**NMR-ML: Nuclear Magnetic Resonance Spectrometry Markup Language**

Status of This Document

This document presents the first draft of a specification for the nmrML data format developed by the COSMOS (Work Package 2: Standards Development) under governance of the Metabolomics Standards Initiative (MSI). Distribution is unlimited.

Version of This Document

The current version of this document is: Version 1.0; December, 2013.

The version of this document matches the schema version with one trailing decimal point and integer to denote specification documentation updates that do not correspond to a schema update. Thus the version numbers correspond to:

majorVersion.minorVersion.maintenanceVersion.documentationOnlyUpdateVersion.

Every new release of the specification document will be available in the COSMOS code repository under

Change Log since 1.0.0.0

…

# Abstract

The COSMOS EU project and the Metabolomics Standards Initiative (MSI) defines community standards for data representation in metabolomics to facilitate data comparison, exchange and verification. The COSMOS WP2 develops standards for describing NMR raw data. This document presents information to the NMR community about the modelling in XML of the experimental results obtained by NMR of metabolite samples and biomolecular compounds. COSMOS here specifies the missing open standard NMR Markup Language (nmrML) for capturing and disseminating Nuclear Magnetic Resonance spectroscopy data in metabolomics. This is urgently needed as long-term archival format if metabolomic databases are to capture all the formats of metabolomic data, as well as supporting developments in cheminformatics and structural biology.

This XML format is inspired by the PSI mzML format and hence consists of an XSD that is accompanied by a CV providing the terminology to describe the data.

This document should be read in conjunction with a kit of auxiliary files, including the XSD and CV documentations as well as the example instance documents. All files related to this proposal are available for download at <http://nmrml.org>.

Contents

[Abstract 1](#_Toc373328027)

[1. Introduction 2](#_Toc373328028)

[1.1 Background 2](#_Toc373328029)

[1.2 Previous formats 4](#_Toc373328030)

[1.3 Design Philosophy 4](#_Toc373328031)

[1.3.1 Overall set-up 5](#_Toc373328032)

[1.3.2 Data validation by means of validator software and CV-to-XSD mapping files 6](#_Toc373328033)

[1.3.3 Competency Questions for CV development 6](#_Toc373328034)

[1.3.4 Tool support and software availability 7](#_Toc373328035)

[2. Implementation of the Format 7](#_Toc373328036)

[2.1 Concepts and Terminology 7](#_Toc373328037)

[2.2 The nmrML XSD 7](#_Toc373328038)

[2.3 Relationship to Other Efforts and Specifications 14](#_Toc373328039)

[**2.4** **The MSI Nuclear Magnetic Resonance Spectroscopy Controlled Vocabulary (nmrCV)** 16](#_Toc373328040)

[2.5 Resolved Design Issues 19](#_Toc373328041)

[2.5.1 Count attributes 19](#_Toc373328042)

[2.5.2 Numerical value and datetimestamp encoding 19](#_Toc373328043)

[2.5.3 The new CV term problem 20](#_Toc373328044)

[2.6 Other supporting materials 21](#_Toc373328045)

[2.7 Open Issues 23](#_Toc373328046)

[2.8 Comments on Specific Use Cases 23](#_Toc373328047)

[2.8.1 Profile (continuous) spectra vs. centroided (peak list) (peak picked) spectra 23](#_Toc373328048)

[2.9 Other implementation guidelines 23](#_Toc373328049)

[2.10 How to contribute and provide feedback 23](#_Toc373328050)

[3. Model in XML Schema 23](#_Toc373328051)

[3.1 Element <nmrML> 24](#_Toc373328052)

[4. Conclusions 26](#_Toc373328053)

[5. Authors and Contributors 26](#_Toc373328054)

[6. References 26](#_Toc373328055)

[7. Intellectual Property Statement 26](#_Toc373328056)

[8. Appendix A: 27](#_Toc373328057)

[Copyright Notice 27](#_Toc373328058)

# Introduction

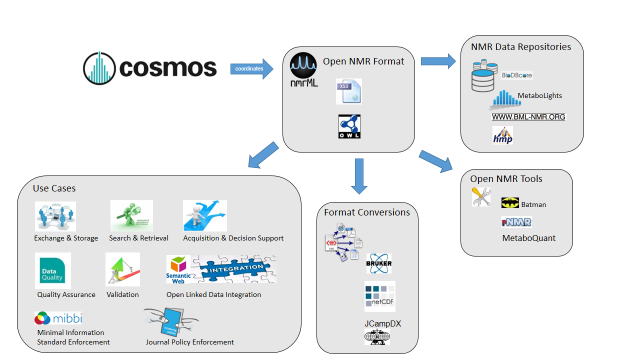
## Background

Nuclear magnetic resonance (NMR) spectroscopy is an important analytical method in metabolomics. As the instrument vendors typically also provide the software to process the vendor specific data, alternative data analysis software needs to put considerable efforts into reading and writing these specific vendor formats. This applies both to commercial software such as NmrPipe, MestReNova (Mnova) or Chenomx NMR Suite, but even more so to community developed open source efforts such as Metaboquant[[1]](#endnote-1) (Matlab-based), the Batman R package or rNMR. Currently existing standard data formats such as the JCAMP family[[2]](#endnote-2) have several drawbacks, especially in metabolomics applications. One problem is that there is no semantic validation of JCAMP-DX files.

From 2005 to 2009 the Metabolomics Standards Initiative (MSI)[[3]](#endnote-3) had kicked off the development to standardize NMR based metabolomics data, including reporting guidelines and an ontology for NMR[[4]](#endnote-4).

To restart this effort, to leverage and canonize existing predecessor artifacts and to coordinate further developments, the COSMOS EU project was granted to create an open exchange data standard to allow metabolomics data, especially NMR raw data, to be shared and stored in an agreed-upon stable and persistent, yet flexible and vendor agnostic XML format. The COSMOS project coordinates efforts from multiple international groups who are working in NMR based metabolomics and NMR software-engineering to design and establish an nmrML data format (<http://nmrml.org>) based on the experience with the PSI (Proteomics Standards Initiative)[[5]](#endnote-5) mzML[[6]](#endnote-6) format for mass spectrometry.

A bird’s eye view on the envisioned nmrML use cases is provided in Fig. 1.



**Figure 1**: Illustration of NMR data management facilitation by means of the common nmrML standard developed in COSMOS

We have formalized these use cases and requirements which the new standard should meet in the Unified Modelling Language (UML) use case diagram (Fig. 2) to illustrate the distinct usages of nmrML in a more standardized manner.



**Figure 2:** UML use case diagram illustrating the multiple application scenarios of the nmrML standard

The following target objectives can be defined for the format: (taken 1:1 from mzML Specification)

1. *The discovery of relevant results,* so that, for example, data sets in a database or public repository that use a particular technique or combination of techniques can be identified and studied by experimentalists during experiment design or data analysis.
2. *The sharing of best practice*, whereby, for example, approaches that have been successful at analysing low abundance analytes can be captured alongside the results produced.
3. *The evaluation of results*, whereby, for example, the number and quality of the spectra recorded from a sample can be assessed in the light of the experimental conditions.
4. *The sharing of data sets,* so that, for example, public repositories can import or export data, multi-site projects can share results to support integrated analysis, or meta-analyses can be performed by third parties from previously published data.
5. *The most comprehensive support of the instruments output,* so that data can be captured [from] all relevant forms of NMR data representations.

The primary focus of the model is to support long-term archiving and sharing, rather than day-to-day laboratory management, although the model is extensible to support context-specific details.

The description of NMR data output and its experimental context requires that models include: (i) the actual data acquired, to a sufficient precision, as well as its associated metadata; and (ii) an adequate description of the instrument characteristics, its configuration and possible pre-processing steps applied. This document details both these parts, as they are required to support the tasks T1 to T5 above.

This document defines a specification and is not a tutorial. As such, the presentation of technical details is deliberately direct. The role of the text is to describe the schema model and justify design decisions made. This document does not provide comprehensive examples of the schema in use. Example documents are provided separately (nmrml.org/examples) and should be examined in conjunction with this document. It is anticipated that tutorial material will be developed in the future to aid implementation. Although the present specification document describes constraints and guidelines related to the content of an nmrML document as well as the availability of tools helping to read and write nmrML, it does not describe any implementation constraints or specifications such as coding language or operating system for software that will generate and/or read nmrML data.

## Previous formats

From 2005 to 2009 the Metabolomics Standards Initiative (MSI)[[7]](#endnote-7) had kicked off the development to standardize NMR based metabolomics data by means of an elaborate XSD (developed by Denis Rubtsov, now used in BMR-NMR), including reporting guidelines and a controlled vocabulary for NMR[[8]](#endnote-8). Another line of NMR standards development, also MSI sanctioned, started in David Wishart’s Group at the [Metabolomics Innovation Centre (TMIC)](http://www.metabolomicscentre.ca/), http://www.metabolomicscentre.ca/exchangeformats.htm). We started the nmrML.xsd development by modification of the J. Cruz XSD predecessor[[9]](#endnote-9) and with additional elements and structures from the BML-NMR XSD developed by Christian Ludwig and D. Rubtsov in Birmingham[[10]](#endnote-10). Both these efforts were integrated by COSMOS, expanding the schema and CV of the TMIC Group, as it was already capturing the basic raw data and had the CV reference mechanism (see Tab 1) already in place. The merged successor artefact described in this document is now called nmrML and invalidates both predecessors as the MSI recommended common standard.

