XLUM-file Format Documentation

Sebastian Kreutzer, Steve Grehl, Michael Höhne

version: 1.0 [2022-06-30]

Contents

1 Motivation and principles **2**

2 General description **2**

3 Detailed description **3**

3.1 The `<curve/>` level . . . 3
3.2 The `<record/>` level . . . 6
3.3 The `<sequence/>` level . . . 8
3.4 The `<sample/>` level . . . 9
3.5 The `<xlum/>` level . . . 10
3.6 General parameters (all nodes) . . . 12

4 Special cases **12**

4.1 Pulsing data (optional) . . . 12

5 Pratical examples **13**

1 Motivation and principles

This document details the `XLUM` format, an XML based file format for long-term data preservation and exchange of luminescence data. The format is readable by humans and machines, and the data can be easily checked with any text reader on any major operating system. This design allows a platform-independent operation. The following documentation provides essential information on the file `XLUM` format and accounts for the needs for individual flexible data analysis, even with self-written software applications.

Three simple design principles underpin the format specification:

- Stored are data recorded over **time** or time derived instances,
- stored are data on a technical component level,
- data stored in a file are self-consistent.

2 General description

The `XLUM`-format is an XML derivative. The base data structure is a tree with five nodes types storing the data. Each level has its unique denotation.

The levels and a short introduction is given in the table below. A detailed description can be found in the following sections. Nodes represent each level in the structure. **Only `<curve>` nodes contain physical quantities, such as luminescence data** all other nodes are parent nodes of `<curve>` to structure the dataset and ship additional metadata.

The first level, `xlum` is a root node to enable storage of luminescence data in arbitrary files following the XML scheme. As a side effect this also allows custom file endings different from `*.xlum`.

Table 1: Node overview

Node name	Number of samples	Number of aliquots	Number of records	Number of data curves
`<xlum/>`	inf	inf	inf	inf
`<sample/>`	1	inf	inf	inf
`<sequence/>`	1	1	inf	inf
`<record/>`	1	1	1	inf
`<curve/>`	1	1	1	1

A node has a name, attributes, and data stored in it. The data stored in the attributes describe the state of this level. The stored data in the node describes the process(es) assigned to the node. A minimal example is shown in the listing. The documentation provides an overview starting with the leaf node description and going up to the root from there. Further notes:

- The format version bases on XML version 1.0 (Fith Edition)
- File encoding should always be UTF-8
- This specification lists only mandatory attributes. Additional, custom attributes are explicitly supported.
- Parser supporting the `XLUM` format must not crash when encountering non-specified node attributes. However, they may skip them.

```
<?xml version="1.0" encoding="utf-8"?>
<xlum>
  <sample>
    <sequence>
      <record>
        <curve>
          1 2 ... n
        </curve>
```

```
      </record>
    </sequence>
  </sample>
</xlum>
```

3 Detailed description

3.1 The <curve/> level

The <curve/> level is the deepest node (leaf) and has no further sub-levels. A curve holds the predefined/simulated or measured output of one single technical component. For example, a typical thermoluminescence measurement may consist of one or many curves.

1. **The three curve example**: (a) Time against temperature recorded by a thermocouple (temperature sensor), (b) time against photon counts recorded by a photomultiplier tube, (c) time against a predefined heating ramp.
2. **The one curve example**: Time against temperature. In this case temperature is a processed quantity, because measurements happen over a time instant. However, for compatibility reasons, this would be allowed although it is not preferred.

In both cases, all mentioned <curve/> nodes belong to one parent <record/>. In case 1, the record contains one curve and three in case 2. Ideally, curves represent technical components and not processed quantities.

