

# Closed-loop experiments and metadata management with *Relacs* and *LabLog*



**Jan Benda & Jan Grewe**

Department Biology II  
Ludwig-Maximilian University  
Munich, Germany



Bernstein Prize in  
Computational Neuroscience

# Content

**Part I** Closed-loop experiments: *RELACS*



**Part II** Data management: *LabLog*



**Part III** Metadata exchange

# Part I — Closed loop experiments with *RELACS*



by **Jan Benda**

[www.relacs.net](http://www.relacs.net)



## “Traditional” Experiments

---

1. A set of stimuli and a more or less fixed experimental protocol are prepared



## “Traditional” Experiments

1. A set of stimuli and a more or less fixed experimental protocol are prepared
2. The recordings are done on a few cells (> one week of experimental work)
  - the raw voltage trace is the only feedback
  - changes during the running experiment often involve manual manipulations  
⇒ precious recording time is wasted



## “Traditional” Experiments

1. A set of stimuli and a more or less fixed experimental protocol are prepared
2. The recordings are done on a few cells (> one week of experimental work)
  - the raw voltage trace is the only feedback
  - changes during the running experiment often involve manual manipulations  
⇒ precious recording time is wasted
3. The data are analyzed offline



## “Traditional” Experiments

1. A set of stimuli and a more or less fixed experimental protocol are prepared
2. The recordings are done on a few cells (> one week of experimental work)
  - the raw voltage trace is the only feedback
  - changes during the running experiment often involve manual manipulations  
⇒ precious recording time is wasted
3. The data are analyzed offline
4. The stimuli and the protocol are modified



## “Traditional” Experiments

1. A set of stimuli and a more or less fixed experimental protocol are prepared
2. The recordings are done on a few cells (> one week of experimental work)
  - the raw voltage trace is the only feedback
  - changes during the running experiment often involve manual manipulations  
⇒ precious recording time is wasted
3. The data are analyzed offline
4. The stimuli and the protocol are modified
5. A new set of recordings is made





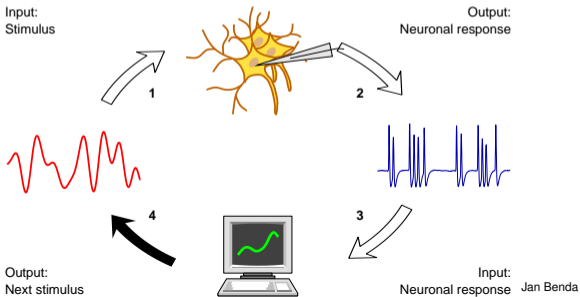
## “Traditional” Experiments

1. A set of stimuli and a more or less fixed experimental protocol are prepared
2. The recordings are done on a few cells (> one week of experimental work)
  - the raw voltage trace is the only feedback
  - changes during the running experiment often involve manual manipulations  
⇒ precious recording time is wasted
3. The data are analyzed offline
4. The stimuli and the protocol are modified
5. A new set of recordings is made
6. After several iterations a paper is written



## Closed-Loop Experiments

1. Present a stimulus
2. Record the response
3. Immediately analyze and visualize the data
4. Generate the next stimulus

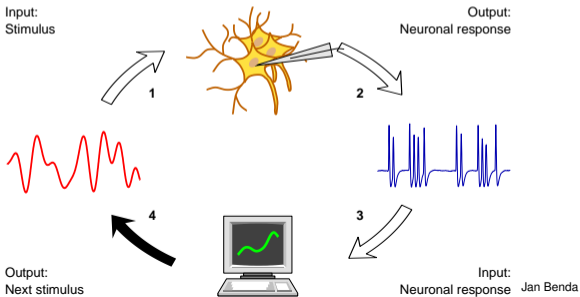




## Closed-Loop Experiments

*RELACS* is designed as a framework for closed-loop experiments that

- considerably speed up this traditional approach
- offer novel experimental possibilities





## Simple Closed-Loop Experiments

- Online visualization of processed data:
    - General infos, e.g. quality of spike detection, sensitivity of the cell, temperature, condition of animal, ...
    - Specific results, e.g. spike raster, firing rates, spike-triggered averages, ...
- ⇒ Speeds up manual closed-loop



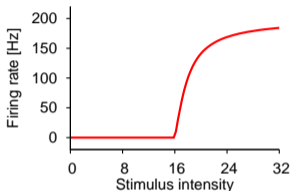
## Simple Closed-Loop Experiments

- Online visualization of processed data:
  - General infos, e.g. quality of spike detection, sensitivity of the cell, temperature, condition of animal, ...
  - Specific results, e.g. spike raster, firing rates, spike-triggered averages, ...
- ⇒ Speeds up manual closed-loop
- Set stimuli relative to the neuron's dynamic range
- Automatically control motorized electrodes (great for dual unit recordings!)
- ...



