter" type="string"/>
<element name="firmware" type="string"/>
<element name="formula" type="string"/>
<element name="humidity">
  <complexType>
    <simpleContent>
      <extension base="double">
        <attribute name="unit" use="required" type="string"/>
      </extension>
    </simpleContent>
  </complexType>
</element>
<element name="identifier" type="string"/>
<element name="laboratoryReference" type="string"/>
<element name="linearDispersion">
  <complexType>
    <simpleContent>
      <extension base="double">
        <attribute name="unit" use="required" type="string"/>
      </extension>
    </simpleContent>
  </complexType>
</element>
<element name="manufacturer" type="string"/>
<element name="maximum">
  <complexType>
    <simpleContent>
      <extension base="double">
        <attribute name="unit" use="required" type="string"/>
      </extension>
    </simpleContent>
```

```
    </complexType>
  </element>
  <element name="measurementType" type="string"/>
  <element name="meltingPoint">
    <complexType>
      <simpleContent>
        <extension base="double">
          <attribute name="unit" use="required" type="string"/>
        </extension>
      </simpleContent>
    </complexType>
  </element>
  <element name="minimum">
    <complexType>
      <simpleContent>
        <extension base="double">
          <attribute name="unit" use="required" type="string"/>
        </extension>
      </simpleContent>
    </complexType>
  </element>
  <element name="model" type="string"/>
  <element name="molecularWeight">
    <complexType>
      <simpleContent>
        <extension base="double">
          <attribute name="unit" use="required" type="string"/>
        </extension>
      </simpleContent>
    </complexType>
  </element>
  <element name="name" type="string"/>
  <element name="numberPoints" type="unsignedInt"/>
  <element name="operatingSystem" type="string"/>
  <element name="pathLength">
    <complexType>
      <simpleContent>
        <extension base="double">
          <attribute name="unit" use="required" type="string"/>
        </extension>
      </simpleContent>
    </complexType>
  </element>
  <element name="pointSeparation">
    <complexType>
      <simpleContent>
        <extension base="double">
          <attribute name="unit" use="required" type="string"/>
        </extension>
      </simpleContent>
    </complexType>
  </element>
  <element name="preparationDescription" type="string"/>
  <element name="pressure">
    <complexType>
      <simpleContent>
        <extension base="double">
          <attribute name="unit" use="required" type="string"/>
        </extension>
      </simpleContent>
    </complexType>
  </element>
  <element name="procedureMethod" type="string"/>
  <element name="proficiencyReference" type="string"/>
  <element name="project" type="string"/>
  <element name="qualificationReference" type="string"/>
  <element name="referenceSample">
    <complexType>
      <simpleContent>
        <extension base="string">
```

```
        <attribute name="sampleDescriptionLink" type="string"/>
      </extension>
    </simpleContent>
  </complexType>
</element>
<element name="refractiveIndex">
  <complexType>
    <simpleContent>
      <extension base="double">
        <attribute name="unit" use="required" type="string"/>
      </extension>
    </simpleContent>
  </complexType>
</element>
<element name="resolution">
  <complexType>
    <simpleContent>
      <extension base="double">
        <attribute name="unit" use="required" type="string"/>
      </extension>
    </simpleContent>
  </complexType>
</element>
<element name="sampleHolder" type="string"/>
<element name="samplePosition" type="string"/>
<element name="scanDuration">
  <complexType>
    <simpleContent>
      <extension base="double">
        <attribute name="unit" use="required" type="string"/>
      </extension>
    </simpleContent>
  </complexType>
</element>
<element name="scanMode" type="string"/>
<element name="scanNumbers" type="unsignedInt"/>
<element name="scanSpeed">
  <complexType>
    <simpleContent>
      <extension base="double">
        <attribute name="unit" use="required" type="string"/>
      </extension>
    </simpleContent>
  </complexType>
</element>
<element name="signalNoise" type="string"/>
<element name="slitWidth">
  <complexType>
    <simpleContent>
      <extension base="double">
        <attribute name="unit" use="required" type="string"/>
      </extension>
    </simpleContent>
  </complexType>
</element>
<element name="software" type="string"/>
<element name="sourceTypes" type="string"/>
<element name="spectralSlitWidth">
  <complexType>
    <simpleContent>
      <extension base="double">
        <attribute name="unit" use="required" type="string"/>
      </extension>
    </simpleContent>
  </complexType>
</element>
<element name="state" type="string"/>
<element name="storageMethod" type="string"/>
<element name="temperature">
  <complexType>
```