3.1.1 Value storage

Numerical values in the <curve/> node are stored white space separated. **Only quantities detected by a detector/sensor are stored in this node.** This includes simulated values. Figure 1 shows three types of (luminescence) detectors, distinguished by their capacity to record spatially resolved information. Regardless of the detector, recorded quantities are stored in the <curve/> node separated by white space. Example:

```
<curve>59.5167 109.5164 149.7003 109.5170 156.2654</curve>
```

Values in the node are real numbers (-1e-307 $\leqslant x \leqslant$ 1e+307; $x \in \mathbb{R}$). Scientific 'E'/'e' notations of numbers are explicitly allowed, example: 1e+2 for 100 or 1e-2 for 0.01. Decimal separator is a . (dot). **Other separators, for instance to separate groups of numbers are not allowed** to avoid decimal delimiter problems. Examples:

- 10000.00 -> OK
- 10 000.00 -> NOT OK
- 10,000.00 -> NOT OK

For consistency reasons, the value in the node span an array with dimensions defined through the node attributes xValues, yValues and tValues (see Fig. 1).

1. The simplest form is a detector that records no spatial information. For instance, a photomultiplier tube only knows about count recorded over time. Alternatively, the detector can be any sensor, measuring, e.g., temperature, pressure or current.
2. A spectrometer (here a camera in front of a spectrograph) is another type of detector with spatial information, here wavelengths split by the spectrograph hitting different pixel areas of the detector.
3. The most complex from is a detector that records events (e.g., luminescence), spatially resolved. Such as a camera.

In the examples and the Fig.1 the detectors detect luminescence, however, any type of sensor, electronic component recording information of relevance for the measurement of luminescence are suitable. Typical examples:

- Temperature sensor (thermocouple), recording the temperature of the heating element