## Example 1: Tuning Curve Measurement

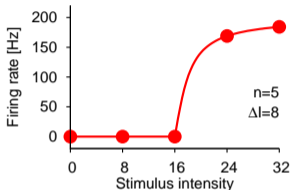
Traditional:





## Example 1: Tuning Curve Measurement

Traditional:



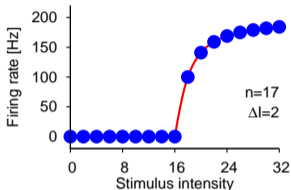
either:

fast  $\rightarrow$  low resolution



## Example 1: Tuning Curve Measurement

Traditional:



either:

fast  $\rightarrow$  low resolution

or:

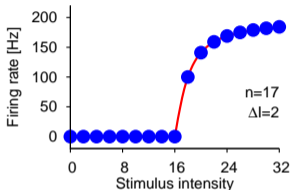
high resolution  $\rightarrow$  slow





# Example 1: Tuning Curve Measurement

Traditional:



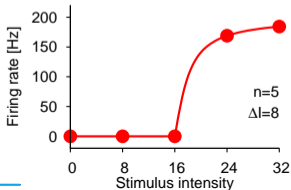
either:

fast  $\rightarrow$  low resolution

or:

high resolution  $\rightarrow$  slow

Closed loop:

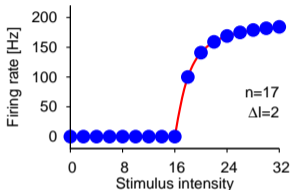


1. start with low resolution



# Example 1: Tuning Curve Measurement

Traditional:



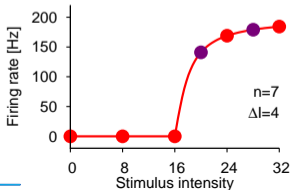
either:

fast  $\rightarrow$  low resolution

or:

high resolution  $\rightarrow$  slow

Closed loop:

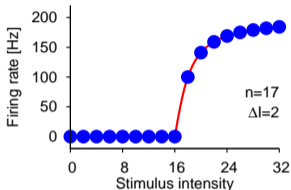


1. start with low resolution
2. increase resolution *where necessary!*



## Example 1: Tuning Curve Measurement

Traditional:



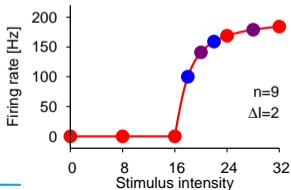
either:

fast  $\rightarrow$  low resolution

or:

high resolution  $\rightarrow$  slow

Closed loop:

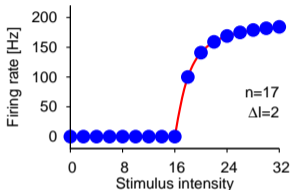


1. start with low resolution
2. increase resolution *where necessary!*
3. further increase resolution



# Example 1: Tuning Curve Measurement

Traditional:



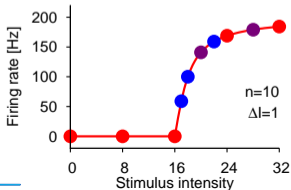
either:

fast  $\rightarrow$  low resolution

or:

high resolution  $\rightarrow$  slow

Closed loop:



1. start with low resolution
2. increase resolution *where necessary!*
3. further increase resolution



## Advanced Closed-Loop Experiments

New experimental designs are possible:

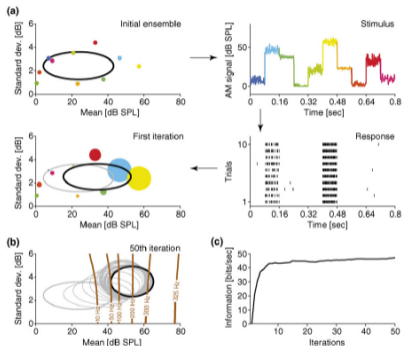
- Optimal search for a neuron's receptive field.
- Search for stimuli that drive a neuron in an "optimal" way.
- Find set's of stimulus parameter that result in the same response (iso-response method).
- ...

