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

“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
     $\Rightarrow$ 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
     $\Rightarrow$ 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
     $\Rightarrow$ precious recording time is wasted
3. The data are analyzed offline
4. The stimuli and the protocol are modified

“Traditional” Experiments

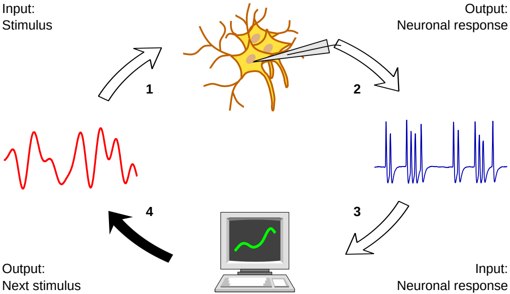
1. A set of stimuli and a more or less fixed experimental protocol are prepared
2. The recordings are done on a few cells ($>$ one week of experimental work)
   - the raw voltage trace is the only feedback
   - changes during the running experiment often involve manual manipulations
     $\Rightarrow$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

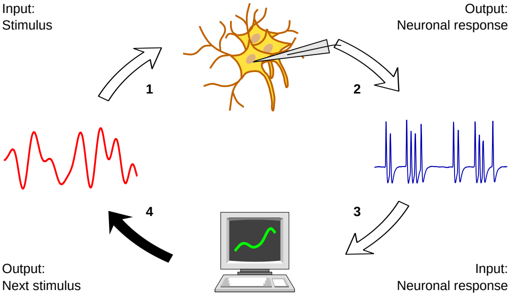
1. Present a stimulus
2. Record the response
3. Immediately analyze and visualize the data
4. Generate the next stimulus



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, ...

$\Rightarrow$ 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, ...

  $\Rightarrow$ Speeds up manual closed-loop

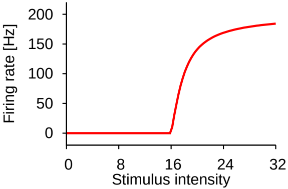
- Set stimuli relative to the neuron's dynamic range

- Automatically control motorized electrodes (great for dual unit recordings!)

- ...

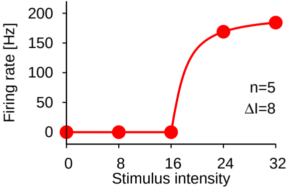
Example 1: Tuning Curve Measurement

Traditional:



Example 1: Tuning Curve Measurement

Traditional:

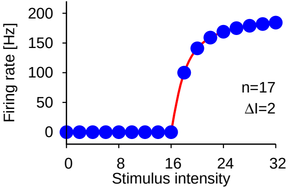


either:

fast $\longrightarrow$ low resolution

Example 1: Tuning Curve Measurement

Traditional:



either:

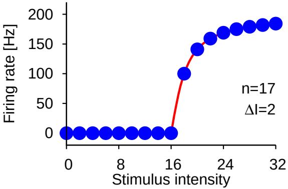
fast ⟶ low resolution

or:

high resolution ⟶ slow

Example 1: Tuning Curve Measurement

Traditional:



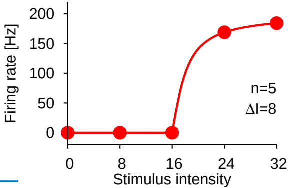
either:

fast $\longrightarrow$ low resolution

or:

high resolution $\longrightarrow$ slow

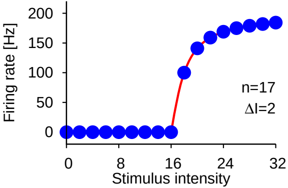
Closed loop:



1. start with low resolution

Example 1: Tuning Curve Measurement

Traditional:



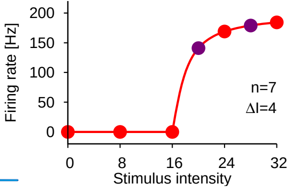
either:

fast $\longrightarrow$ low resolution

or:

high resolution $\longrightarrow$ slow

Closed loop:

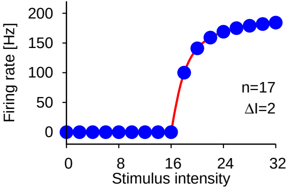


1. start with low resolution

2. increase resolution where necessary!