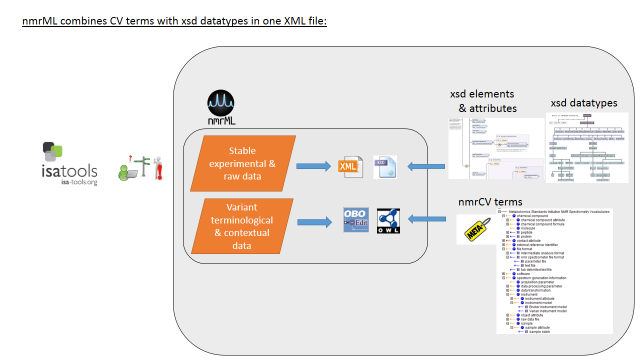
## Design Philosophy

Since the development of nmrML brought together many different philosophies, the primary nmrML designers agreed on the following design principles that would guide its development:

1. Keep the format simple. The data format should be easy for developers to understand and integrate into software. Many elaborate extensions were proposed but most were rejected in favour of a simple implementation.
2. Minimize alternate ways of encoding the same information. Such flexibility, while sometimes touted as a benefit for some products, is bad for data formats as long as semantic equivalence can not be discovered computationally, e.g. via DL reasoning.
3. Build in some flexibility for encoding new important information but keep the format stable. There is a strong desire from companies that develop software for their customers to keep the data format stable over long periods of time with updates to an auxiliary file.
4. Allow 1D and 2D NMR spectra and raw data to be easily shared in a vendor agnostic manner
5. Record enough information about an NMR spectrum acquisition to allow for further processing of the raw spectrum without referring to the original vendor files.
6. The data format should reference the original files for the sake of posterity and in the case where original vendor specific information is required.
7. The data format should be flexible and allow for multiple use cases of NMR experiments.

### Overall set-up

As in our PSI role-model, we agreed on implementing a combined standard using XML schema and accompanying CV terms (Fig. 4). The approach to have an exchange-syntax mixed of an XSD and CV stems from the PSI mzML effort., In areas where the terminology is likely to change faster than the nmrML XSD could be updated and aligned, branching out from an XSD into a CV can compensate for such dynamics in a more flexible way. The CV can be maintained externally and even in a decentralized manner. For example, of new NMR probe types can be represented in an nmrML file by adding new controlled vocabulary terms, without a need for a full schema revision.

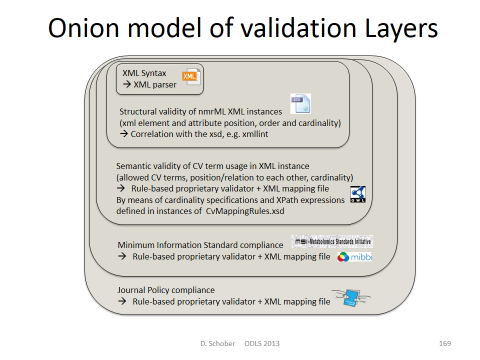


**Figure. 4:** nmrML consists of an XSD specification capturing the more data-near and less variant raw data and a CV in OWL format, capturing the more variant contextual terminology on NMR as a simple taxonomy. For example when capturing information about the NMR instrument configuration there are many different possibilities such as probe heads, auto-samplers, brands, models, etc.

### Data validation by means of validator software and CV-to-XSD mapping files

The XSD-CV set-up also allows for multiple data validation levels to be established (Fig. 5) contributing to data consistency, completeness and overall quality assurance.

To avoid the problem of inconsistent usage of vocabulary terms and the appearance of different dialects encoding the same information in subtly different ways, we are implementing a semantic validator that checks if the CV terms are used correctly within the XML. This semantic validator enforces rules not only making sure that the terms are in the CV, but also that the correct terms are used in the correct location in the XML document and the required terms are present the correct number of times. This allows greater flexibility in the schema, but enforces order in how the CV terms are used. This will require the discipline of using the semantic validator, not just an XML validator. The result is that new technologies or information can be accommodated with adjustments to the controlled vocabulary and validator, not to the schema. Opinions differ on whether this is a benefit or a curse.



**Fig. 5**: XML syntax and structural validity of XML instances (XML element and attribute positions, order and cardinality) can be validated by an XML parser against the XML Schema. Additional mapping files can enforce semantic validity[[11]](#endnote-11) by specifying which CV terms are allowed in an element as well as the order and cardinality those terms. A proprietary validator tool, to be developed for the next deliverable checks that the criteria outlined by the mapping file are being met in a given XML instance. The mapping file combined with the CV can also be used for intelligent support in data acquisition, i.e. when creating an interface that records NMR experiment information it can populate a drop down menu or an autocomplete box with plausible entries.

### Competency Questions for CV development

A set of Competency Questions (CQ) was defined for nmrCV & nmrML. CQs are exemplary queries for a data resource based on the CV. The finished CV should then cover the required areas to annotate the data for successful retrieval and serve to evaluate the format for coverage and structural suitability at a later evaluation phase. Possible queries for raw data annotations could be the following:

• Find 1D 1H NMR spectra from 500MHz field-strength Bruker machines (on human urine samples for doping chemicals).

• Find spectra generated via Bruker CryoProbe and D 2O solvent.

• Find spectra that used a flow high resolution probe in the instrument?

• Find experiments generated with sample pH range from 6.5 to 7.

• Find spectra according to decoupling method for fluxomics (1H{13C}).

• Find NMR spectra that have been Fast Fourier Transformed and were smoothed with Gaussian smoothing.

• Find reference spectra for 1-Methylhistidine with a frequency of 600 MHz.

Additional CQs for nmrCV expansions for Identification and quantification (IdentML & QuantML):

• Find 1D spectra with doublets in ppm range 2.5 to 3.

• Find NMR spectra for changes in metabolites involved in TCA cycle after fat consumption in human.

• How does the aromatic amino acid fraction differ in (hop) plant variants?

• Find spectra that were generated via a certain NMR software.

### Tool support and software availability

One of the aspects of nmrML that will enable its swift adoption is the timely availability of a ready set of open source tools that implements the format (Batman, nNMR, pynmrML,…). With these tools many users are able to begin using the format immediately without coding their own software. Therefore, to insure that nmrML will be adopted quickly and implemented uniformly, the format is presented with several tools that write, read, and validate the format. At the moment, the following software is or will implement nmrML:

1. Two or more converters that convert from vendor formats like Bruker and Varian to nmrML.
2. The popular XYZ parser library that currently supports J-Camp-DX and ….
3. An nmrML structure and semantic validator that checks for correct implementation of files.
4. nmrML bindings for multiple programming languages (Java, R, Python, Ruby,…)

The remainder of this document is structured as follows. Section 2.1 describes a number of concepts and information about the implementation of nmrML, including aspects of terminology, design issues, the controlled vocabulary, etc. The schema model itself is presented in XML schema (XSD) notation in Section 3; some conclusions are presented in Section 5.

# Implementation of the Format

## Concepts and Terminology

This document assumes familiarity with two data modelling notations, namely XML Schema ([www.w3.org/XML/Schema](http://www.w3.org/XML/Schema)) for the XSD part of the standard and OWL, which is an XML application, for the CV part of the standard.

The keywords “MUST,” “MUST NOT,” “REQUIRED,” “SHALL,” “SHALL NOT,” “SHOULD,” “SHOULD NOT,” “RECOMMENDED,” “MAY,” and “OPTIONAL” are to be interpreted as described in RFC-2119 (Bradner 1997).

## The nmrML XSD

An nmrML instance is split up into multiple sections that organize the information in an intuitive way that facilitates easy understanding of the format as well as making development of software application easier. The current top level structure of the nmrML XSD is described in Fig. 4.



**Figure 4:** The first XML elements of the current nmrML.xsd schema, illustrating its main elements. For detailed documentation we refer to the HTML documentation, or the XSD itself, in which extensive element annotations explain the usage of the elements.

The XSD branches out into CV-usage, where:

* The terms describe contextual metadata, rather than NMR raw data
* The terms are unstable, variant & dynamically evolving, or need to be changed and updated often
* The terms refer to fast paced dynamically changing terms such as software names/versions, processing parameters etc.
* The terms are better maintained by a fast reacting NMR user community
* The terms reside at the domains’ leaf node level
* The terms represent search attributes for data querying and database-integration
* The terms should be accessible to rule-based reasoning and validation
* The terms should be exploited by profiting from robust subsumption, e.g. exploiting the taxonomic CV backbone to generalize over query attributes.

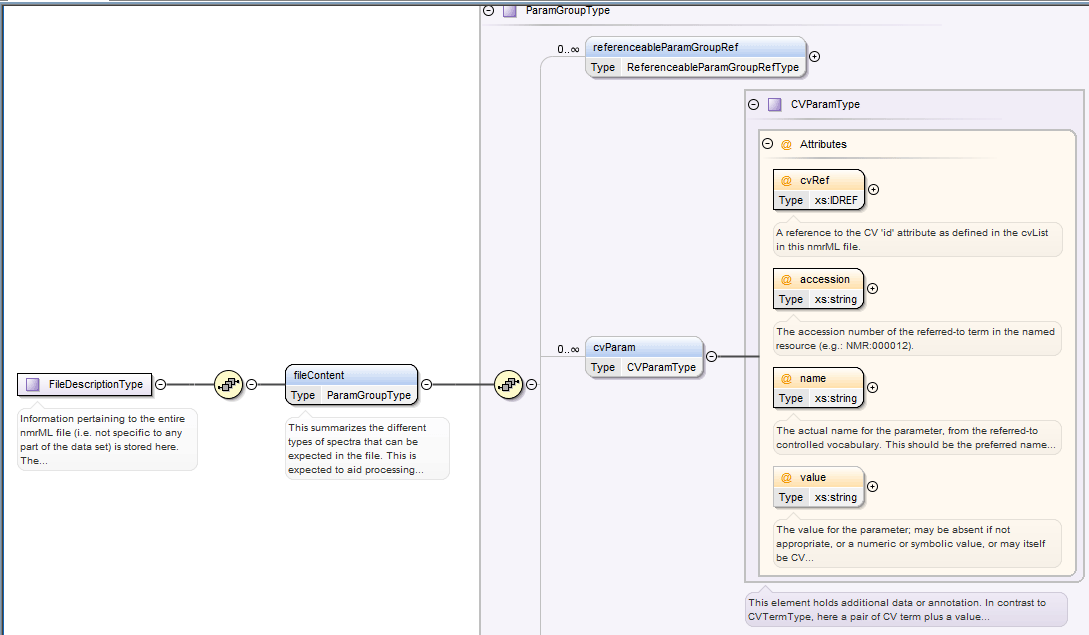
1. **CV term referencing mechanism**

We here outline how CV term usage in nmrML is specified in the XSD. The requirement and modality for a CV term occurrence in an XML instance is specified in the XSD by *reference elements/types* as illustrated in Table 1. Keep in mind that the last element (UserParamType) captures free text and makes no CV reference.