Benda et al. (2007): "From response to stimulus: adaptive sampling in sensory physiology." *Curr. Opin. Neurobiol.* **17**: 430–436.

⇒ currently 13 scientific publications based on *OEL/RELACS* data in *Neuron*, *J Neurosci*, *PLoS Biol*, *Nat Neurosci*, *J Neurophysiol*, etc.



## Example 2: Optimal Stimulus Ensembles



Machens et al. (2005) *Neuron* 17: 47–56.



## Example 3: Iso-Response

---

Gollisch et al. (2002) *J Neurosci* **22**: 10434–10448.



# RELACS

... enjoy your recordings

## Relaxed Electrophysiological data Acquisition, Control, and Stimulation

The screenshot displays the RELACS software interface with the following components:

- Spikes-1 Panel:** Configuration for spike detection with a minimum threshold of 14.5 mV, a current threshold of 35.2 mV, a ratio threshold/size of 40%, and a spike size of 95.8 mV. It includes a legend for detected (green), not detected (red), threshold (white), and min thresh (yellow) spikes, along with a small waveform plot and 'Dialog'/'Help' buttons.
- Main Plot:** A graph of voltage (V-1 [mV]) over time (ms) showing a series of green action potentials.
- Info Panel:** Contains two sub-plots: 'Spectrum' showing Threshold [dB] vs Frequency [kHz] with 'Best side' set to 'left' speaker and 'Best frequency' at 4.0 kHz; and 'Firing rate' showing Firing rate [Hz] vs Intensity [dB SPL] with 'Best threshold' at 42.5 dB SPL. A 'Cell Found' button is present.
- F4 Curve Panel:** Shows 'Mean firing rate = 101Hz' with a plot of firing rate [Hz] over time [ms]. Below it is a 'Stimulus [dB]' plot over time [ms].
- Control Bar:** Features buttons for 'Search (ESC)', 'FField (F1)', 'Stimulus (F2)', and 'Help (F3)'. A status bar at the bottom shows 'no files open', '12%' zoom, and file information: 'AM: OU-noise20Hz1228223405 , intensity: 55.0 dB SPL, Amplitude: 6.0 dB, Loop 19 of 20'.

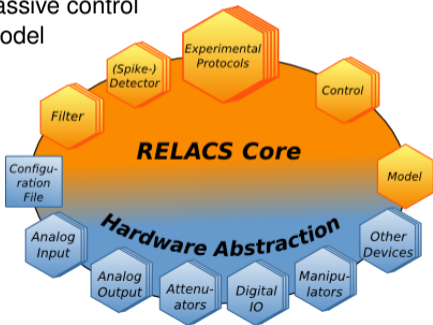




## Modular Design

RELACS core with flexible C++ Plugins for

- hardware abstraction
- data pre-processing (filter, spike detectors)
- **experimental protocols**
- passive control
- model



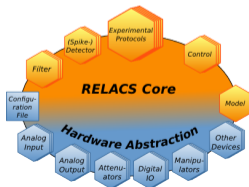


# Hardware Independent Protocols

RELACS integrates all hardware components.

## *Experimental protocols* for RELACS

- are implemented independently of specific hardware
- can be used on all the different experimental setups in your lab without any modifications
- can be **shared** with other labs





## Metadata Acquisition with *RELACS*

*RELACS* records many metadata:

- All *RELACS*-controlled hardware settings (e.g. sampling rate)
- All settings and version numbers of the experimental protocols
- Properties of the stimuli used by the experimental protocols
- Main characteristics of the recorded cell
- General infos about the experiment (from a dialog)



# Metadata Acquisition with RELACS

RELACS - Relaxed Electrophysiological data Acquisition, Control, and Stimulation: Version 0.9.6 (11/24/0...)