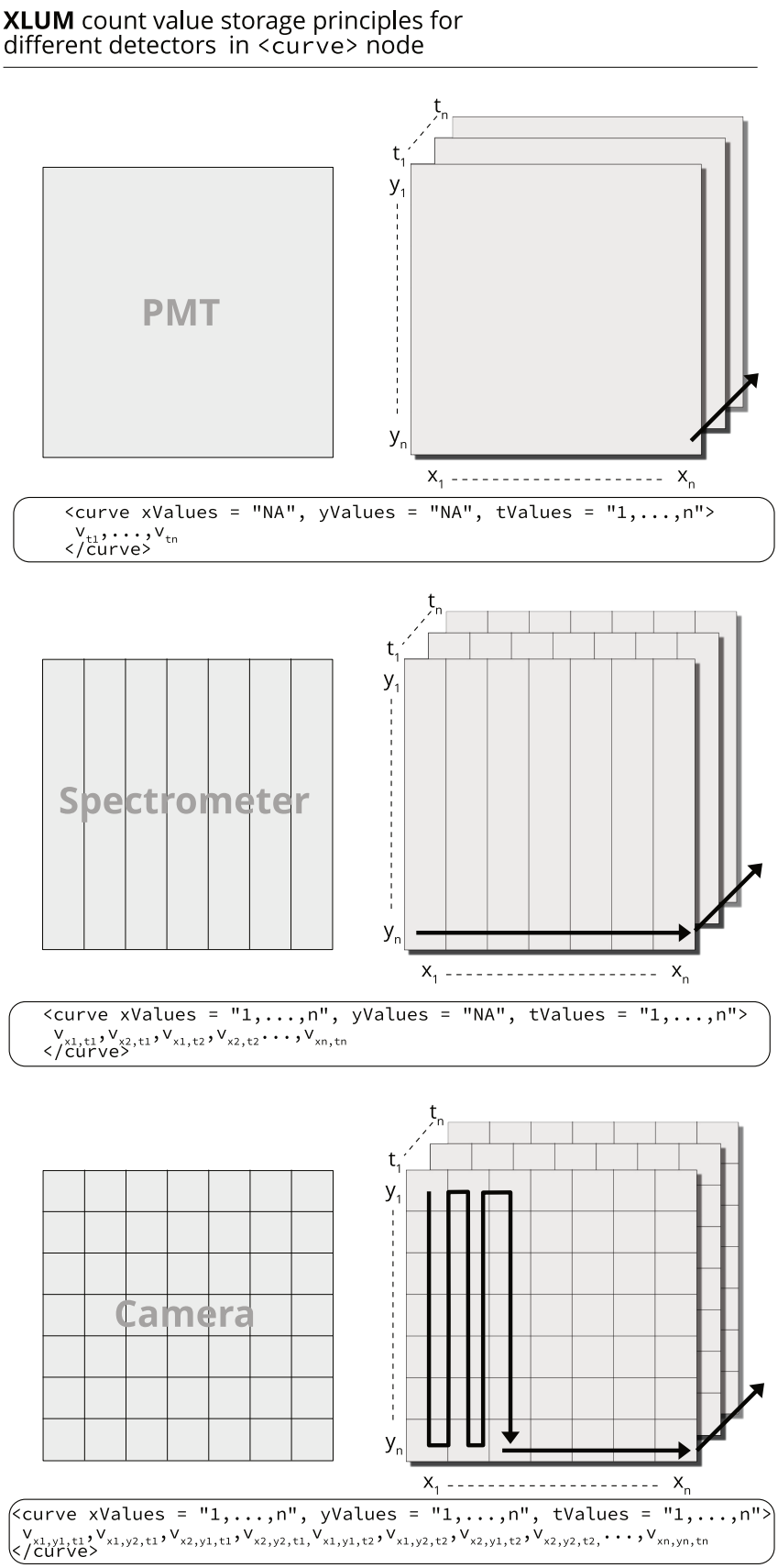

Figure 1: The three cases for data storage depending on the detector. The black solid arrows indicate how the array is constructed from the values stored in the node.

- Photo diode, monitoring the optical power density at the sample position
- Resistor, measuring the current of a LED
- Pressure sensor, monitoring the atmosphere in the measurement chamber
- Inductive sensor, monitoring closing and opening of a shutter

3.1.2 Node attributes

Table 2: Specified curve attributes. Attributes in `[]` are optional, hence `NA` is not allowed.

Identifier	Type	Allows `NA`?	Example	Information
component	`string`	yes	`"EMI 9235QB15"`	name of the technical component. `NA` allowed only for `"predefined"`
startDate	`dateTime`	no	`"2021-07-14T22:59:35.0Z"`	ISO 8601-1:2019: `YYYY-MM-DDThh:mm:ss[.mmm]Z` recalculated to Zulu time
curveType	`string`	no	`"predefined"`	Values allowed are only `"predefined"` or `"measured"`
duration	`number`	no	`"20.000"`	Duration of the detection in seconds or a fraction of it
offset	`number`	no	`"10.000"`	Before the detection starts in seconds or a fraction of it
xValues	`list unsignedInt`	no	`"1 2 3"`	x-coordinate values of the detector; `0` indicates that that this parameter is not used
yValues	`list unsignedInt`	no	`"1 2 3"`	y-coordinate values of detector; `0` indicates that that this parameter is not used
tValues	`list double`	no	`"1. 2.0 3.0"`	time values in seconds or fractions of it. Values must be positive
vLabel	`string`	no	`"Luminescence"`	Measured physical quantity
xLabel	`string`	yes	`"pixel"`	label of the x-coordinate values
yLabel	`string`	yes	`"pixel"`	label of the y-coordinate values
tLabel	`string`	no	`"time"`	label of t-values, usually 'time'
vUnit	`string`	no	`"cts"`	label of the v-values, for luminescence usually photon counts
xUnit	`string`	yes	`"nm"`	SI unit or equivalent for x-values
yUnit	`string`	yes	`"px"`	SI unit or equivalent for y-values

Identifier	Type	Allows NA?	Example	Information
tUnit	string	no	"s"	SI unit or equivalent for t-values
detectionWindow	string	yes	"375"	Centre wavelength if applicable. can be set to NA
filter	list (string)	yes	"Hoya U340; Delta BP 365/50EX"	Filter names separated by ;

