XLUM-file format Documentation

Sebastian Kreutzer, Steve Grehl, Michael Höhne,... <further authors>

version: 1.0 [2021-07-09]

Contents

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.

Two 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.

2 General description

The `XLUM`-format is an XML derivative. The base data structure is a tree with five nodes 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 luminescence data** all other nodes are parent nodes of `<curve>` to structure the dataset and ship additional metadata.

The first level, `xlum` is a supernode 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
- Attributes and
- 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 record only one or more 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

Values in the `<curve>` node are stored comma separated. **Only luminescence is stored in the nodes.** Fig. 1

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
startDate	`date`	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 measurement in seconds or a fraction of it
offset	`number`	no	`"10.000"`	Before the detection starts in seconds or a fraction of it
xValues	`number`	yes	`"1.0,2.0,3.0"`	x-coordinate values of the detector
yValues	`number`	yes	`"1.0,2.0,3.0"`	y-coordinate values of detector
tValues	`number`	no	`"1.0,2.0,3.0"`	time values in seconds or a fraction of it
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'
xUnit	`string`	yes	`"nm"`	SI unit or equivalent for x-values
yUnit	`string`	yes	`"px"`	SI unit or equivalent for y-values
tUnit	`string`	no	`"s"`	SI unit or equivalent for t-values
stimulator	`string`	yes	`"SiN heater"`	name of the technical component. `NA` allowed only for `"predefined"`
[detector]	`string`	no	`"EMI 9235QB15"`	technical name of the component can be `NA` for `"predefined"`
[detectionWindow]	`string`	no	`"375"`	Centre wavelength if applicable. can be set to `NA`
[filter]	`list` (`string`)	no	`"Hoya U340; Delta BP 365/50EX"`	Filter names separated by `;`