File Plugins Devices Detectors/Filters BeProS Macros View Help

**Spikes-1**

Minimum threshold: 14.5 mV  
Threshold: 36.5 mV  
Ratio threshold / size: 40 %  
Spike size: 84.7 mV

— detected  
— not detected  
— threshold  
— min thresh

Dialog Help

**Info** Spectrum

Best side: left speaker  
Best frequency: 6.0 kHz  
Best threshold: 45.0 dB SPL

Cell Found

**ie** CalbMicrophone CalbSpeakers F-I Curve F-I Field Search

Threshold [dB SPL] vs Frequency [kHz]  
Firing rate [Hz] vs Intensity [dB SPL]

Search (ESC) FField (F1) Stimulus (F2) Help (F3)

00:01 2008-12-02-ab/ 12% Frequency 4 kHz, Increment 4, Intensity 55 dB SPL, Loop 2

# Metadata Acquisition with *RELACS*

The screenshot displays the RELACS software interface with a 'Stop Session Dialog' window open. The dialog is divided into 'Meta Data' and 'Setup' tabs. The 'Recording' section includes fields for Recording quality (Good), Comment, Scientist (Jan Benda), Temperature (22 °C), and Humidity (54 %). The 'Cell' section includes Cell type (Low-frequency receptor), Recording location (Auditory nerve), Side (Left), and Depth (0 μm). The 'Subject' section includes Species (Locusta migratoria), Sex (Female), Age (Middle-aged), and Preparation (in vivo dorsal). The dialog has Save, Discard, Reset, and Cancel buttons. The background shows the main software interface with a 'Spikes-1' plot, a 'Spectrum' plot, and a 'Firing rate' plot. The status bar at the bottom shows '00:01 2008-12-02-ab/ 12% Frequency 4 kHz, Increment 4, Intensity 55 dB SPL, Loop 2'.

**Spikes-1**

Minimum threshold: 14.5  
Threshold: 36.5  
Ratio threshold / size: 40  
Spike size: 84.7

**Info** | Spectrum

Threshold [dB]: 100, 80, 60, 40, 20  
Frequency [kHz]: 0, 10, 20, 30, 40  
Best side: left  
Best frequency: 6.0  
Firing rate [Hz]: 10, 5, 0  
Intensity [dB SPL]: 50, 100  
Best threshold: 45.0

**Stop Session Dialog**

Meta Data | Setup

**Recording**

Recording quality: Good  
Comment:   
Scientist: Jan Benda  
Temperature: 22 °C  
Humidity: 54 %

**Cell**

Cell type: Low-frequency receptor  
Recording location: Auditory nerve  
Side: Left  
Depth: 0 μm

**Subject**

Species: Locusta migratoria  
Sex: Female  
Age: Middle-aged  
Preparation: in vivo dorsal

Save | Discard | Reset | Cancel

Search (ESC) | FField (F1) | Stimulus (F2) | Help (F3)

00:01 2008-12-02-ab/ 12% Frequency 4 kHz, Increment 4, Intensity 55 dB SPL, Loop 2



# Metadata Acquisition with *RELACS*

The screenshot displays the RELACS software interface with a 'Stop Session Dialog' window open. The dialog has two tabs: 'Meta Data' and 'Setup'. The 'Setup' tab is active, showing the following parameters:

- Recording**
  - Recording quality: Good
  - Comment: (empty)
  - Scientist: Jan Benda
  - Temperature: 22 °C
  - Humidity: 54 %
- Cell**
  - Cell type: Low-frequency receptor
  - Recording location: Auditory nerve
  - Side: Left
  - Depth: 0 μm
- Subject**
  - Species: Locusta migratoria
  - Sex: Female
  - Age: Middle-aged
  - Preparation: in vivo dorsal

Buttons at the bottom of the dialog are Save, Discard, Reset, and Cancel. The background shows the main software interface with a 'Spikes-1' plot (0-100 mV), a 'Spectrum' plot (0-40 kHz), and a 'Firing rate' plot (0-100 Hz). The status bar at the bottom indicates '00:01 2008-12-02-ab/ 12% Frequency 4 kHz, Increment 4, Intensity 55 dB SPL, Loop 2'.

Minimal manual input necessary!