**Table 1:** Illustration how xml element types are used for CV and user parameter entry.

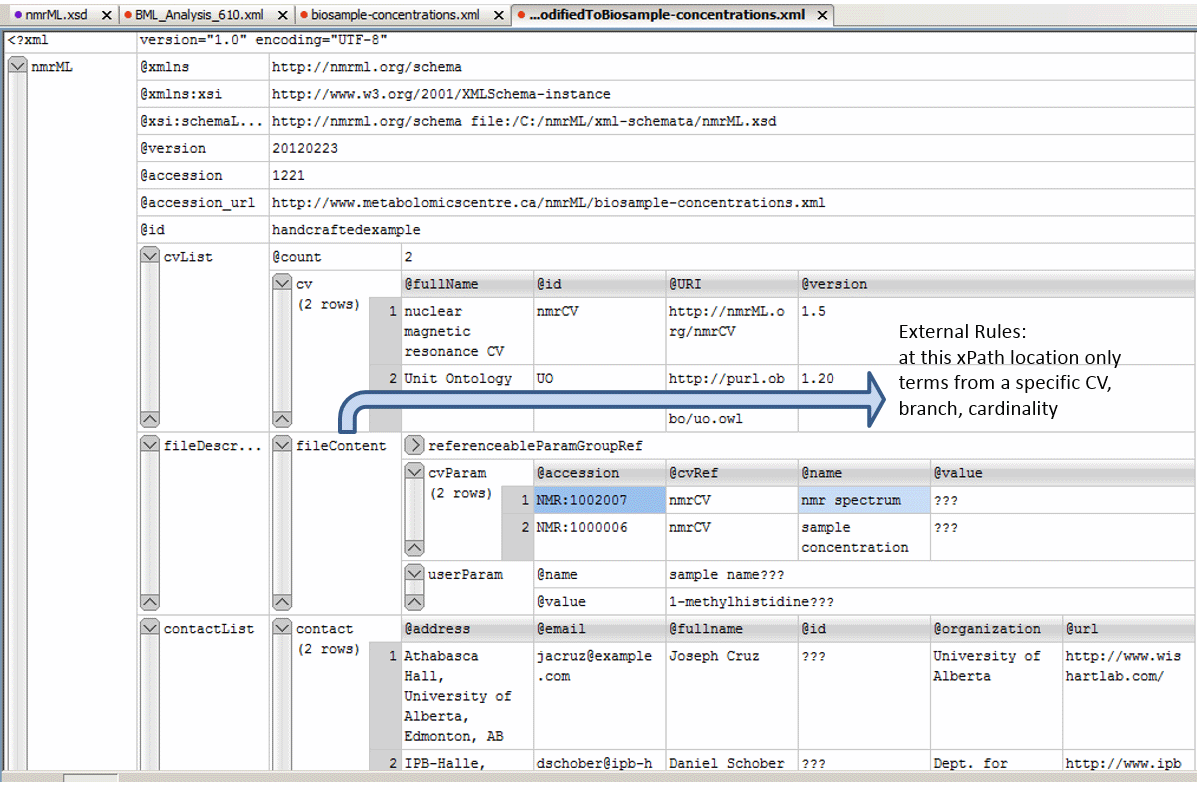
|  |  |  |  |
| --- | --- | --- | --- |
| **Reference Type** | **Definition** | **Attributes** | **Comment** |
| CVTermType | Elements of this type hold additional data or annotation as a simple CV term with no further values (Parameters) associated with it. Only controlled CV terms are allowed here. | *CVRef, accession, name* | The “CVRef” attribute contains an id unique to the XML instance that is defined in the cvList element. This allows for multiple CVs to be referenced unambiguously. The “accession” attribute contains the ID of the CVterm which is unique within the CV. The “name” attribute contains the term which allows using the term in a program (for example displaying it to a user) without requiring the CV file to be downloaded and parsed. |
| CVParamType | Elements of this type hold additional data or annotation. In contrast to CVTermType, here a pair of CV term plus a value (=Parameter) is captured. Only controlled terms are allowed here. | CVRef, accession, name, *value* | The ‘value’ attribute stores the parameter to be captured as value. |
| CVParamWithUnitType | Elements of this type hold additional data or annotation, i.e. a controlled term describing a parameter, as well as a value and a description of the unit the value is recorded in. Only controlled values are allowed here. The unit ontology is typically used to provide the terms for the unit. | CVRef, accession, name, value, *unitCVRef, unitAccession, unitName* | The ‘unitCvRef’, ‘unitAccession’ and ‘unitName’ attributes are used in the same way to describe the unit as the ‘cvRef’, ‘accession’ and ‘name’ terms are used to describe other CVTerms. |
| ValueWithUnitType | Elements of this type hold additional data or annotation. Only controlled values are allowed here. For cases where only a Value with an ontologically defined Unit should be given. Elements of this type hold a value and a reference to the unit the value is recorded in, but is used in locations where the type of value is already defined by the element, but the unit of the value still needs to be recorded. | Value, unitAccession, unitName, unitCvRef |  |
| UserParamType | Elements of this type hold uncontrolled user parameters (essentially allowing free text). For cases where no suitable CV term exists. Before using these, one should verify whether there is an appropriate CV term available, and if so, use the CV term instead. This list can however later be exploited to generate corresponding term requests in given ontologies or CVs. | Name, *valueType,* value, unitAccession, unitName, unitCvRef | The ‘valueType’ attribute |

References to the specific CVs used in the instance are recorded in the ‘cvList’ element at the top, which allows for unambiguous reference to CV terms. The ‘fileDescription’ element captures a general description about the file and its contents which allows for easy categorization of different types of nmrML instances e.g. 1D vs. 2D. The ‘contactList’ element captures information that allows one to contact the original creators of a file in the case that further clarification is needed. The ‘sourceFileList’ contains information about the original files used to make the nmrML instance including files that were required during the acquisition of the spectrum, for example a Varian processing parameter file or a source code file for a pulse program. Similarly the ‘softwareList’ element captures references to software that was used during data acquisition and processing, and may include several different pieces of software. The ‘informationConfigurationList’ element contains information about the configuration of an instrument beyond the acquisition parameters, for example the brand and model of the instrument. The ‘acquisition’ element captures the processing parameters used during the acquisition. Since vendors have their own set of names for each of these parameters, we have standardized them with hopefully intuitive clear names. This element also contains the captured FID data. The ‘spectrumList’ element contains one or more spectra in the frequency domain.



*Figure 5: Specification of CV term usage via the CVParam element in the XSD. The accession attribute encodes the CV term ID and the name encodes the CV term (label).*

An example of how a CV term is used in an example XML instance can be found in Fig. 6.



**Figure 6**: Example XML instance (Oxygen grid view) for the example data from the original J. Cruz XML example[[12]](#endnote-12). This figure illustrates instantiation of CV terms to describe a concrete file content via CV Parameters.

*Further XSD design principles*

* The large binary FID data is stored in the xml at the end of the file in order to allow researchers to look at the first part of the file which contains the more human readable data
* *The Element descriptions were added as annotations to help rendering the schema self-explanatory and intuitive. To foster element selection we added a synonym field per XSD element (as we capture synonyms for the CV). These are useful user entry points when searching for suitable annotations.*
* ML and CV versioning and release policy
  + - * pre-release checks (script based?)

1. **Example implementations** **(nmrML.xml instances)**

We created three example xml files to serve as data-driven check on the format and allow end-users to grasp it more easily. Criteria for good example data are

**Selecting good example NMR data sets for nmrML xml instances**

We defined characteristics of intelligible/intuitive example data set:

* The data was gathered in a prototypical, abundant experiment set up, representative for metabolomics data acquisition
* The data should stem from a simple experimental set-up (e.g. 1D 1H NMR data)
* The data has a published paper available (not a method-, but a research-paper)
* The data has a database entry available, e.g. in MetaboLights[[13]](#endnote-13) or HMDB[[14]](#endnote-14)
* The data has accompanying original data files (FIDs)
* The data is using an abundant vendor format like Bruker or Varian standard files
* The data is associated with a responsive contact person, in case someone needs to get back to the data producers to be able to gather additional information or resolve questions
* The data has been analyzed further with open source tools like Batman or MetaboQuant, so that we can later reproduce the same results based on the converted nmrML data.

According to these criteria we have collated example data sets to be converted into nmrML. These example instances can be found in the corresponding github ‘example’ folder, together with an accompanying readme file illustrating its generation or on the documentation page at nmrml.org/schema.