3.1.3 Example

In order to define a suitable standard only a few attributes are required. A few of them are useful for every kind of component, others are only meaningfully in combination. For instance, providing information on the detection window is not meaningful for a heating element.

```
{...}
  <curve component="heating element" startDate="2021-02-14T22:57:00.0Z"
    curveType="predefined" duaration="5"
    offset="0" xValues="NA" yValues="NA"
    tValues="1 2 3 4 5" xLabel="NA" yLabel="NA"
    tLabel="time" vLabel="temperature" xUnit="NA" yUnit="NA" vUnit="K" tUnit="s">
        293 303 313 323 333
  </curve>
  <curve component="thermo couple" startDate="2021-02-14T22:57:00.0Z"
    curveType="measured" duaration="5"
    offset="0" xValues="NA" yValues="NA" tValues="1 2 3 4 5" xLabel="NA" yLabel="NA"
    vLabel="temperature" tLabel="time" xUnit="NA" yUnit="NA" vUnit="K" tUnit="s">
        293 303 313 323 333
  </curve>
  <curve component="EMI 9123QB15" startDate="2021-02-14T22:57:00.0Z"
    curveType="measured" duaration="5" offset="0" xValues="NA" yValues="NA"
    tValues="1 2 3 4 5" xLabel="NA" yLabel="NA"
    vLabel="luminescence" tLabel="time" xUnit="NA" yUnit="NA" vUnit="cts"
    tUnit="s" detectionWindow="380 nm" filter="Hoya U340 + Delta 365/40">
        20 25 32 41 46
  </curve>
{...}
```

One curve is related to one technical device. To define a accurate time window, the parameters *duration* and *offset* should be used. These parameters are related to the start of the parent <record/> node.

3.2 The <record/> level

A <record> defines one process involving many components, hence <record> may contain $1 \leq n \leq Inf; n \in \mathbb{Z}$ <curve> nodes. A record can also be understood as one step in a measurement sequence, e.g., a TL measurement. All curves within the record should have been detected within the same time frame defined through the attributes <startDate> and the duration of the step.

3.2.1 Attributes

Table 3: Allowed attributes for <record> node.

Identifier	Type	Allows NA?	Example	Information
recordType	string	no	"TL"	valid values according the recordType table
sequenceStepNumber	unsignedInt	yes	5	index in the sequence
sampleCondition	string	yes	dose	valid values according to the table below

The recordType names are not carved in stone, but should be chosen to best describe a record in one word or using an accepted abbreviation (e.g., OSL for optically stimulated luminescence). However, it is important to keep in mind that the recordType makes **no** claim on the technical components involved, but is solely set to ease the understanding of the record. The hard physical information a stored in the <curve> node.

Table 4: Valid recordTypes

recordType	Information
"bleaching"	any kind of bleaching
"irradiation"	irradiation with ionising source
"atmosphereExchange"	atmosphere exchange
"heating"	any kind for heating without signal detection
"spectrometer"	spectrometer
"camera"	camera, e.g. any kind of imaging other than spectrometry
"TL"	thermoluminescence measurement
"ITL"	Isothermal luminescence
"IRSL"	Infrared stimulated luminescence
"TM-OSL"	thermally modulated luminescence
"RF"	radiofluorescence
"UV-RF"	radiofluorescence with detection in the UV wavelength range
"IR-RF"	radiofluorescence with detection in the infrared wavelength range
"IR-PL"	infrared photoluminescence
"OSL"	optically stimulated luminescence (wavelength unspecified)
"BSL"	blue optically stimulated luminescence
"GSL"	green optically stimulated luminescence
"VSL"	violet optically stimulated luminescence
"YSL"	yellow optically stimulated luminescence
"POSL"	pulsed optical stimulated luminescence
"pause"	pause
"custom"	any undefined or custom record type not listed above

valid entries for sampleConditions are:

Table 5: Valid values for sampleConditions

sampleCondition	Information
"Natural"	Natural luminescence signal
"Natural+Dose"	Natural luminescence signal with additive dose

sampleCondition	Information
"Bleach"	Artificially depleted luminescence
"Bleach+Dose"	Artificially depleted luminescence with an additive dose
"Nat.(Bleach)"	Naturally depleted luminescence
"Nat.+Dose(Bleach)"	Naturally depleted luminescence with dose
"Dose"	Artificially irradiated sample
"Background"	Background