## Metadata Acquisition with *RELACS*

How should  
the metadata  
be stored?



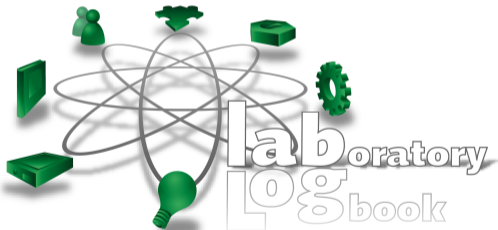
How should  
the metadata  
be stored?

Fileformat?  
Vocabulary?



# Part II —

# LabLog — the long-term memory of your lab —



by **Jan Grewe**

[lablog.sourceforge.net](http://lablog.sourceforge.net)



## LabLog - the Laboratory Logbook

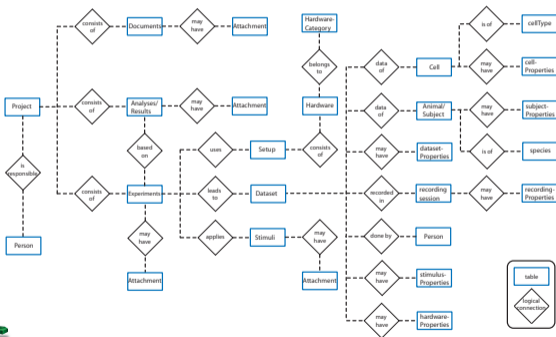
- Storage of project related information:
  - Lab-journal (ideas, diary)
  - Experimental setup and hardware
  - Projects
- Management of acquired data:
  - Storage of metadata directly from *RELACS*
  - Search for data within/across projects (by SQL queries)
  - Export search results for further analysis
- Platform independent Java frontend to a mySQL database





# Structure

The structure of the underlying relational database (about 60 tables):





# Screen Shots

LaboratoryLogbook - the long term memory of your lab - connected to Lab08@localhost

File Persons Groups Journal Data

Context Diagnostics People Groups Journal

### Journal

projects

Name	RI	Status
Influences of non-linearities on TC response reliability	Jan Grewe	active
Impact of photon noise on HT reliability	Jan Grewe	done
Optomotor head pitch	Romy Rosner	active
Information and Flexibility in the Mammalian Cortex	Jan Grewe	done

Your Diary Project Definition Project Documentation Dataset Manager Dataset Templates Project Report Setups Hardware

data edit dataset edit subject edit cell

enter retrieve

name	file	folder	rec. date	experimenter
experiment	comment	rem. comment	quality	properties
rec. settings	rem. rec. settings	rem. subject	add subject	rem. cell
add cell	delete			

datasets

- ▼ @CH\_09012008
  - 11\_43\_25.ds
  - 11\_43\_49.ds
  - 11\_44\_47.ds
  - 11\_45\_10.ds
  - 11\_45\_55.ds
  - 11\_46\_18.ds
  - 11\_46\_41.ds
  - 11\_47\_03.ds
  - 11\_49\_58.ds

summary

**Dataset: 11\_44\_47.data (ID: 65) Quality: 3**

recording date: 2008-01-09 recorded by: Jan Grewe

file name:  
11\_44\_47.data

file location:  
data2008.01.09

comment:  
no comment found

driver registered  
connection established



# Screen Shots

Retrieve Data

field selection

table	field
datasets	

condition editor

connec...	table	field	condition
-----------	-------	-------	-----------

sql command

```
SELECT d.datasetID,concat(d.filename,d.datafolder,d.filename) as file, e.name  
FROM datasets d, experiments e  
WHERE d.experimentID = e.id AND e.name like '%Control 1%';
```