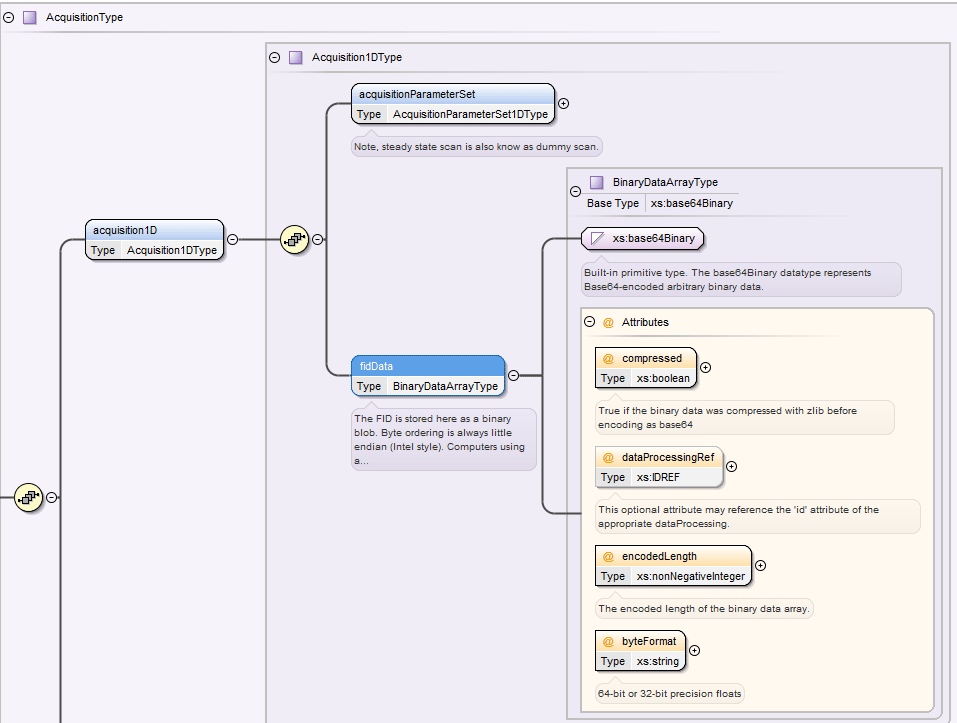
**Example 1:** At first we analyzed, if our schema compensated for all data required by the original predecessor. The original J. Cruz nmrML XML example was taken from <http://www.metabolomicscentre.ca/nmrML/biosample-concentrations.xml> and was transliterated into an nmrML XML instance (see Fig. 6) generated via Oxygen as described at <http://www.oxygenxml.com/doc/ug-editor/topics/xml-schema-instance-generator.html>

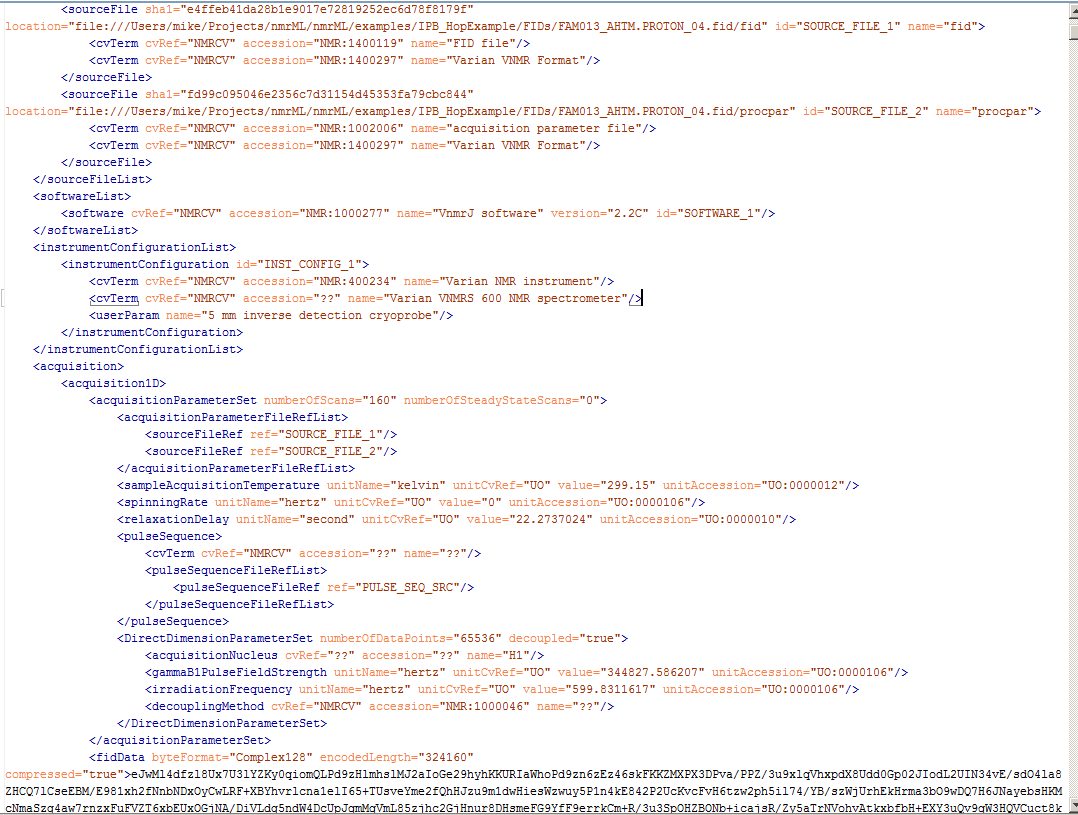
Where the correct entity usage for some values was doubtful, value entries were marked with the String "???". Unused elements and attributes containing the mere default autogenerated values were deleted in the final version.

**Example 2:** An example was created from a reference spectrum obtained from HMDB (<http://www.hmdb.ca/spectra/nmr_one_d/1024>). The file was initially written manually by hand, obtaining values to fill in the file from the Varian procpar file and a python script for encoding the raw FID data into the correct format. This example also proved useful for creating the conversion software since the output could be compared.

**Example 3:** At the IPB, we worked on Hop plant data[[15]](#endnote-15) where thirteen hop ecotypes were profiled for interesting secondary metabolites using MS and NMR in combination. Figure 7 illustrates how 1D acquisition and raw FID data is stored in an nmrML xml instance for one of the hop variants (AHTM).

After we developed the conversion software, more example files were generated from MetaboLights entries MTBLS1 and 25 data. All examples can be browsed in the corresponding GitHub folder.





**Figure 7:** We here provide an nmrML XSD snippet where the FID element is shown (above). The code screenshot (below) illustrates how basic 1D acquisition parameters are stored in the example XML and how Varian raw FID data is stored. The FID is stored as a binary blob (base64 encoded binary data). Byte ordering is always Intel-style little endian. Computers using a different endian style must convert to/from little endian when writing/reading nmrML. The FID should be converted into an array of complex numbers before encoding.

## Relationship to Other Efforts and Specifications

The specification described in this document is not being developed in isolation; indeed, it is designed to be complementary to, and thus used in conjunction with, several existing and emerging models. Related specifications include the following:

1. BioSharing: A consortium that collects and promotes common representations for Minimum Information Standards for Metabolomics. We will build on the BioSharing and the ISA-Tab efforts to harmonize representation of the metadata recommendations with other -omics communities, and use automated tests to ensure the interoperability of the metadata between the involved data producers, -consumers and -repositories.
2. *MIAMET* (Minimal Requirements for a Metabolomic Experiment): *Bino, R.J., Hall, R.D., Fiehn, O. et al .: Potential of metabolomics as a functional genomics tool. Trends Plant Sci. 9 (2004) 418-42*
3. *CIMR (***Core Information for Metabolomics Reporting)** will specify the minimal guidelines reporting metabolomics work. It will do so in a textual form: *Sumner LW, Amberg A, Barrett D, Beale MH, Beger R, et al. 2007. Proposed minimum reporting standards for chemical analysis. Metabolomics 3:211–21* [*http://mibbi.sourceforge.net/projects/CIMR.shtml*](http://mibbi.sourceforge.net/projects/CIMR.shtml)
4. *MI NMR:* [*http://link.springer.com/article/10.1007%2Fs11306-006-0040-4*](http://link.springer.com/article/10.1007%2Fs11306-006-0040-4). The nmrML format has been designed to encode the requirements specified in MI NMR. However, nmrML does not enforce MI NMR compliance itself; nmrML documents may be valid and useful without being fully MI NMR compliant. The nmrML validator has settings to validate at either the basic nmrML level or at a MI NMR compliant level, depending on the needs of the user.
5. SMRS (Standard Metabolic Reporting Structures): The Standard Metabolic Reporting Structure -An Open Standard for Reporting Metabolic Data (http://www.smrsgroup.org/ - March 09, 2005
6. ARMET: ( <http://www.armet.org> )
7. *AniML*
8. *mzML. The XML based open and PSI approved Mass Spectrometry data standard and role model for the development of nmrML.*
9. *mzTab*
10. *ISA-Tab: As research in biomedical and life sciences is increasingly moving towards multi-omics studies, metabolomics must not be an island. The ‘Investigation/Study/Assay’ ISA-Tab format was developed to represent experimental metadata independently from the assay technology used. We will use ISA-Tab to standardize metabolomics reporting requirements and terminologies through customized configurations*
11. *nmrQuant. An extension of the nmrML schema that allows to capture NMR based quantification data on metabolites*
12. *nmrIdent: An extension of the nmrML schema that allows to capture NMR based Identification data on metabolites.*
13. *J-Camp-DX*
14. *Journal Policies*
15. *Metabolights*
16. *Metabostore. This DB will …. It* will also explore semantic web standards that facilitate linked open data (LOD) throughout the biomedical and life science realms, and demonstrate their use for metabolomics data. To connect different sources of data and knowledge, the “Semantic Web for Health Care and Life Sciences Interest Group” (HCLSIG) has started work to represent ISA-Tab metadata as RDF, in compliance with the recommendations of the international Linked Data community (http://linkeddata.org), which will allow to expose any ISA-Tab data set to the semantic web.
17. *NMR-ML Parsers: We use nmrGlue API to access the most important parameter from the Bruker and Varian Files. Our Parser, developed by M Wilson them writes these parameters into an nmrML xml file. The nmrRIO parser parses nmrML and makes its content available to R based statistics tools such as Batman and rNMR. [D.Jacobs and LF tools ?]*