```
      <simpleContent>
        <extension base="double">
          <attribute name="unit" use="required" type="string"/>
        </extension>
      </simpleContent>
    </complexType>
  </element>
  <element name="time" type="time"/>
  <element name="title" type="string"/>
  <element name="transmittanceReference" type="string"/>
  <element name="value">
    <complexType>
      <simpleContent>
        <extension base="double">
          <attribute name="dim" use="required" type="string"/>
        </extension>
      </simpleContent>
    </complexType>
  </element>
  <element name="values">
    <complexType>
      <simpleContent>
        <extension base="valueList">
          <attribute name="dim" use="required" type="string"/>
        </extension>
      </simpleContent>
    </complexType>
  </element>
  <element name="version" type="string"/>
  <element name="wavelengthReference" type="string"/>

<simpleType name='valueList'>
  <list itemType='double'/>
</simpleType>
</schema>
```

Stylesheet

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!-- SpectroML stylesheet file, 6/5/01 -->
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html><xsl:apply-templates/></html>
  </xsl:template>

  <xsl:template match="SpectroML">
    <head><title>SpectroML</title></head>
    <body><font size="6"><b>SpectroML <sup><xsl:value-of select="@version"/></sup></b></font>
      <xsl:apply-templates/></body>
  </xsl:template>

  <xsl:template match="experiment">
    <hr/><table><tr><td>
    <font size="5">experiment <b>[<xsl:value-of select="@experimentId"/>]</b>
    <i> (<xsl:value-of select="@type"/>, <xsl:value-of select="@language"/>)</i></font>
      <xsl:apply-templates/>
    </td></tr></table>
  </xsl:template>

  <xsl:template match="file|instrument|sample|measurement|data">
    <tr><td><br/><font size="4"><b><xsl:value-of select="name()"/> group</b></font></td></tr>
      <xsl:apply-templates/>
  </xsl:template>

  <xsl:template match="instrumentDescription|instrumentProperty|sampleDescription|sampleProperty|
    measurementDescription|measurementProperty|dataProperty|dataCore">
    <tr><td><font size="4"><xsl:value-of select="name()"/> block
    <b>[<xsl:value-of select="@*"/>]</b></font></td></tr>
      <xsl:apply-templates/>
  </xsl:template>

  <xsl:template match="instrumentDesignation|instrumentApplication|instrumentSetting|
    instrumentParameter|sampleDesignation|samplePreparation|sampleAttribute|
    sampleParameter|measurementDesignation|measurementExecution|measurementParameter|
    measurementCorrection|dataParameter|dataCalculation">
    <tr><td><font size="3"><b><i> <xsl:value-of select="name()"/> section</i></b></font></td></tr>
      <xsl:apply-templates/>
    <tr><td> </td></tr>
  </xsl:template>

  <xsl:template match="path">
    <tr><td><b>path [<xsl:value-of select="@pathId"/>]:</b></td>
    <td><b><xsl:value-of select="@instrumentDescriptionLink"/> - <xsl:value-of
      select="@instrumentPropertyLink"/> - <xsl:value-of
      select="@sampleDescriptionLink"/> - <xsl:value-of
      select="@samplePropertyLink"/> - <xsl:value-of
      select="@measurementDescriptionLink"/> -  <xsl:value-of
      select="@measurementPropertyLink"/> - <xsl:value-of
      select="@dataPropertyLink"/> - <xsl:value-of select="@dataCoreLink"/></b></td></tr>
      <xsl:apply-templates/>
  </xsl:template>

  <xsl:template match="axis|value|values">
    <tr><td><b><xsl:value-of select="name()"/><i> (<xsl:value-of select="@dim"/>)</i>:</b></td><td>
      <xsl:apply-templates/></td></tr>
  </xsl:template>

  <xsl:template match="*">
    <tr><td><b><xsl:value-of select="name()"/>:</b></td><td>
      <xsl:apply-templates/>
    <xsl:if test="boolean(@*)"><i> (<xsl:value-of select="@*"/>)</i></xsl:if></td></tr>
  </xsl:template>
</xsl:stylesheet>
```