build  
open  
store  
got

search results

datasetID	file	name
63	11_43_25.data\data\2008.01.09\11_43_25.data	Control 1: Apparent motion with 5 and 10 Hz
64	11_43_49.data\data\2008.01.09\11_43_49.data	Control 1: Apparent motion with 5 and 10 Hz
65	11_44_47.data\data\2008.01.09\11_44_47.data	Control 1: Apparent motion with 5 and 10 Hz
66	11_45_10.data\data\2008.01.09\11_45_10.data	Control 1: Apparent motion with 5 and 10 Hz
67	11_45_55.data\data\2008.01.09\11_45_55.data	Control 1: Apparent motion with 5 and 10 Hz
68	11_46_18.data\data\2008.01.09\11_46_18.data	Control 1: Apparent motion with 5 and 10 Hz
69	11_46_41.data\data\2008.01.09\11_46_41.data	Control 1: Apparent motion with 5 and 10 Hz
70	11_47_03.data\data\2008.01.09\11_47_03.data	Control 1: Apparent motion with 5 and 10 Hz
71	11_49_56.data\data\2008.01.09\11_49_56.data	Control 1: Apparent motion with 5 and 10 Hz
72	14_32_14.data\data\2008.01.30\14_32_14.data	Control 1: Apparent motion with 5 and 10 Hz
73	14_33_38.data\data\2008.01.30\14_33_38.data	Control 1: Apparent motion with 5 and 10 Hz
74	14_34_00.data\data\2008.01.30\14_34_00.data	Control 1: Apparent motion with 5 and 10 Hz
75	14_34_23.data\data\2008.01.30\14_34_23.data	Control 1: Apparent motion with 5 and 10 Hz
76	14_34_48.data\data\2008.01.30\14_34_48.data	Control 1: Apparent motion with 5 and 10 Hz

export close





How should  
the metadata  
be imported  
into the database?

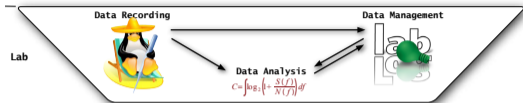


**Part III —**

# **Talking about data**

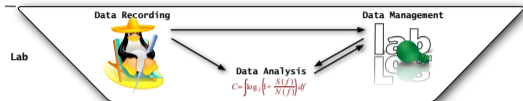
**an Extensible Framework  
for Metadata Exchange**

# The Data-Chain



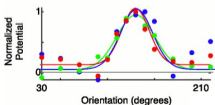
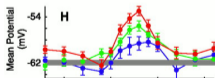
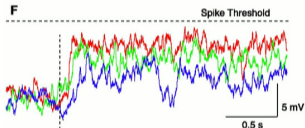


# The Data-Chain

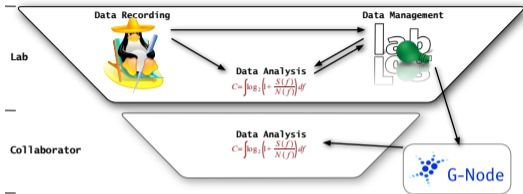


## The Contribution of Noise to Contrast Invariance of Orientation Tuning in Cat Visual Cortex

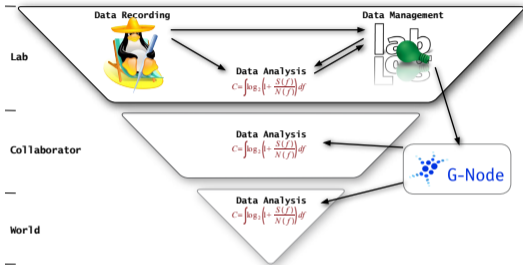
Jeffrey S. Anderson, Ilan Lampl, Deda C. Gillespie, David Ferster\*  
Science, 2000



# The Data-Chain



# The Data-Chain



- All data transfer requires talking about data.
- How to exchange metadata?





## Metadata



- is “data about data”.
- describe recording conditions.

stimulus.type = white noise

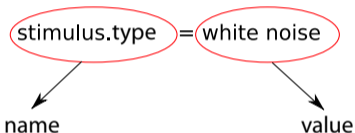




## Metadata



- is “data about data”.
- describe recording conditions.





## Metadata



- is “data about data”.
- describe recording conditions.

cell.baselineRate = 50 +/- 5 Hz

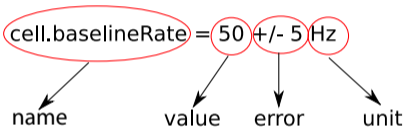




## Metadata



- is “data about data”.
- describe recording conditions.

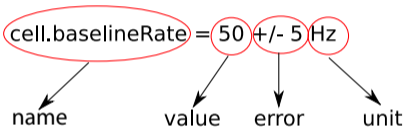




## The Metadata Problem



- is “data about data”.
- describe recording conditions.