XLUM count value storage principles for different detectors in <curve> node

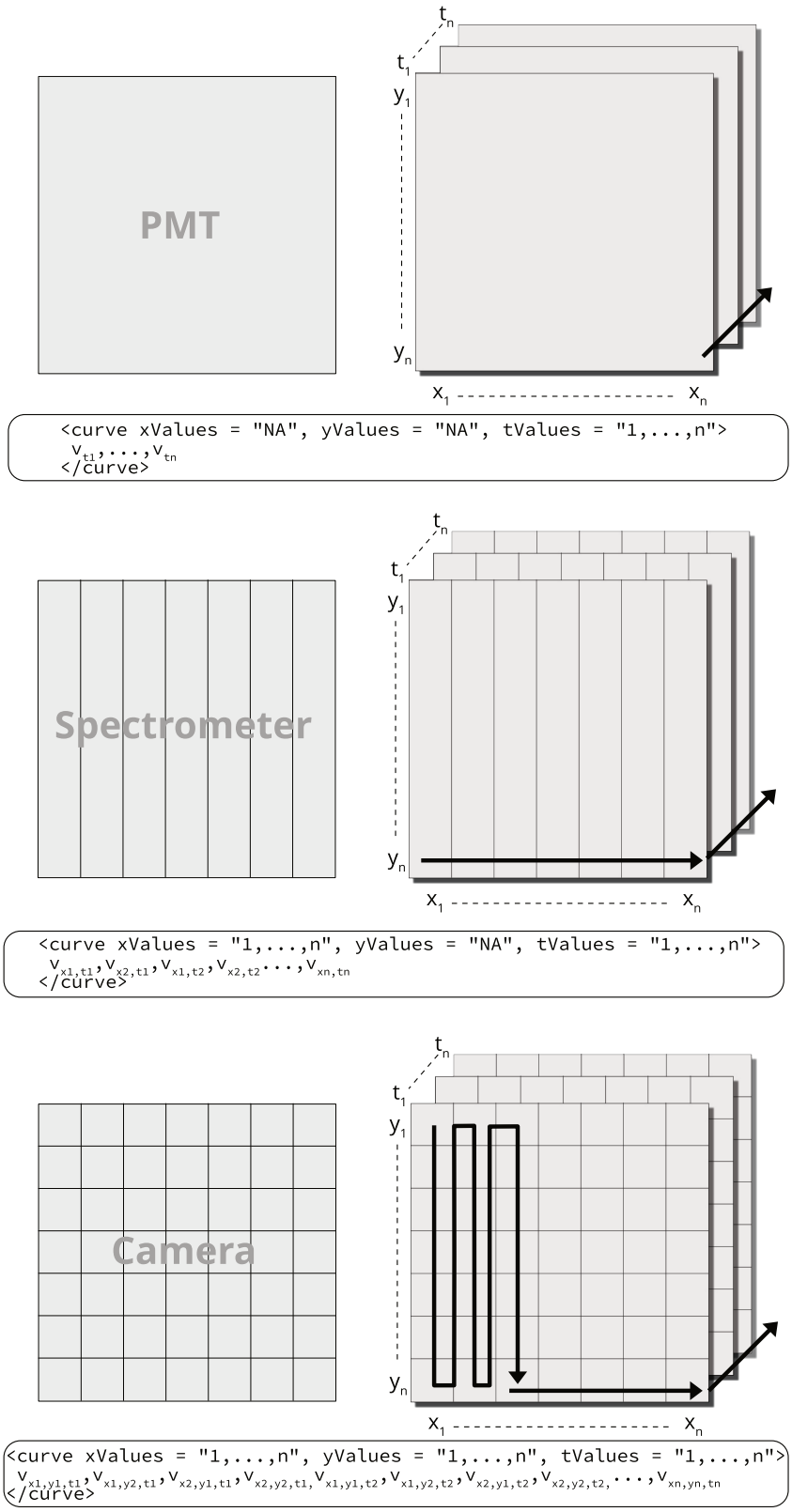

Figure 1: The three cases for data storage depending on the detector.

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. Listing shows some definitions that are intended on this level.

```
{...}
<record name="TL measurement">
  <curve startDate="2021-02-14T225935.0Z" duaration="120" offset="0"
     curveType="predefined" curveDescriptor="time (s), temperature (K)"
     stimulator="heating element">
        0,273;60,393;120,493
  </curve>
  <curve startDate="2021-02-14T225935.0Z" duaration="120" offset="0"
     curveType="measured" curveDescriptor="time (s), luminescence (cts)"
     stimulator="heating element" detector="EMI 9235QB" detectionWindow="375">
        0,30;10,33;20,452;...;120,200;
    </curve>
    {...}
</record>
{...}
```

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.

3.2 The <record> level

The Record defines one process. That process is further described by the stored curves. A record can also be understood as one step of measurement. Curves are stored in a record and used to describe it.

3.2.1 Attributes

identifier	data type / structure	example	information
startDate	`Date`	20130214100122	date when the computer software started the measurement
endDate	`Date`	20130213100242	date when the computer software finished the measurement
recordType	`Enumeration`	TL	legal values found in next table
name	`String`	2nd thermolumi-nescence	user given name of the record
sequenceStepNumber	`number`	5	index in the sequence
metaIrrType	`String`	none	describes the type of irradiation
metaIrrDuration	`number`	7200.0	time [s] the aliquot has been irradiated
sampleCondition	`Enumeration`	dose	legal values found in next table

Valid recordTypes:

recordType	information
bleaching	bleaching
irradiation	irradiation
atmosphereExchange	atmosphere exchange
preheat	preheat

recordType	information
spectrometer	spectrometer
camera	camera
TL	thermoluminescense
RF	radiofluorescense
OSL	optical stimulated luminescense
pause	pause
custom	custom
none	none
POSL	pulsed optical stimulated luminescense

valid sampleConditions:

sampleConditions	information
Natural	Natural
Natural+Dose	Natural+Dose
Bleach	Bleach
Bleach+Dose	Bleach+Dose
Nat.(Bleach)	Nat.(Bleach)
Nat.+Dose(Bleach)	Nat.+Dose(Bleach)
Dose	Dose
Background	Background

3.2.2 Example

All curves within the record should have been detected within the same time frame. An example is given in listing

```
{...}
<Sequence name="SAR OSL CW">
      {...}
      <Record startDate="20130214225935" endDate="20130214230124"
       recordType="OSL" name="green OSL CW" comment="standard OSL curve for Le and Ne"
      >
    <Curve {...}>
      {...}
        </Curve>
        <Curve {...}>
          {...}
        </Curve>
    {...}
      </Record>
      <Record startDate="20130214230132" endDate="20130222013020"
       recordType="bleach" name="bleaching"
      >
        <Curve {...}>
          {...}
        </Curve>
      </Record>
      {...}
</Sequence>
{...}
```

3.3 The <sequence> level

A sequence describes multiple measurements sequential used at one aliquot. The first step often is the unloading of a probe from the storage, and the last to transfer the probe back.

