Keeping track of your data with NIX: tools for comprehensive data organization

Christian J Kellner[1], Achilleas Koutsou[1], Michael Sonntag[1], Adrian Stoewer[1], Andrey Sobolev[1], Jan Benda[2], Thomas Wachtler[1], Jan Grewe[2]

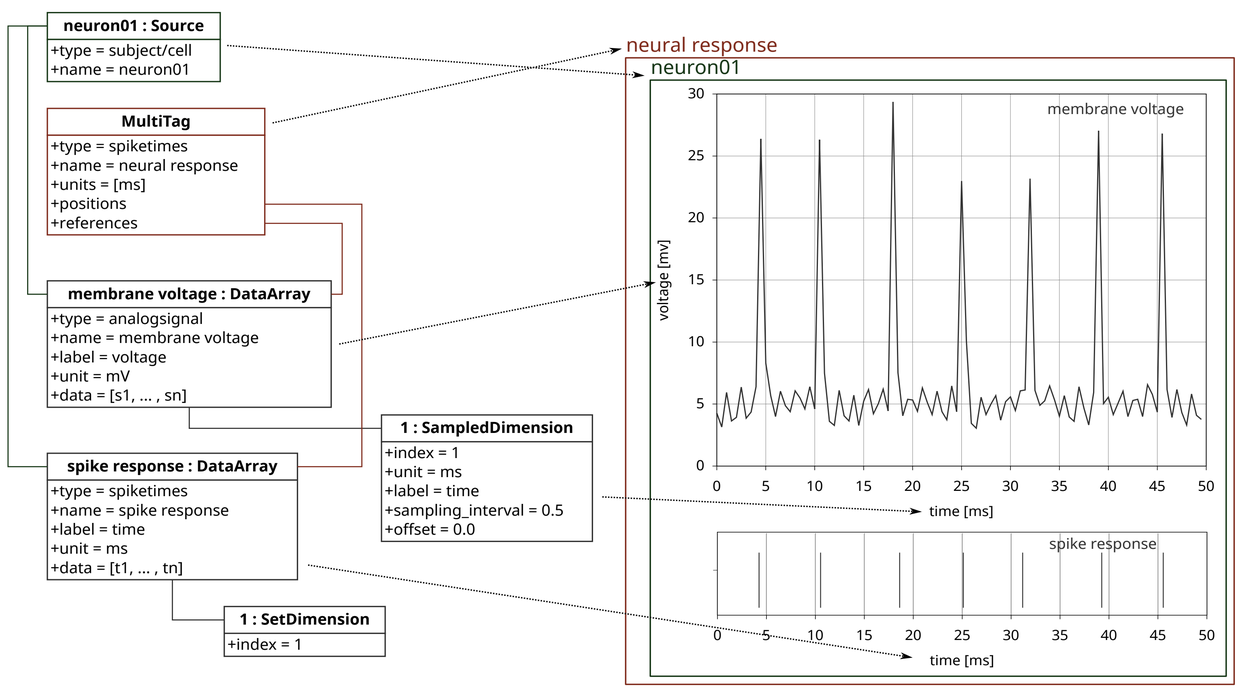
[1]German Neuroinformatics Node, Department Biologie II, Ludwig-Maximilians-Universität München, Germany;
[2]Institut für Neurobiologie, Universität Tübingen, Germany

Introduction

Managing neuroscience data requires the integration of information from multiple sources. Background information, or metadata, about the experiment is necessary to interpret the resulting data correctly. Storing such information consistently is an essential part of experimental research and depends crucially on available file formats. Many existing formats provide only limited support for storing metadata along with the data. Here we present the NIX project [1], consisting of an open format and software tools to store and organize data and metadata.

The **NIX** project specifies a versatile format for neuroscientific data. It provides libraries for accessing these files from different platforms. NIX is based on a well defined data model which can be used to represent both **data** and related **metadata**. In particular, it provides **generic entities** designed to store a wide variety of data types like **continuous signals**, **spike events**, **image stacks**, or other **multi-dimensional data**. Metadata storage is supported via adaption of the **odML** data model [2].

The NIX data model

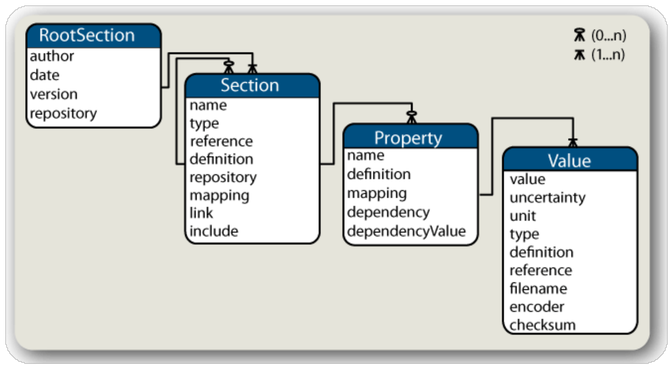


Main Entities:
- **Array**: stores n-dimensional data with information about data type and units, defines dimensions using **Dimension** entity.
- **Tag**: Defines points or regions, representing segments, spike times, events, and relationships between data.