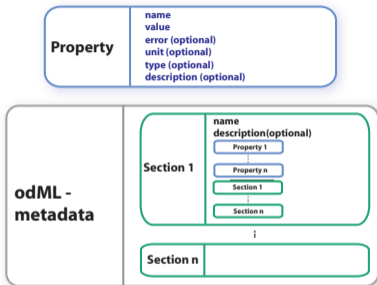
- What name to choose?
- What does it mean?
- How to organize metadata?







# Structure — How to organize metadata?



Implemented as the **odML** XML Schema





## The odML Schema



“normal” XML:

- A schema defines a fixed set of valid tags (names) and their relation
- Unknown properties invalidate the XML file
- The schema needs to be updated





## The odML Schema



“normal” XML:

- A schema defines a fixed set of valid tags (names) and their relation
- Unknown properties invalidate the XML file
- The schema needs to be updated

“odd” XML — [odML](#) (open metadata markup language):

- The schema just defines the section and property structure (the grammar)
- Unknown properties do not invalidate the XML file
- The vocabulary (definition of properties) is [inherently extensible](#)





## Definition — What name to choose?



- Sections group properties logically. E.g.:
  - Stimulus
  - Experiment description
  - Cell, experimental subject, preparation
  - Hardware properties, hardware settings
  - Dataset
  - Analysis parameter
  - etc.

Implementation: [eVoc](#) XML file conform to odML





## Definition — Examples



### HardwareSettings:

#### Amplifier:

<b>name</b>	<b>type</b>	<b>description</b>
Gain	float	The amplifier gain.
HighpassCutoff	float	The cutoff frequency of the amplifier's highpass filter. Given in Hz.
LowpassCutoff	float	The cutoff frequency of the amplifier's lowpass filter. Given in Hz.
Mode	string	The amplifier mode. E.g. Bridge, CC, VC etc.





## Definition — Examples



### HardwareSettings:

### DataAcquisition:

<b>name</b>	<b>type</b>	<b>description</b>
AIUsedChannelCount	int	The number of used analog input channels.
AI SampleRate[n]	int	The sample rate with which an individual input channel was sampled. Given in Hz.
AIChannelGain[n]	float	The gain of an input channel.
AIReference[n]	string	The reference to which voltages were measured. Usually either "common ground" or "differential".
AI Polarity[n]	string	The polarity of the measurement "unipolar" or "bipolar".





## Definition — Examples



### Cell:

<b>name</b>	<b>type</b>	<b>description</b>
Name	string	An identifier of this cell.
Type	string	The type of the recorded cell.
BrainRegion	string	The Region the cells are located in. For example Retina, Cortex, Cerebellum etc.
BrainSubRegion	string	For example CA1 in hippocampus.
Layer	string	For example layer 4 in CA1.
Ganglion	string	Like the pro-, meta- and mesothoracal ganglion in invertebrates.
RecLocation	string	The recording location in the cell. Axonal, dendritic,somatic?
RestingPotential	float	What is the cell's resting potential. An indicator for the recording quality.
BaselineRate	float	For spiking cells, the spontaneous activity which might be an indicator for the cell's health status and thus recording quality.





## How to use eVoc?



### 1. Assemble properties:

- If you find an appropriate property in **eVoc**, use it!
- Ignore all properties that do not match.
- Add your own properties that are not yet in **eVoc**, if possible with a description.







## How to use eVoc?



1. Assemble properties:
  - If you find an appropriate property in **eVoc**, use it!
  - Ignore all properties that do not match.
  - Add your own properties that are not yet in **eVoc**, if possible with a description.
2. Write them into an **odML** XML file





## How to use eVoc?



1. Assemble properties:
  - If you find an appropriate property in **eVoc**, use it!
  - Ignore all properties that do not match.
  - Add your own properties that are not yet in **eVoc**, if possible with a description.
2. Write them into an **odML** XML file
3. Transfer them to an analysis or database program

This is all done by *RELACS* and *LabLog* automatically!





## Vision



- [odML](#) and [eVoc](#) are publicly available on the G-Node server.
- Discussion forum for extensions to vocabulary and schema.
- Recording, datamanagement, and analysis software should use [eVoc](#) for metadata exchange.
- Data can be published on G-Node database which can also import this kind of metadata.