## **The MSI Nuclear Magnetic Resonance Spectroscopy Controlled Vocabulary (nmrCV)**

**Scope and coverage**

This artefact is an MSI approved controlled vocabulary developed under COSMOS governance. This CV was derived from two predecessors (The NMR CV from the David Wishard Group, developed by Joseph Cruz) and the MSI NMR CV developed by Daniel Schober at the EBI. This simple taxonomy of terms (no DL semantics used, for a short overview on the differences of CV, taxonomy and ontology look at http://infogrid.org/trac/wiki/Reference/PidcockArticle) serves the nuclear magnetic resonance markup language (nmrML) with meaningful descriptors to amend the nmrML XML. Metabolomics scientists are encouraged to use this CV to annotate their raw and experimental context data.

**Diversions form the PSI mzML CV design principles**

No separation in different branches to reflect nmrML and nmrQuant CV needs

Correct taxonomy

OWL instead of OBO Format as exchange syntax

**Development history**

After agreement on the set up of development tools (Protégé 4), we formulated our CV design principles, namely agreed on file names, format syntax, namespaces, (auto) term ID schemes, a term obsoletion policy, as well as versioning & release procedures. We analyzed existing CVs on suitability and modelling errors[[16]](#endnote-16). From the given predecessor CVs, we proceeded in a bottom-up and middle-out approach to expand the CV. We first added CV terms as required in the XSD leafs, i.e. where CVTermType, CVParamType, CVParamWithUnitType references occur in XSD elements. After this, we continued with a use-case driven term population, taking multiple data sets (criteria see X) as examples. No high throughput term-additions were attempted in the early design phase, as this would clutter the CV with terms of doubtful need, impair orientation in the term tree as too many terms distract us from getting the main structure right. The nmrCV.owl ontology momentarily contains ~ 600 classes under the nmr: namespace. Around 2000 terms are imported from the Unit Ontology and the BioTopLight upper level ontology.

1. ***Syntax of Format***

We choose the OWL Syntax[[17]](#endnote-17) over the OBO format[[18]](#endnote-18) as exchange syntax for the CV, as the OBO tools are less stable. Because the OBO format is only established in the biology domain (lack of off-the-shelf development tools, OBO expressivity is not as rich as OWL-DL) and there are hence less resources to integrate with.

1. ***Semantics of Format***
2. No OBO Foundry compliance so far, and we subscribe to a moderate/pragmatic realist attitude. We maintain a pure taxonomy without use of axiomatic definitions. Multiple parenthood is allowed, but needs to be maintained manually, as DL reasoning is not possible without DL axiomatisations. Rather flat CV. Taxonomic backbone with few relations used. No OWL DL complexity such as cardinalities, blank nodes, nested class definitions.

At the moment we avoid any usage of object properties from the CV. E.g. for coding the vendor of an NMR instrument, we could have the following axiom in the CV: ‘NMR Instrument’ hasVendor Vendor

Instead, we say in the mapping file that for an Instrument, the Name and Vendor has to be specified. In an equal way we amend CV information describing Software, e.g. the version info is stored in an XSD attribute.

In case we like to be able to convert this owl CV back into the obo format, we should only use DL/owl constructs that are supported by obo. Hence, editors of this CV should take care not to use any higher description logics semantics, i.e. cardinality restrictions or defined terms using constructors. We should start to build the taxonomic backbone first and later connect the main axis via relations. If we want to use restrictions, we should only use existential quantifiers as the OBO format does not support universal quantification. An equal restriction counts if having OWL to RDF conversions in mind, e.g. as used in the BioPortal SPARQL endpoints at <http://www.bioontology.org/wiki/index.php/SPARQL_BioPortal>

1. ***ID scheme***
2. ***Versioning and release procedure***
3. ***Term obsoleting policy***
4. ***Import structure and CV cross references***
5. ***Orthogonality ensusance***

We strive to be non-redundant to existing ontologies. We only include the NMR specific terms right under the TLO terms, assuming at some later point we can close gap via bio upper level ontology, e.g. OBI

E.g. nmrCV:NMR-Sample SubclassOf btl2:Role will then be nmrCV:NMR-Sample SubclassOf OBI:Sample

1. ***Minimal metadata on a CV term***

Representational Unit (RU) metadata is captured via standardized OWL annotation properties drawn from imported artefacts like DC, SKOS and Information Artefact Ontology (IAO). Not all of our terms currently have natural language definitions, as these are time-intensive and not needed for our use case, given the terms are usually self-explanatory. None has deeper provenance data explicitly annotated (there is only an implicit indication from which predecessor CV a term came in the ID ranges). We try to avoid getting stuck in the meta-ether, and have been pragmatic about this.

A term batch-submission table should have the following mandatory fields:

term name (rdfs:label)🡪skos:prefLabel

term definition in natural language (IAO\_0000115)-\_>skos:definition

superclass (ideally a term from the current nmrCV.owl, or an own suggestion)

Optional fields:

synonym (oboInOwl:hasExactSynonym)🡪skos:altLabel

term definition source (dc:source)

dc:contributor

dc:creator🡪dc:author

example of usage (skos:example)

Additional allowed metadata:

We link to the github issue discussions from the terms in the cv via the doap:bug-database annotation property

* + CV term deprecation policy
    - * \_purgatory clean up

1. ***Top Level Ontology usage***

There are a few top and upper level ontologies (TLO) already established. From BFO, OBILight & BioTopLight (btl2), we choose btl2[[19]](#endnote-19) as top level ontology to guide our CV upper level development. The reason was that the WP2 leads are involved in the btl2 development (fast to react) and it provides a proper set of object properties (close to Relations Ontology). At the moment only a few relations from unit ontology (UO) are used. Bridges from btl2 to BFO and other prominent TLOs exist and we can at some later point still switch the TLO, as we do not use any axioms (It is only ~10 classes, so rebinning will be quick). It can be argued why we use a TLO when developing a CV not an Ontology. There has already been a case where the TLO provided modeling restrictions that allowed an automatic DL reasoner to discover CV modelling errors, e.g.<https://github.com/nmrML/nmrML/issues/62>

At some later point we might switch to BFO (as long as temporal models are not understandable in owl, all formal distinguishing between Roles, Processes, Functions, Plans, … are blah blah) or remove the TLO imports altogether to render the CV simpler and increase user compliance.

Although the structure of the CV and the nmrML schema are related, the details of which terms are allowed/recommended in a given schema section is reported in the mapping file. The mapping file is a list of associations between a cvParam element in a specific schema location (described with an XPath) and the branches of the CV terms expected in that location. This file is read and interpreted by the validator, checking that the data annotation is consistent. The mapping file needs to be checked and eventually updated when the CV terms or structure are changed.

1. **External ontology term reference and import mechanism**

There are four possible ways to reuse existing CV terms from other ontologies. We used the first method.

1. Use the terms in the CV by ID reference (e.g. as done with IAO metadata). This option is fast and flexible, but no metadata on used terms available.

2. Use the MIREOT term referencing method. This option is too complicated and relies on outdated scripts

3. Use full owl:import statements (e.g. as done for UO). This option however clutters the CV with seldom used terms, occupies RAM, but retains all metadata. This option is overshot for most use cases.

4. Use dbxref statements. These are easy but not a standard way in OWL (these annotation properties are provided by the OBOinOWL namespace).

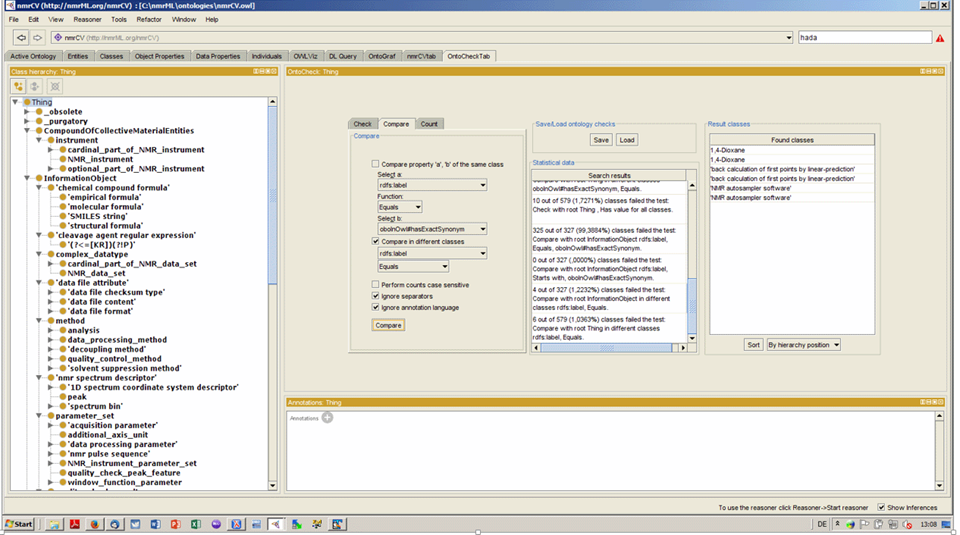
* + - * Interfacing with external ontologies/CVs (imports, mireoting)
        + foaf?, units, pato, ChEBI, Obi, IAO, Newt, …
    - How to deal with violating design principles i.e. DL class post-coordination vs. CV term pre-coordination ? (DS, PRS)
    - Agree on CV term-gathering policy
      * drawing new terms from external users’ requests
      * drawing new terms from freetext fields in xml ?
      * draw terms from ISA tools ?