All entities have:
- a unique **id**: allows synchronization and identification across files.
- a **name**: serves as a human readable identifier.
- a **type**: provides semantic context, domain-specificity.

The model provides all information to interpret the data correctly.

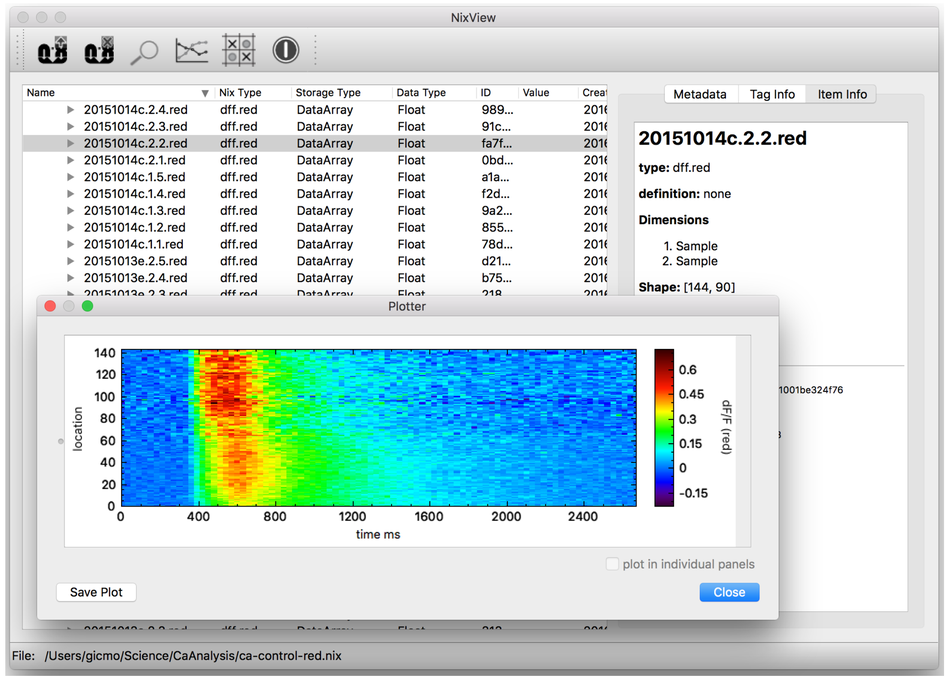
The odML model



- Metadata is stored as a hierarchically organized structure of key-value pairs.
- Any metadata can be stored, according to the specifics of the experiment or dataset.
- Linking of metadata and data enables search and selection of data based on metadata.

Any kind of metadata can be stored and can be organized to reflect the structure of the experiment.

Easy Access: NixView



- NixView [4] enables convenient exploration of data and metadata of NIX files.
- Facilitates plotting support for a large variety of raw data.
- Provides tabular display of raw data with CVS export.

Libraries and language bindings

The schema definition for HDF5 [3] represents all entities of the data model hierarchy. It was designed to be easily readable even without a special library.

Easy reading and writing of the NIX file format, even without deep knowledge about the exact format specification, is provided by an IO-library in C++ [1], supporting major compilers and operating systems such as Linux, OSX and Windows, and language bindings for Python [5], Matlab [6] and Java [7]. The Python package [5] additionally provides quick and easy installation without requiring the C++ library.

C++ python™ MATLAB Java™

NIX backend for Neo

An I/O backend for Neo [8] maps the Neo data model to the NIX format:
- Data stored in any format supported by Neo can be converted to NIX.
- Enables easy storage of data analysis done with Neo compatible tools, e.g., the Elephant [9] toolkit.
- Converting data to NIX, or using NIX as a backend for Neo, enables the use of NIX as well as HDF5 tools and viewers:
  - NIX language bindings for C++, Python, Matlab, Java.
  - NixView.
  - HDFView, h5dump, h5ls, etc.

neo

elephant
ELECTROPHYSIOLOGY ANALYSIS TOOLKIT

Resources

This work was performed in connection with the activities of the HDF5 working group of the INCF Electrophysiology Data Sharing Task Force; Supported by BMBF grant 01GQ1302.

Contact: *nix@g-node.org*

[1] https://github.com/G-Node/nix
[2] Grewe et al (2011), Frontiers in Neuroinformatics 5:16
[3] https://github.com/G-Node/nix/wiki/Implementation-in-HDF5
[4] http://bendalab.github.io/NixView/
[5] https://github.com/G-Node/nixpy
[6]https://github.com/G-Node/nix-mx
[7] https://github.com/G-Node/nix-java
[8] http://neuralensemble.org/neo/
[9] http://neuralensemble.org/elephant/

Federal Ministry of Education and Research