Example 1: Tuning Curve Measurement

Traditional:



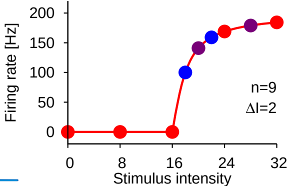
either:

fast $\longrightarrow$ low resolution

or:

high resolution $\longrightarrow$ slow

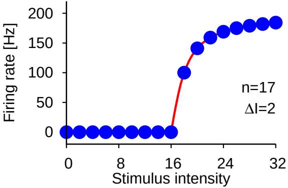
Closed loop:



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

Example 1: Tuning Curve Measurement

Traditional:



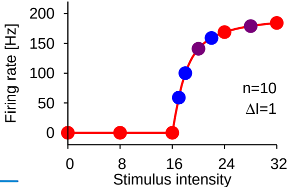
either:

fast $\longrightarrow$ low resolution

or:

high resolution $\longrightarrow$ slow

Closed loop:



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

Advanced Closed-Loop Experiments

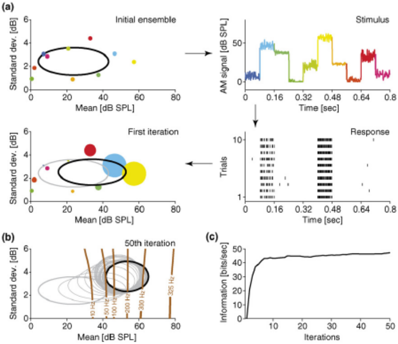
New experimental designs are possible:

- 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.

$\Rightarrow$ 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



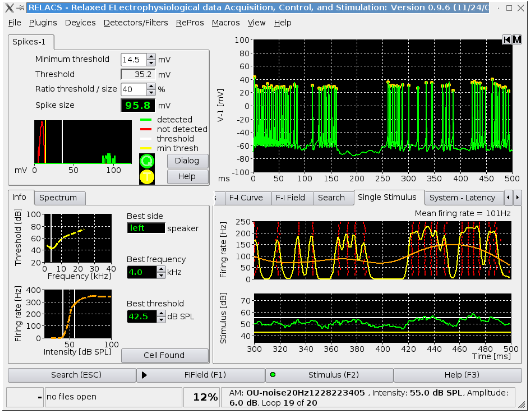
Machens et al. (2005) *Neuron* **17**: 47–56.

Example 3: Iso-Response

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

RELACS ... enjoy your recordings

Relaxed Electrophysiological data Acquisition, Control, and Stimulation

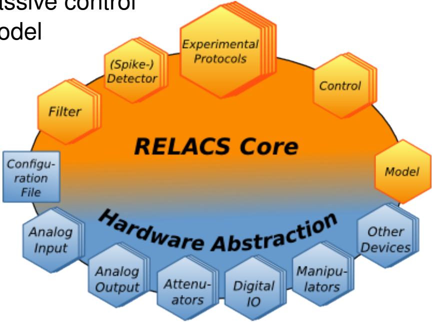


Jan Benda

Modular Design

RELACS core with flexible C++ Plugins for

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

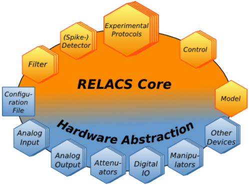


Hardware Independent Protocols

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

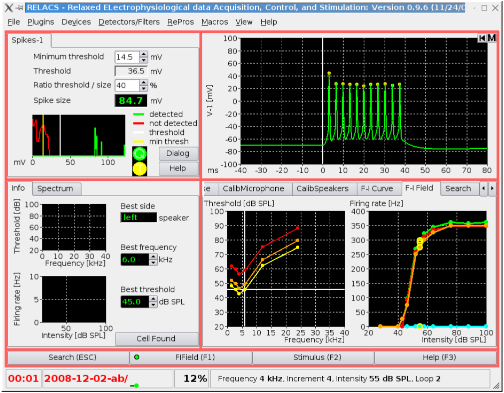


Metadata Acquisition with *RELACS*

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)