1. **G. CV term naming conventions**

We apply a labelling scheme in accordance to <http://www.obofoundry.org/wiki/index.php/FP_012_naming_conventions>.

We strived to avoid redundance between XSD and CV and to keep names in coherence to PSI mzML, as in the future there might be the case where people have to orient themselves in both PSI and MSI Standards, i.e. when considering integrated systems biologic research questions. This means we have to find a cutoff between a) making the ML and ontology more intuitive, and b) keeping names similar to PSI MLs…..

The OntoCheck P.4 plugin[[20]](#endnote-20) is used (Fig. 8) to avoid term redundancies, i.e. to check on redundant labels. OntoCheck detected that ‘TecMag’ was included twice, once under http://nmrML.org/nmrCV#NMR\_400285 (NMR data format) and once under http://nmrML.org/nmrCV#NMR:1400255 (NMR\_vendor). This redundancy could then be removed by specifying a more explicit label.



**Figure 8:** A screenshot displaying maintenance of the CV in the ontology editor Protégé 4. The OntoCheck Tab is shown which displays the CV term hierarchy to the left and allows to specify and label comparison check to discover redundant labels.

1. **Domain dependent Design principles**

NMR magnets are captured down to the level of vendor magnet types. The available frequencies are not captured in order to avoid combinatorial explosion of terms.

Reduplication in disjoint trees, e.g. software names or Vendors in file formats, …

Classifying spectrum processing methods according to pre/post acquisition, or pre/post FT ?

Agreed on Time domain Processing and Frequency domain processing

Border nmrCV (raw data) to nmrIdentCV ?

 Both in one & the same nmrCV file, but could add slim indicator for that.

--

The nmrCV should be dynamically maintained via the COSMOS nmrML mailing lists (<https://groups.google.com/forum/?hl=en#!forum/nmrml/join> ) that allows any user to request new terms in agreement with the community involved. Once a consensus is reached among the community the new terms are added within few days. If there is no obvious consensus, the CV coordinators committee should vote and make a decision. A new psi-ms.obo should then be released by updating the file on the CVS server without changing the name of the file (this would alter the propagation of the file to the OBO website and to other ontology services that rely on file stable URI). For this reason an internal version number with two decimals (x.y.z) should be increased:

* x should be increased when a first level term are renamed added deleted or rearranged in the structure. Such rearrangement is supposed to be rare and is very likely to have repercussion on the mapping.
* y should be increased when any other term except the first level one is altered.
* z should be increased when there is no term addition or deletion but just editing on the definitions or other minor changes.

It was decided that the CV not contain “unknown” terms as much as possible, thus there is no “unknown instrument”, etc. There are two cases where it is tempting to use unknown:

1. The information is really known, but none of the existing terms fit. In this case, instead of choosing “unknown”, a user should send email to the psidev-ms-vocab list proposing a new term. The CV committee should approve and add the desired term, or point out an existing synonym that should be used. Then once the term has been assigned an accession number and official name, then the user should begin using that.
2. The second case is there the information is truly not known. We want to avoid cases where the user picks something just to appease a validator. In this case, the use of the base class is recommended, e.g.:

<instrumentConfiguration id="unknownInstrument">

<cvParam cvRef="MS" accession="MS:1000031" name="instrument model"/>

</instrumentConfiguration>

The rationale is that the most specific information available is provided, i.e. that a mass spectrometer of some model generated the data, but it is not known which one. This also avoids the situation where the tags are optional. If optional tags are not provided, it is never clear whether the information is truly unknown, or the writer simple forgot to write the information or was too lazy to write the information.

In the above example, if the writer knows that the instrument was one from a specific vendor, but not the model, then the vendor model term should be used instead of the completely generic “instrument model”.

To obsolete a term, the following must be done:

* Put OBSOLETE at the beginning of the definition
* Add a comment to the term describing the reason for obsoleting.
* Set the OBO-format is\_obsolete tag to true

This is a summary of the procedure given in <http://psidev.info/files/CommunityPractice-revised.doc>.

If a term name needs to be changed, the accession number should stay the same and the term name simply changed. One should NOT obsolete the term and create a new one with the revised name.

The following ontologies or controlled vocabularies specified below may also be suitable or required in certain instances:

* Unit Ontology (http://www.obofoundry.org/cgi-bin/detail.cgi?id=unit)
* ChEBI (<http://www.ebi.ac.uk/chebi/>)
* OBI (Ontology of Biological Investigations - <http://obi.sourceforge.net/>) (formerly called FuGO)

## Resolved Design Issues

There were several issues regarding the design of the format that were not clear cut, and a design choice was made that was not completely agreeable to everyone. So that these issues do not keep coming up, we document here the issues and why the decision that is implemented was made.

### Count attributes

It was decided that all list elements would have a count attribute. The reason is that parsers implemented in languages where memory allocation or array sizing is important, it is a nice performance enhancement to have a count attribute indicating how many elements there are in the list. Although it was felt that this is an easy target for creating inconsistent files (i.e. writing out a count=”5” attribute followed by 6 items in the list), this was deemed to be rare and in the vast majority of cases the value can be relied on. The code would need to handle cases where the count was incorrect, but this is no more difficult than not knowing the value ahead of time. Validators of mzML should check that the counts are correct when validating file.

### Numerical value and datetimestamp encoding

All numerical values shall appear in the XML schema datatype specification (http://www.w3.org/TR/xmlschema-2/). The number 1/10 must always appear as 0.10 and never as 0,10. A preceeding + before a number (+5.0) is prohibited.

Datetimestamps must also be encoded as in the XML specification such as 2007-06-27T15:23:45.00035.

All id attributes follow the XML schema datatype xs:ID (http://www.w3.org/TR/xmlschema-2/#ID), which means that no two id attributes may be the same within a document, and id attributes must be purely alphanumerical strings with at least one letter. Thus they may not contain spaces or underscores, and id attributes may not be a plain number.

### The new CV term problem

It was decided that all annotations on the data that should come from a list of allowed values using the <cvParam> element. All information encoded as element attributes are never controlled vocabulary terms. Thus, as an example to describe spectrum type, the cvParam element must be provided to specify a term below MS:1000559 “spectrum type”, such as:

<cvParam cvLabel="MS" accession="MS:1000580" name="MSn spectrum" value=""/>

Note that both accession and the term name are provided. Parsers should focus on the accession number as this should never change, even if the term name is adjusted in the controlled vocabulary later. Note that there is no value. The mere presence of the term is the annotation.

The problem comes when there is a new term to be added. Let’s assume that it becomes necessary to add a new spectrum type “SRM spectrum”. Vendor X would like to start writing mzML with this spectrum type. What should happen and what could also happen?

In an ideal world, Vendor X would contact the PSI-MS WG controlled vocabulary coordinators list and request a new child term of “spectrum type”. A CV coordinator would verify that this is a new concept, not simply a synonym of an existing concept, add the term to the CV and release a new version of the psi-ms.obo file at the same location. Vendor X would obtain the accession number and could begin writing out valid mzML. The semantic validator would (and already does) automatically download the new .obo file and validate that the new mzML is semantically valid using the new term.

If the file is then distributed to arbitrary site Y, local software will suddenly encounter this new term:

<cvParam cvLabel="MS" accession="MS:1000583" name="SRM spectrum" value=""/>

and there is a possibility of failure. The first problem that may occur is that reader software may try to understand what the spectrum type is, but it will not find a spectrum type that it understands. Therefore it can only conclude that either no spectrum type was provided, or one of the terms it doesn’t recognize is a spectrum type but it won’t know which one. If the software could connect to the Internet, and could automatically download the latest .obo file, and look to see if this term was in the file, and then determine what parent the above term had, and understand that the parent is “spectrum type”, then the software could conclude that the above cv term is a spectrum type, but a new one that it doesn’t not know how to handle yet. Such a string of logic is not terribly difficult but it is not trivial and is objectionable to some.

Worse yet, Vendor X could have been lazy and not even contacted the PSI-WG CV coordinator and just started publishing mzML with (“option A”):

<cvParam cvLabel="MS" accession="MS:9999999" name="SRM spectrum" value=""/>

There is no way to resolve this. No reader could possibly know how to handle this. It should be avoided at all cost. Although it should be noted that until the proper accession number is furnished, such an approach will at least initially be used. In any case, processing software may still not know how to properly handle SRM spectra, but it should already be able to confidently understand what the spectrum type is and admit it cannot handle it.

In order to make it easier for software developers, several possible solutions were considered at length, notably one known as “option C”:

<cvParam cvLabel="MS" categoryAccession=”MS:1000035” categoryName=”spectrum type” accession="MS:1000583" name="SRM spectrum" value=""/>

In this case, both the accession and name of both the parent/category and the child/leaf term is explicitly provided. This could help the reader software to know exactly that the spectrum type is “SRM spectrum” without complex logic and proceed with processing based on that.

In this alternative, both the category and the leaf terms are explicitly provided. Reader software could easily determine what the spectrum type is without complex logic even in the face of a MS:9999999 accession shown above. The downside of option C is that files are more verbose than otherwise and that there is the possibility of conflicting, incorrect stated category and left nodes. A final argument against this is that such a scheme was tried with MAGE-ML (MicroArray Gene Expression Markup Language) version 1, and it caused massive confusion. An alternate contention is that the massive confusion resulted from nesting of terms rather than the flat attribute structure proposed here.