3.3.1 Attributes

identifier	data type / structure	example	information
position	number	42	position in the sample wheel
creationDate	Date	20130214100242	date when this sequence was created
name	String	SAR OSL CW - derivative	user name for this sequence
protocol	String	SAR OSL CW	name of the protocol used
mineral	String	aluminium oxide	name of the substance used

In the luminescence context a sequence can fit a SAR protocol. It is defined that all measurements hold by one sequence are made with one aliquot.

3.3.2 Example

```
<Sequence state="finished" parentID="1010071454551909" name="TL spectra Feldspar" position="18" comment=
  {...}
</Sequence>
```

3.4 The <sample> level

The sample builds the top level. In one file is always only one sample stored. It defines the parameter of the used system.

3.4.1 Attributes

identifier	data type / structure	example	information
name	number	analyse of Hanburg/Germany/1989	user given name
user	String	Jane Doe	name of the person in authority
startDate	Date	20130217100242	start of the analyse
sampleCarrier	String	unknown	used sample carrier
lexsyg	number	201312	ID to identify the used measurement syste
lexStudioVersion	decimal	1.0	version of the used measurement software
firmwareVersion	decimal	0.6	version of the firmware in the lexsyg syste
os	String	aluminium oxide	operating system of the computer

3.4.2 Example

```
<Sample state="finished" parentID="0" name="TL spectra Feldspar" user=""
    startDate="20131007145455" sampleCarrier="" leXLUMID="11-re-01-0006"
    lexStudioVersion="Lexstudio 2beta 0.15.0"
    firmwareVersion="unknown" os="Microsoft Windows NT 6.1.7601 Service Pack 1" comment="">
    {...}
</Sequence>
```

3.5 The <xlum> level

The <xlum> is the format parent node

3.5.1 Attributes

Identifier	Type	Allows NA?	Example	Information
xml:lang	string'	no	: *predefined*	ISO 639-1 language code, must not be changed
version	numeric	no	"1.0"	The xlum format version number relevant
flavour	string	no	"generic"	Allows to specify a particular flavour of the format
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	string	yes	"10.5281/zenodo.596252"	A digital document identifier referencing the dataset already archive in a data repository

3.5.2 Example

```
{...}
<xlum xml:lang="en" formatVersion="1.0" flavour="generic" author="Marie Curie"
  license="CC BY 4.0" doi="NA">
{...}
```

3.6 General parameters

There are a few parameters that are stored on every level.

identifier	data type / structure	example	information
state	enumeration	analyse of Hanburg/Germany/1989	Values: {finished, recording, ...}
parentID	number	201007145551910	ID of the parent node
comment	String	measurement to verify a assumtion	

3.7 Pulsing data

In measurements with pulsed light stimulation, multiple detector curves might be stored per record, one for each (x) pulse(-s). The record contains some additional Metadata

identifier	data type	example	information
OnTime	`number`	1E-01	on time of stimulation per pulse, [s]
OffTime	`number`	1E-01	on time of stimulation per pulse, [s]
NrPulses	`number`	10	number of stimulation pulses total
Summations	`number`	2	how many consecutive pulse records are summated (e.g. 10 pulses, 2 summations -> 5 records)
ChannelsPerPulse	`number`	1000	number of channels that are recorded in one curve
CountsNormalized	`number`	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

identifier	data type	example	information
PulseNr	`number`	1	current Pulse

Pulsing Curves typically contain value pairs of channel time in microseconds and counts, see curveDescripter.

4 Practical guidlines

An import function must not crash parsing the following minimal example.

```
<?xml version="1.0" encoding="utf-8"?>
<Sample>
  <Sequence>
    <Record>
      <Curve>
      </Curve>
    </Record>
  </Sequence>
</Sample>
```