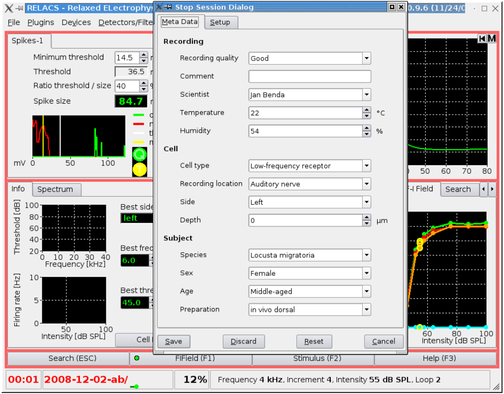
Metadata Acquisition with *RELACS*



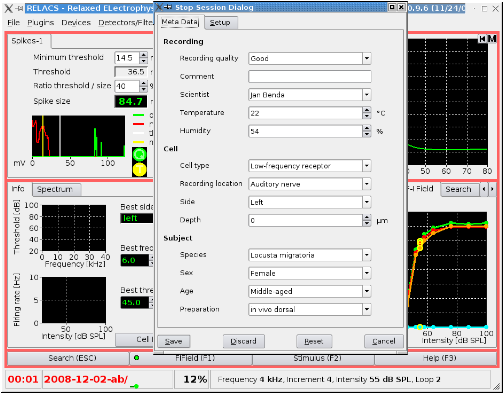
relacs

Jan Benda

Metadata Acquisition with *RELACS*



Metadata Acquisition with *RELACS*



Minimal manual input necessary!

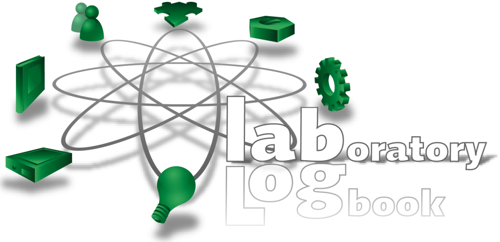
How should
the metadata
be stored?

How should the metadata be stored?

Fileformat?
Vocabulary?

Part II —

LabLog — the long-term memory of your lab —



by **Jan Grewe**

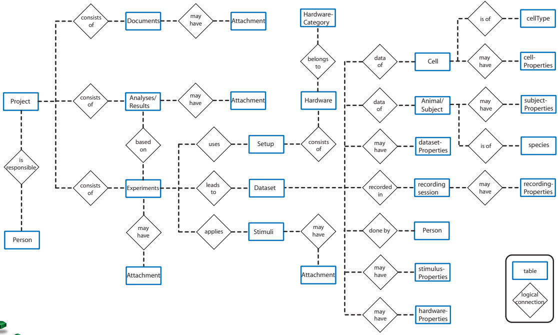
lablog.sourceforge.net

LabLog - the Laboratory Logbook

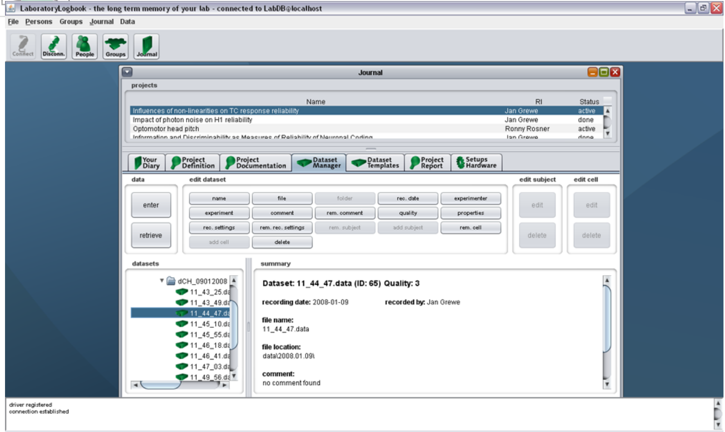
- Storage of project related information:
  - Lab-journal (ideas, diary)
  - Experimental setup and hardware
  - Projects
- Management of acquired data:
  - Storage of metadata directly from *RELACS*
  - Search for data within/across projects (by SQL queries)
  - Export search results for further analysis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

Structure

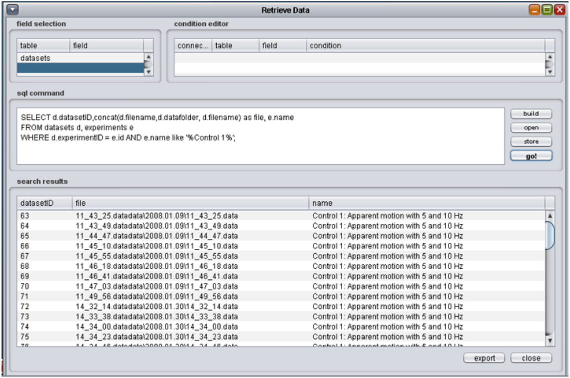
The structure of the underlying relational database (about 60 tables):



Screen Shots



Screen Shots



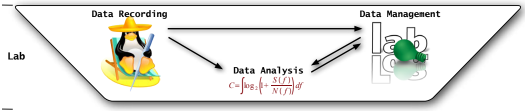
How should the metadata be imported into the database?