After much discussion, it was decided to stick with the original implementation (option A). There were too many arguments against option C, and the main author of the ProteoWizard reference implementation saw no need for option C. Thus, the final specification proceeds with option A.

## Other supporting materials

This document cannot be fully judged on its own. It is important to study the accompanying sample instance documents, controlled vocabulary, schema files, and the software that implements this pre-release version of nmrML. In fact, the content of section 3 in this document (as well as the on-line HTML documentation) is completely auto-generated from these files and not maintained by hand.

All these files and programs are available from nmrml.org. Further development versions with accompanying readme files can be found on the nmrML GitHub site ( <https://github.com/nmrML/nmrML> ).

1. **Source files and documentation**

The following describes the more important files and documents that we have prepared and their respective download locations:

**nmrML.xsd**

**(nmrml.org/schema/1.0.rc1/nmrML.xsd):** An XML schema that defines the structure, content and parts of the semantics of the allowed nmrML XML documents. The XML schema definition (XSD) uses XML Schema version 1.1 format following the W3C recommendation (w3.org/XML/Schema). The schema allows for the capture of raw NMR spectrum data and acquisition parameters for both one-dimensional and two-dimensional spectra, including two-dimensional J-resolved spectra.

**nmrCV.owl**

**(nmrml.org/cv/1.0.rc1/nmrCV.owl):** The controlled vocabulary (CV) describing the more variant terminology in an unambiguous and standardized way. This ontology is the MSI-sanctioned successor of artifacts developed previously at EMBL-EBI, Hinxton, UK (D. Schober, Sansone Group) and the Wishart Research Group, Edmonton, Canada (J. Cruz). This CV currently covers the description of NMR spectrum acquisition set up and raw data generated during the acquisition. There is less coverage of data generated by analysis of the spectrum such as metabolite quantification and identification. The CV terms are used within the nmrML xml file, at positions specified in the XSD, e.g. by CVParam references.

**XML example files**

**(https://github.com/nmrML/nmrML/tree/master/examples/working.tmp/nmrML & https://github.com/nmrML/nmrML/tree/master/examples):** Multiple XML instances complying with the XSD were generated to illustrate the usage of nmrML in a practical experiment data annotation. These instances also served to test the XSD and CV on coverage, structural soundness and to test parser software.

**XSDToCV mapping file**

**(nmrml.org/schema/1.0.rc1/nmrml-mapping.xml):** This xml file specifies rules to constrain CV term usage during data entry, i.e. it allows to verify validness of CV term usage in the nmrML XML files. This mapping file will also be used to enforce minimal metadata standards[[21]](#endnote-21). Only a very first draft has been created for testing purposes.

**HTML documentation files**

**(nmrml.org/schema/1.0.rc1/doc & nmrml.org/cv/1.0.rc1/doc):** Documentation was generated with automated tools that describes the nmrML XSD and the CV OWL. The documentation allows non-XML and non-ontology savvy end-users to open, browse and comment on the standards as well as facilitating the use of the data format by developers and the implementation of tools that use, read or write nmrML.

To further ease adoption we also created supplemental documentation and tutorials.

All source files are available on the project Github pages, together with an accompanying readme file.

**GitHub:** <https://github.com/nmrML/nmrML>

**Cosmos website:** [http://www.cosmos-fp7.eu](http://www.cosmos-fp7.eu/)

**nmrML website:** [http://nmrml.org](http://nmrml.org/)

**nmrML wiki:** <http://cosmos-fp7.eu/nmrML/>

**nmrML google forum:** [https://groups.google.com/forum/#!forum/nmrml](https://groups.google.com/forum/)

Or the above as table:

|  |  |
| --- | --- |
| Filename | Description |
| mzML1.1.0.xsd | Main mzML XML schema definition file |
| mzML1.1.0.html | HTML documentation of the model |
| mzML1.1.0\_idx.xsd | Wrapper schema for indexing an mzML file for random access |
| mzML1.1.0\_idx.html | HTML documentation of the index |
| psi-ms.obo | PSI-MS controlled vocabulary in OBO format |
| specialNotes.txt | A set of special notes associated with individual elements |
| ms-mapping.txt | XML-encoded rules for where certain cvParams MAY/MUST appear in the document. |
| tiny.pwiz.1.1.mzML | Tiny hand-crafted four spectrum MS1 + MS2 LCQ example |
| small.pwiz.1.1.xml | ProteoWizard generated example for generic MS/MS data |
| small\_miape.pwiz.1.1.xml | ProteoWizard generated MIAPE-compliant demonstration example for generic MS/MS data |
| MRM\_example\_1.1.0.mzML | Example of encoded SRM data |
| dta\_example.mzML | Example of conversion from generic dta files to mzML |
| neutral\_loss\_example\_1.1.0.mzML | Example of a neutral loss spectrum |
| validateMzML.pl | Small Perl program to use Xerces to perform crude (not semantic) validation of an mzML file |
| mzMLContentHandler.pm | Reader class used by validateMzML.pl |
| mzML\_1.1.0\_validator.zip | Java implementation of a full semantic validator for mzML |

1. **Next steps**

The next step is to plan the first official release of the core XSD and initial CV. Further testing of the XSD is required with diverse experimental configurations, to ensure that our goal of flexibility has been achieved. Continuing to improve the documentation and building a community of users will provide further feedback for improvements to the Schema. At the same time we will continue the data-driven CV expansions and add new terms according to the additional examples selected by our different partners. We must also ensure that the schema is compatible with the steps we are taking toward QuantML and IdentML. On the CV side we also need to integrate new EBI-NMR CV classes (using tabular mass term import).

In general we have to extend the format specification, e.g. adding more experimental metadata, such as sample types as well as more information on metabolite identification and quantification (both XSD and CV side).

Also we need to work out an evaluation pipeline. As part of the next deliverable (D2.5 - Real data, Converters, Validators and Parsers for NMR-ML, m24), we will implement the CV-aware validator software and extensive mapping files containing the verification rules to check XML instances on semantic errors and completeness.

In parallel we will implement the parsers for format conversions and I/O to open source tools. The creation of ISA Tab specifications for easy tabular data entry and minimal reporting requirement enforcement is considered a further next step (D2.6).

## Open Issues

All open issues should be resolved by the end of the nmrML 1.0 design phase. This may be an appropriate section to describe issues that arise after the 2.0 release that are considered sufficiently disruptive that they require a new major or minor release number. Such disruptive changes to the schema are expected to be rare. It is hoped that it will be several years before the next major revision.

We note that usage of resource description framework (RDF) was considered and rejected for nmrML. It remains a possibility that a later version will be based on RDF completely.

## Comments on Specific Use Cases

Many special use cases for mzML were considered during its development. Most of these use cases have a corresponding example file that exercise the relevant part of the schema and provide a reference implementation example. Authors of mzML writing software are encouraged to examine the examples that accompany this format release before implementing the writer. Further, such authors are encouraged to use the validator before releasing any new writer code and working with the MSI NMR Working Group to resolve and issues. In the subsections below, we comment on a few of the notable use cases that were considered.

### Profile (continuous) spectra vs. centroided (peak list) (peak picked) spectra

…

## Other implementation guidelines

* For semantic validation, only SubclassOf (IS\_A) relationships should be considered as child terms
* The official namespace for nmrML is <http://nmrml.org/schema> . The reference URL for the current release candidate version of the schema is <http://nmrml.org/schema/v1.0.rc1/nmrML.xsd> .

## How to contribute and provide feedback

Change and expansion requests on the nmrML.xsd and nmrCV should be send to the COSMOS nmrML mailing list at <https://groups.google.com/forum/?hl=en#!forum/nmrml/join>

Alternatively use the Git Issue Tracker: Use „CV:“-prefix on issues subject line to render issues easy findable. Others could be „XSD:“ (the default), „DP“ for design principle, „FC:“ for format and conversions, … ?

Feedback on the documentation can be send to

Daniel Schober (Cosmos WP2 Lead on nmrML) at [dschober@ipb-halle.de](mailto:dschober@ipb-halle.de)

Concrete questions on the XSD should be sent to

Michael Wilson at XYZ

# Model in XML Schema

The nmrML XSD is described below.

