XSYG-file format: Documentation

Steve Grehl, Sebastian Kreutzer, Michael Höhne

version: 1.0 [23 October 2013]

Table of Contents

# Motivation

In 2011 Freiberg Instruments introduced a newly developed luminescence reader called lexsyg. Along with routine measurement data the system records a variety of addi- tional system information (e.g. LED power, heating ramp curves), which provide fully control over the experimental conditions. However, so far, in the luminescence dating community no appropriate data format exists to sufficiently store the obtained data.

With this documentation a new XML based file format (XSYG) is introduced to support the variety of needs of luminescence measurements. In contrast to the existing binary file format from Risø, which mainly focuses on low data volumes, the XSYG format, as an XML derivate, is easy to read and extendible. With this approach the various setups of the lexsyg measurement system is fully covered. As the format is human readable, the data can be easily checked with any kind of text reader, on any major operating system. This allows a platform independent operation. The following documentation provides basic information on the file XSYG format and accounts for the needs of individual flexible data analysis, even with self written software applications.

# General description

First of all the XSYG format is an xml derivative. That means the based data structure is a tree with nodes storing the data. The depth of the tree is 4 levels. Any level has its unique denotation.

Level overview

|  |  |  |  |
| --- | --- | --- | --- |
| level name | number of aliquots | number of measurements | number of raised data curves |
| sample |  |  |  |
| sequence | 1 |  |  |
| record | 1 | 1 |  |
| curve | 1 | 1 | 1 |

<?xml?>  
<Sample>  
 <Sequence>  
 <Record>  
 <Curve stimulator="irradiation unit" />  
 <Curve detector="PMT">  
 x0,y0;x1,y1;x2,y2;x3,y3  
 </Curve>  
 </Record>  
 </Sequence>  
</Sample>

The levels and a short introduction is given in table, a deeper description can be found in the following sections. Each level in the structure is represented by a node. This node has a name, attributes, and data stored in it. The data stored in the attributes describes the state of this level. The stored data in the node describes the process(es) assigned to the node. An abstract example is shown in listing. The documentation provides an overview starting with the leaf node description and going to the root from there. This is done to start in a comprehensible environment and abstract from there.

# Detailed description

## Description of the curve level

The curve level is the deepest level (leaf) and so has no further sub-levels. A curve can be assassinated with the activity of one component of the measurement system during a record. For example the heating is predefined by a ramp, this data is stored in one curve. Another example is the measuring through a photomultiplier tube. The measured data and the setup of the PMT will be stored in one curve node, may in the same record as the heating (f.e. to document a TL).

Allowed curve attributes

|  |  |  |  |
| --- | --- | --- | --- |
| identifier | data type structure | example | information |
| startDate | Date | 20130214225935 | Format: yyyyMMddhhmmss |
| duration | number | 20 | duration of the measurement / stimulation in seconds |
| curveType | enumeration | predefined | Values: {predefined, measured} |
| stimulator | String | standard heating component | technical name of the component |
| detector | String | UV-VIS | technical name of the component |
| curveDescripter | Pair (String, String) | time, counts | x, y |
| offset | number | 60 | idle time till the component starts [s] |
| [detectionWindow] | number | 375 | wavelength window being detected |
| [filterIDs] | List (number) | 14783;89362 | technical identifiers of the used filters |
| [filterNames] | List (String) | infrared filter; interference filter | name of the used filters |

In order to define a standard a few attributes are required. A few of them are useful for every kind of component, some other are only meaningfully in combination. To define a detection window for a heating component is not intended, but for a measurement device (f.e. a PMT) it is useful. Listing shows some definitions that are intended on this level.

{...}  
<Record name="ITl core measurement">  
 <Curve startDate="20130214225935" duaration="120"   
 curveType="predefined" stimulator="standard heating component"  
 curveDescripter="time,temperature">  
 0,273;60,393;120,393  
 </Curve>  
 <Curve startDate="20130214225935" duaration="60"  
 curveType="measured" detector="UV-VIS"  
 detectionWindow="375" offset="60"  
 curveDescripter="time,counts">  
 0,30;100,33;200,452;{...}  
 </Curve>  
 {...}  
</Record>  
{...}

One curve is related to one technical device. To define a accurate time window, it is in intended to use the parameters *duration* and *offset*. These parameters are related to the start of the superior record. Since new devices find their way into measurement systems will be extended by more attributes.  
All measurements, predefinitions and settings should be saved.

### Spectrometer data

Typically two dimensional data is stored inside the curve nodes. Therefore only two separators are needed. The comma to separate two coordinates and the semicolon to separate two points. When it comes to spectrometer data this concept doesn’t fit. The reason for this are multiple values (one per wavelength) for one point in time. Therefore we defined the values for every wavelength at one time in an array. This starts with and finishes with , the wavelength are separated by .

{...}  
<Curve {...} >  
 1.0,[554|555|559|553|556|550| {...} |570|569|572];10.0,[562|561|564| {...}  
</Curve>  
{...}

## 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.

|  |  |  |  |
| --- | --- | --- | --- |
| 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 thermoluminescence | 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 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 |
| 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 |

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>  
{...}

## 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.

|  |  |  |  |
| --- | --- | --- | --- |
| 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.

<Sequence state="finished" parentID="1010071454551909" name="TL spectra Feldspar" position="18" comment="" creationDate="20131007145455" protocol="" mineral="">  
 {...}  
</Sequence>

## 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.

|  |  |  |  |
| --- | --- | --- | --- |
| 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 |
| lexsygID | number | 201312 | ID to identifie the used measurement system |
| lexStudioVersion | decimal | 1.0 | version of the used measurement software |
| firmwareVersion | decimal | 0.6 | version of the firmware in the lexsyg system |
| os | String | aluminium oxide | operating system of the computer |

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

## 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 |  |

## 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.