## Vision

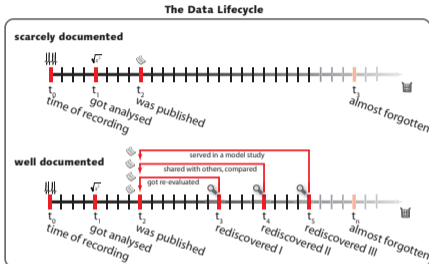


- **odML** and **eVoc** are publicly available on the G-Node server.
  - Discussion forum for extensions to vocabulary and schema.
  - Recording, datamanagement, and analysis software should use **eVoc** for metadata exchange.
  - Data can be published on G-Node database which can also import this kind of metadata.
- ⇒ You can easily share your data!
- ⇒ Your data can be found!





# The Data Life-Cycle

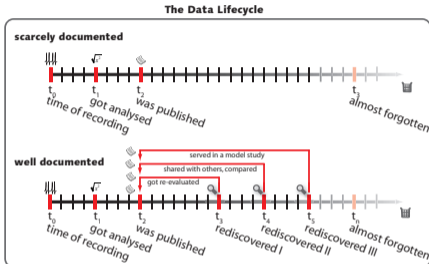


- Meta information tends to vanish with time.
- Thus, re-using of old data is a tedious business.
- Data should be annotated as early as possible (preferentially at the time of acquisition).





# The Data Life-Cycle



- Well annotated data can be found and reused easily

⇒ Your data deserves it!





## Summary



- Closed-loop experiments may considerably speed up electrophysiological experiments and allow for new experimental designs.
- *RELACS* is a software platform for closed-loop experiments and automatically annotates your data ([www.relacs.net](http://www.relacs.net)).
- *LabLog* helps you keeping track of your data ([lablog.sourceforge.net](http://lablog.sourceforge.net))
- A unifying framework for metadata exchange is needed for data sharing between data acquisition, analysis, and management software and services.









## Experimental Protocol Example

```
int Example::main( void ) {
    // some initialization ...
    OutData signal;
    signal.setTrace( "LeftSpeaker" );
    signal.sineWave( frequency, duration, amplitude );
    SampleDataD rate( 0.0, duration, 0.001 );
    for ( int counter=0; counter<Repeats; counter++ ) {
        write( signal );
        sleep( duration + pause );
        eventData spikes( events( SpikeEvents[0] ),
                        events( SpikeEvents[0] ).signalTime(),
                        events( SpikeEvents[0] ).signalTime() + duration );
        double meanrate = spikes.rate( 0.3*duration, duration );
        spikes.addRate( rate, counter, GaussKernel( sigma ) );
        P.clear();
        P.plot( rate, 1000.0, Plot::Yellow, 2, Plot::Solid );
        P.draw();
        if ( meanrate < targetrate ) {
            amplitude *= 2.0;
            signal.sineWave( frequency, duration, amplitude );
        }
    }
}
```



## Why C++

- well structured (object oriented)
- platform independent
- efficient and controllable memory usage
- very fast



## C++ Library for Data Analysis

Data structures (classes, container):

- *Array* — Basic 1-D vector
- *SampleData* — 1-D data vector with regularly sampled time axis
- *Map* — Sequence of  $x|y$  data pairs

Algorithms:

- basic statistics (moments, quartiles, histogram)
- power spectra, coherence, transfer function
- linear fits
- non-linear fits (Simplex, Levenberg-Marquardt)



## C++ Library for Data Analysis

Data structures (classes, container):

- *EventData* — Spikes and other point process data
- *EventList* — Multi-trial spike trains

Algorithms:

- firing rates (mean, PSTH binned/kernel,  $1/ISI$ )
- CV, Fano factor, ISI correlation
- vector strength, reliability, jitter
- mutual information (lower and upper bound)



## Free and Open Source Software

---

*RELACS* is open source and free software distributed under the GNU General Public License (GPL).

- No hassle with licenses of commercial software.
- Add whatever new feature you need directly to the program.
- Share the program and your specific experimental protocols with your collaborators.
- Know what the data-analysis algorithms are doing!