## Element <nmrML>

|  |  |
| --- | --- |
| **Definition:** | This is the root element for the Proteomics Standards Initiative (PSI) mzML schema, which is intended to capture the use of a mass spectrometer, the data generated, and the initial processing of that data (to the level of the peak list). |
| **Type:** | dx:mzMLType |
| **Attributes:** | |  |  |  |  | | --- | --- | --- | --- | | **Attribute Name** | **Data Type** | **Use** | **Definition** | | accession | xs:string | optional | An optional accession number for the mzML document used for storage, e.g. in PRIDE. | | id | xs:string | optional | An optional id for the mzML document used for referencing from external files. It is recommended to use LSIDs when possible. | | version | xs:string | required | The version of this mzML document. | |
| **Subelements:** | |  |  |  |  | | --- | --- | --- | --- | | **Subelement Name** | **min** | **max** | **Definition** | | [cvList](http://www.peptideatlas.org/tmp/mzML1.1.0_plain.html#cvList) | 1 | 1 | Container for one or more controlled vocabulary definitions. | | [fileDescription](http://www.peptideatlas.org/tmp/mzML1.1.0_plain.html#fileDescription) | 1 | 1 | Information pertaining to the entire mzML file (i.e. not specific to any part of the data set) is stored here. | | [referenceableParamGroupList](http://www.peptideatlas.org/tmp/mzML1.1.0_plain.html#referenceableParamGroupList) | 0 | 1 | Container for a list of referenceableParamGroups | | [sampleList](http://www.peptideatlas.org/tmp/mzML1.1.0_plain.html#sampleList) | 0 | 1 | List and descriptions of samples. | | [softwareList](http://www.peptideatlas.org/tmp/mzML1.1.0_plain.html#softwareList) | 1 | 1 | List and descriptions of software used to acquire and/or process the data in this mzML file. | | [scanSettingsList](http://www.peptideatlas.org/tmp/mzML1.1.0_plain.html#scanSettingsList) | 0 | 1 | List with the descriptions of the acquisition settings applied prior to the start of data acquisition. | | [instrumentConfigurationList](http://www.peptideatlas.org/tmp/mzML1.1.0_plain.html#instrumentConfigurationList) | 1 | 1 | List and descriptions of instrument configurations. At least one instrument configuration MUST be specified, even if it is only to specify that the instrument is unknown. In that case, the "instrument model" term is used to indicate the unknown instrument in the instrumentConfiguration. | | [dataProcessingList](http://www.peptideatlas.org/tmp/mzML1.1.0_plain.html#dataProcessingList) | 1 | 1 | List and descriptions of data processing applied to this data. | | [run](http://www.peptideatlas.org/tmp/mzML1.1.0_plain.html#run) | 1 | 1 | A run in mzML should correspond to a single, consecutive and coherent set of scans on an instrument. | |
| **Graphical Context:** | mzML_figure_001_mzML |
| **Example Context:** | <mzML xmlns="http://psi.hupo.org/ms/mzml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://psi.hupo.org/ms/mzml http://psidev.info/files/ms/mzML/xsd/mzML1.1.0.xsd" id="urn:lsid:psidev.info:mzML.instanceDocuments.tiny.pwiz" version="1.0">  <cvList count="2">  <cv id="MS" fullName="Proteomics Standards Initiative Mass Spectrometry Ontology" version="1.18.2" URI="http://psidev.cvs.sourceforge.net/\*checkout\*/psidev/psi/psi-ms/mzML/controlledVocabulary/psi-ms.obo"/>  <cv id="UO" fullName="Unit Ontology" version="04:03:2009" URI="http://obo.cvs.sourceforge.net/\*checkout\*/obo/obo/ontology/phenotype/unit.obo"/>  </cvList>  <fileDescription>  <fileContent>  ...  </mzML> |
| **Notes and Constraints:** | The <mzML> element and all content below may occur by itself in an XML document, but is also designed to be wrapped in the mzML indexing schema in order to facilitate random access within the file with appropriate reader software. |

…

# Conclusions

This document contains the specifications for using the nmrML format to represent NMR spectrometry results, metadata and associated context. This specification, in conjunction with the XML Schema and the Reference Manual constitute a proposal for a standard from the Metabolomics Standards Initiative coordinated by COSMOS.

# Authors and Contributors

Daniel Schober

Mike Wilson

Steffen Neumann

...

The following people contributed to the model development, controlled vocabulary development, gave feedback or tested nmrML:



# References

Bradner, S. (1997). "Key words for use in RFCs to Indicate Requirement Levels, Internet Engineering Task Force, RFC 2119, <http://www.ietf.org/rfc/rfc2119.txt.>"

Jones, A. R., M. Miller, et al. (2007). "The Functional Genomics Experiment model (FuGE): an extensible framework for standards in functional genomics." Nat Biotechnol **25**(10): 1127-1133.

Julian, R. K., P.-A. Binz, et al. (2005). "<http://psidev.info/index.php?q=node/80#mzdata.>"

Pedrioli, P. G., J. K. Eng, et al. (2004). "A common open representation of mass spectrometry data and its application to proteomics research." Nat Biotechnol **22**(11): 1459-66.

# Intellectual Property Statement

COSMOS and MSI take no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the MSI/PSI Chair.

COSMOS and MSI invite any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the main author.

# Appendix A:

# Copyright Notice

Copyright (C) Cosmos and the Metabolomics Standards Initiative (2013). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the PSI or other organizations, except as needed for the purpose of developing Proteomics Recommendations in which case the procedures for copyrights defined in the PSI Document process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the PSI or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE PROTEOMICS STANDARDS INITIATIVE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."

1. Wolfram Gronwald, Matthias Klein and Peter Oefner (submitted ?), MetaboQuant: A Tool Combining Individual Peak Calibration and Outlier Detection for Accurate Quantification from NMR Spectra [↑](#endnote-ref-1)
2. P Lampen, J Lambert, RJ Lancashire et al., AN EXTENSION TO THE JCAMP-DX STANDARD FILE FORMAT, JCAMP-DX V.5.01, Pure Appl. Chem., Vol. 71, No. 8, pp. 1549-1556, 1999 [↑](#endnote-ref-2)
3. Sansone,S.A., Fan,T., Goodacre,R. et al. (2007) The metabolomics standards initiative. Nat. Biotechnol., 25, 846–848. [↑](#endnote-ref-3)
4. Sansone SA, Schober D, Atherton HJ, Fiehn O, Jenkins H, Rocca-Serra P, Rubtsov DV, Spasic I, Soldatova L, Taylor C, Tseng A, Viant MR (2007) Metabolomics standards initiative: ontology working group work in progress. Metabolomics 3, 249-256. ISSN 1573-3882 [↑](#endnote-ref-4)
5. <http://www.psidev.info/> [↑](#endnote-ref-5)
6. Martens,L., Chambers,M., Sturm,M. et al. (2011) mzML—a community standard for mass spectrometry data. Mol. Cell Proteomics, 10, R110000133. http://www.ncbi.nlm.nih.gov/pubmed/20716697 [↑](#endnote-ref-6)
7. Sansone,S.A., Fan,T., Goodacre,R. et al. (2007) The metabolomics standards initiative. Nat. Biotechnol., 25, 846–848. [↑](#endnote-ref-7)
8. Sansone SA, Schober D, Atherton HJ, Fiehn O, Jenkins H, Rocca-Serra P, Rubtsov DV, Spasic I, Soldatova L, Taylor C, Tseng A, Viant MR (2007) Metabolomics standards initiative: ontology working group work in progress. Metabolomics 3, 249-256. ISSN 1573-3882 [↑](#endnote-ref-8)
9. <http://www.metabolomicscentre.ca/exchangeformats.htm> [↑](#endnote-ref-9)
10. Taylor CF, Field D, Sansone SA, et al., Promoting coherent minimum reporting guidelines for biological and biomedical investigations: the MIBBI project, Nat Biotechnol. 2008 Aug;26(8):889-96. doi: 10.1038/nbt.1411. , PMID:18688244 [↑](#endnote-ref-10)
11. Montecchi-Palazzi L., Kerrien S., Reisinger F. et al. (2009) The PSI semantic validator: a framework to check MIAPE compliance of proteomics data. Proteomics, 9, 5112–5119. [↑](#endnote-ref-11)
12. <http://www.metabolomicscentre.ca/nmrML/biosample-concentrations.xml> [↑](#endnote-ref-12)
13. Haug, K., Salek, R. M., Conesa, P., Hastings, J., de Matos, P., Rijnbeek, M., et al. (2013). MetaboLights--an open-access general-purpose repository for metabolomics studies and associated meta-data. [Research Support, Non-U.S. Gov't]. *Nucleic acids research, 41*(Database issue), D781-786, doi:10.1093/nar/gks1004. [↑](#endnote-ref-13)
14. Wishart, D. S., Jewison, T., Guo, A. C., Wilson, M., Knox, C., Liu, Y., et al. (2013). HMDB 3.0--The Human Metabolome Database in 2013. [Research Support, Non-U.S. Gov't]. *Nucleic acids research, 41*(Database issue), D801-807, doi:10.1093/nar/gks1065.

    Rubtsov DV, Jenkins H, Ludwig C, Easton J, Viant MR, Günther U, Griffin JL, Hardy N (2007) Proposed reporting requirements for the description of NMR-based metabolomics experiments. Metabolomics 3, 223–229. [↑](#endnote-ref-14)
15. Farag, M., Porzel, A., Schmidt, J. & Wessjohann, L. Metabolite profiling and fingerprinting of commercial cultivars of Humulus lupulus L. (hop) - a comparision of MS and NMR methods in metabolomics Metabolomics 8, 492-507, (2012) [↑](#endnote-ref-15)
16. Schober D., Mayer G., Moing A., Eisenacher M., Neumann S., Ontological analysis of controlled vocabularies used in PSI/MSI supported XML standards, Workshop: ODLS 2013, GI-Edition Lecture Notes in Informatics, Proceedings of the Jahrestagung der Gesellschaft für Informatik 2013, Matthias Horbach (Hrsg.), Koblenz, Germany, 16.–20. September 2013, p. 1875-1888, https://wiki.imise.uni-leipzig.de/Gruppen/OBML/Workshops/2013-ODLS-en [↑](#endnote-ref-16)
17. <http://www.w3.org/TR/owl2-syntax/> [↑](#endnote-ref-17)
18. <http://www.geneontology.org/GO.format.obo-1_2.shtml> [↑](#endnote-ref-18)
19. <http://www.imbi.uni-freiburg.de/ontology/biotop/> [↑](#endnote-ref-19)
20. <http://www.ncbi.nlm.nih.gov/pubmed/23046606> [↑](#endnote-ref-20)
21. Montecchi-Palazzi L., Kerrien S., Reisinger F. et al. (2009) The PSI semantic validator: a framework to check MIAPE compliance of proteomics data. Proteomics, 9, 5112–5119. [↑](#endnote-ref-21)