Part III —

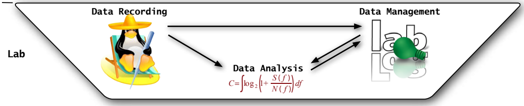
Talking about data

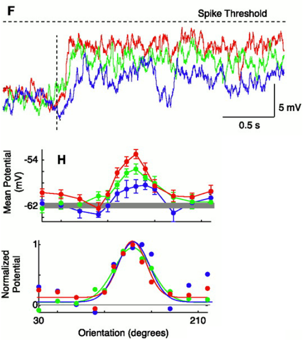
an Extensible Framework for Metadata Exchange

The Data-Chain



The Data-Chain





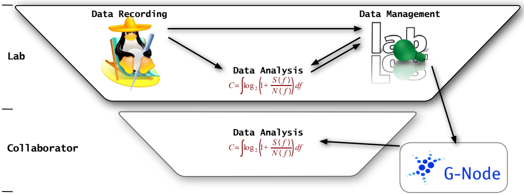
The Contribution of Noise to Contrast Invariance of Orientation Tuning in Cat Visual Cortex

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

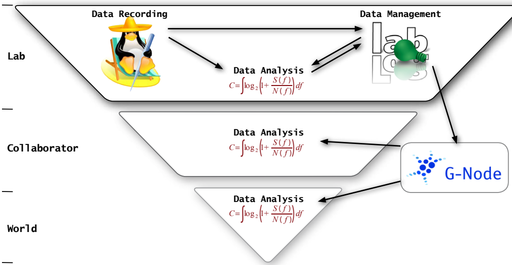
Science, 2000

relacs

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.

stimulus.type = white noise

name

value

Metadata

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

cell.baselineRate = 50 +/- 5 Hz

Metadata

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

$$\underbrace{\text{cell.baselineRate}}_{\text{name}} = \underbrace{\text{50}}_{\text{value}}\ \underbrace{\text{+/- 5}}_{\text{error}}\ \underbrace{\text{Hz}}_{\text{unit}}$$

The Metadata Problem

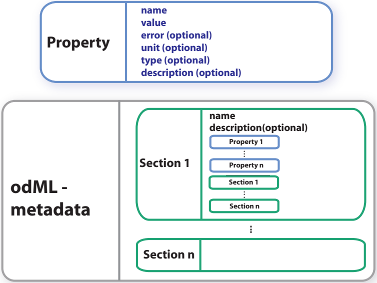
- is “data about data”.
- describe recording conditions.

cell.baselineRate = 50 +/- 5 Hz

name value error unit

- 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:

name	type	description
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:

name	type	description
AIUsedChannelCount	int	The number of used analog input channels.
AISampleRate[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".
AIPolarity[n]	string	The polarity of the measurement "unipolar" or "bipolar".

Definition — Examples

Cell:

name	type	description
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 indiator 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.

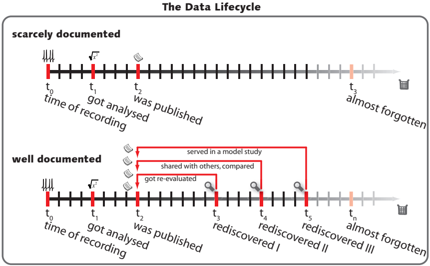
Vision

- 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.

$\Rightarrow$ You can easily share your data!

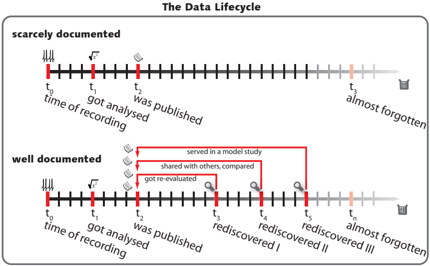
$\Rightarrow$ Your data can be found!

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

$\Rightarrow$ 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).
- *LabLog* helps you keeping track of your data (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!