Please note that the attribute `sampleCondition` *is of informative nature, accounting for information stored in BIN/BINX-files. The field can be set to* `NA`*.*

3.2.2 Example

The following examples consists of two records.

```
{...}
    <record recordType="GSL" sequenceStepNumber="1" sampleCondition="NA">
      <curve ...>
        {...}
      </curve>
      <curve ...>
        {...}
      </curve>
    </record>
    <record recordType="bleaching" sequenceStepNumber="2" sampleCondition="NA">
      <curve ...>
        {...}
      </curve>
    </record>
{...}
```

3.3 The <sequence/> level

A sequence describes multiple measurements sequentially used at **one** aliquot, for instance as cup/disc with multiple grains or a single grain. Hence, another aliquot from the **same** sample and the sample sequence needs to be wrapped in a new `<sequence/>` node. Naturally the information stored an that level can be very verbose, but are limited to
a few required attributes only. In the luminescence context a sequence can fit, e.g., a SAR protocol, but also a succession of protocols measured on one aliquot.

3.3.1 Attributes

Table 6: Mandatory attributes <sequence> node

Identifier	Type	Allows NA?	Example	Information
position	numeric	no	"42"	position of aliquot in the reader, can be 0 in case samples are changed manually, i.e. the equipment has only one single measurement position
name	string	yes	"SAR measurement"	name of the sequence
fileName	string	yes	"SAR_OSL.seq"	sequence file so far applicable
software	string	yes	"Sequence Editor 4.4"	software used to write the sequence if applicable
readerName	string	yes	"Risø OSL/TL"	name of the luminescence reader employed for the measurement
readerSN	string	yes	"234-23"	reader serial number
readerFW	string	yes	"reader OSL 12.3"	reader firmware version

3.3.2 Example

```
{...}
<sequence position="1" name="SAR measurment"
  fileName="Testsequence.seq" software="DeviceStudio, 2.37">
  <record ...>
    <curve ...>
      {...}
    </curve>
    {...}
  </record>
  {...}
</sequence>
{...}
```

3.4 The <sample/> level

The sample constitutes the top level for one particular sample. The sample level may contain an infinite number of sequences measured for one sample.

3.4.1 Attributes

Table 7: Mandatory attributes <sample> node

Identifier	Data type	Allows NA?	Example	Information
name	number	yes	"LUM-21321"	unique sample identifier

Identifier	Data type	Allows NA?	Example	Information
mineral	`string`	yes	`"quartz"`	dominant mineral composition of the sample
latitude	`numeric`	yes	`"52.4091392"`	latitude in decimal degrees of the sample origin
longitude	`numeric`	yes	`"-4.0702446"`	longitude in decimal degrees of the sample origin
altitude	`numeric`	yes	`"50"`	altitude in m above see level
doi	`anyURI`	yes	`"10.1016/j.radmeas.2017.02.003"`	valid DOI with further information on the sample, if applicable

3.4.2 Example

```
{...}
  <sample name="LUM-21321" mineral="quartz" latitude="52.4091392" longitude="-4.0702446"
   altitude="50" doi="10.1016/j.radmeas.2017.02.003">
    <sequence ...>
      <record ...>
       <curve ...>
        {...}
       </curve>
       {...}
      </record>
      {...}
    </sequence>
    {...}
  </sample>
{...}
```

3.5 The <xlum/> level

The <xlum> is the format parent node and it can contain an infinite number of samples. Only this level allows to set a license for the distribution and usage of the data. All data within a particular <xlum> are considered distributed under the license. You have to create new instances of the <xlum> to set different licenses.

3.5.1 Attributes

Table 8: Mandatory attributes of <xlum> node

Identifier	Type	Allows NA?	Example	Information
xmlns	`namespace`	no	*predefined*	Format specific namespace
lang	`string`	no	*predefined*	ISO 639-1 language code, must not be changed

Identifier	Type	Allows NA?	Example	Information
version	numeric	no	"1.0"	the xlum format version number relevant
flavour	string	no	"generic"	allows to specify a particular flavour of the format, for instance for particular equipment
author	list(string)	yes	"Max Planck; Marie Curie"	names of the author(s) of this dataset
license	string	yes	"CC BY 4.0"	license for the distribution of the dataset.
doi	anyURI	yes	"10.5281/zenodo.596252"	A digital document identifier referencing the dataset already archive in a data repository

Table 9: Allowed license types

license	License name	Reference	Comment
"CC BY"	Creative commons	https://creativecommons.org	all flavours of the creative commons license scheme (e.g., CC BY-SA, CC BY-NC)
"CC0"	Creative Commons Public Domain Dedication	https://creativecommons.org	you give up the copyright on your data and make it public domain. Please note this cannot be revoked!
"Copyright"	Copyright protected	-	Data are copyright protected and cannot be used or distributed without the creator's agreement

3.5.2 Example

```
{...}
<xlum xmlns:xlum="xlum.org" lang="en"
  formatVersion="1.0" flavour="generic" author="Marie Curie; Max Planck"
  license="CC BY 4.0" doi="NA">
  <sample ...>
    <sequence ...>
      <record ...>
```

```
        <curve ...>
         {...}
        </curve>
        {...}
      </record>
    </sequence>
    {...}
  </sample>
  {...}
</xlum>
{...}
```

3.6 General parameters (all nodes)

A few attributes are available on all node levels except the root node `<xlum>`. There are mainly of technical nature to ease the sequential of data storage during the measurement process.

Table 10: Attributes valid in all nodes

Identifier	Data type	Allows `NA`?	Example	Information
state	`string`	yes	`"recording"`	defines the state of the node; only used by equipment manufactures
parentID	`token`	yes	`"201007145551910"`	unique identifying the parent node, the root node has the ID `0`, this is usually not needed but may help to store data
comment	`String`	yes	`"what a wonderful sample"`	comment field

4 Special cases

This section provides suggestions for attributes for particular types of measurements. These attributes are not compulsory, but may be used for the sake of consistency.

Either way, these metadata must not be required to store and analyse the data!

4.1 Pulsing data (optional)

In measurements with pulsed light stimulation data processing usually requires summations of photon counts on the hardware level, because it is neither meaningful nor feasible to store every single count in separate curves for photon arrival times of $10^{-6} \leq \tau \leq 10^{-3}$ s. Hence, multiple detector `<curve/>` data may be stored per `<record/>`, one for each (x) pulse(-s). The `<record/>` node contains some additional metadata that may help to understand the data and the undertaken processing on the hardware level better. All attributes are optional.

Table 11: Additional metadata `<record/>` node pulsing data

Identifier	Data type	Example	Information
onTime	`numeric`	`"1E-01"`	on time of stimulation per pulse, [s]
offTime	`numeric`	`"1E-01"`	on time of stimulation per pulse, [s]
nPulses	`numeric`	`"10"`	number of stimulation pulses total
summations	`numeric`	`"2"`	how many consecutive pulse records are summed (e.g. 10 pulses, 2 summations -> 5 records)
channelsPerPulse	`numeric`	`"1000"`	number of channels that are recorded in one curve
countsNormalised	`numeric`	`"1"`	if not 0, counts per channel are normalized to counts per second (helpful with uneven distributed channel times)

Also the curves contain additional metadata

Table 12: Additional metadata `<curve>` node pulsing data

Identifier	Data type	Example	Information
pulseID	`numeric`	`"1"`	current pulse number

5 Pratical examples

This section contains detailed examples for particular curve types in order to provide usable examples for the data format.

```
<?xml version="1.0" encoding="utf-8"?>
<xlum lang="en" version="1.0" flavour="xlum" doiRef="">
 <sample ...>
  <sequence ...>
   <record ...>
    <curve ...>
     12 23 23 23 13 23
    </curve>
   </record>
  </sequence>
 </sample>
</xlum>
```