



HAL
open science

Neo: a Data Model for Electrophysiological Data Representation and its Python Implementation

Florent Jaillet, Samuel Garcia, Andrew P. Davison, Luc Estebanez, Pierre Yger

► **To cite this version:**

Florent Jaillet, Samuel Garcia, Andrew P. Davison, Luc Estebanez, Pierre Yger. Neo: a Data Model for Electrophysiological Data Representation and its Python Implementation. 17th Annual Meeting of the Organization for Human Brain Mapping (HBM 2011), Jun 2011, Québec, Canada. . hal-04128682

HAL Id: hal-04128682

<https://hal.science/hal-04128682>

Submitted on 14 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

NEO AND ITS OBJECTIVE

Neo is a hierarchical data model for the representation of intracellular and extracellular electrophysiology data with support for multi-electrodes (for example tetrodes) and EEG data.

The Neo project distributes an implementation of this model with a set of input/output (IO) modules for various file formats, as an open-source, multi-platform Python package under a BSD license. It is available at <http://pypi.python.org/pypi/neo>.

The objective of Neo is to facilitate the cooperation between independent software projects working on simulation, modeling, representation and analysis of electrophysiological signals in electroencephalography or intracellular and extracellular electrophysiology.

For this, Neo provides a common representation of the core data in order to improve the reusability of the developed tools and share data between different projects. It also provides a common layer for datafile and database reading and writing through its IO modules.

The core objects and IO modules provided in the Neo 0.2 implementation are described below.

BRIEF HISTORY OF THE PROJECT

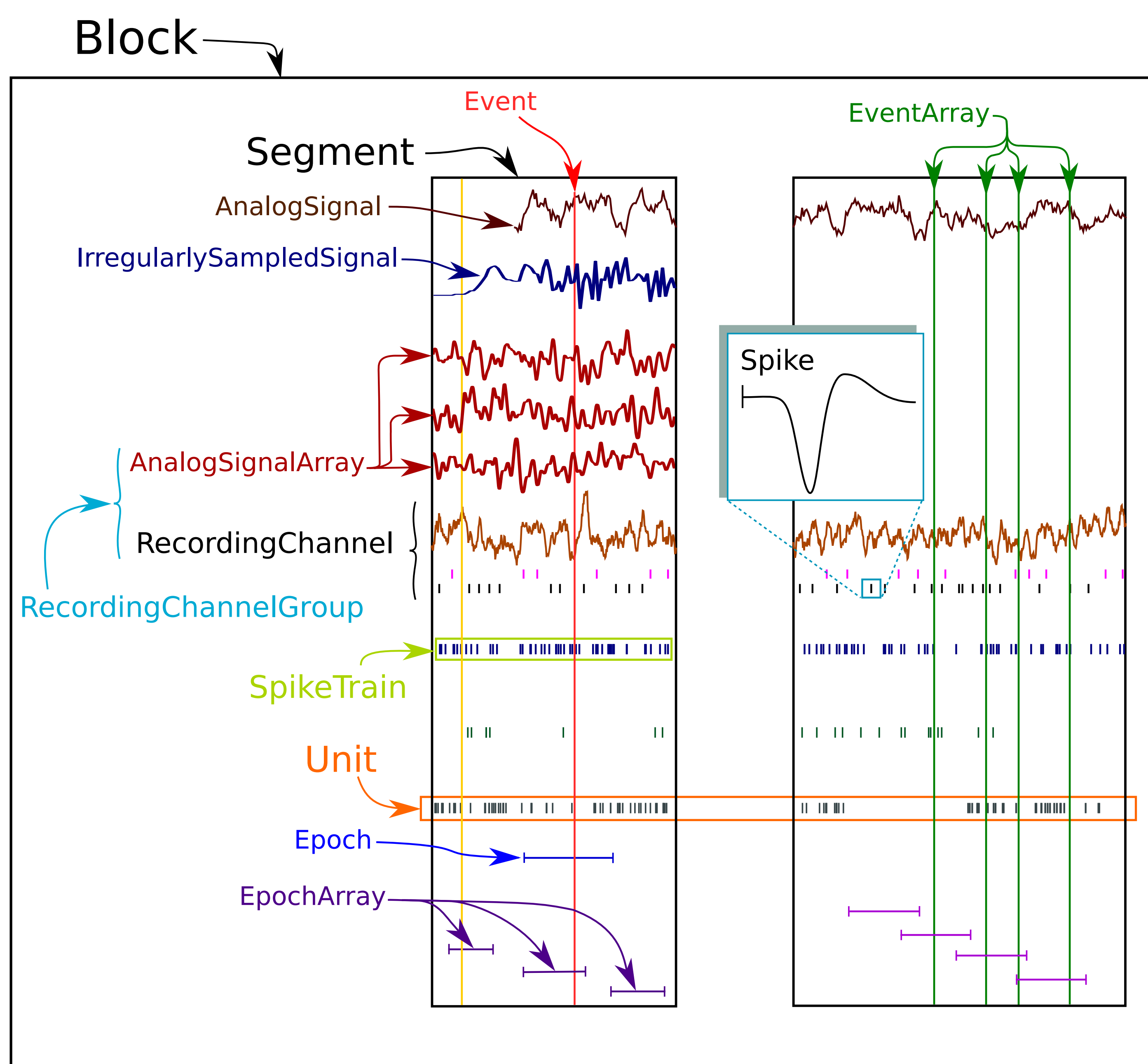
The need for a common data model appeared in the NeuralEnsemble community (neuralensemble.org). The specification process started by considering the available data models in existing NeuralEnsemble tools for processing of experimental data (OpenElectrophy, NeuroTools) and for simulation (PyNN). Other projects with related goals were also considered, in particular the Neuroshare project (neuroshare.sourceforge.net). In this process, the limits of the data models in these different projects as well as the current and future uncovered needs were identified.

After discussions between Neural Ensemble members, a first implementation in Python of Neo (version 0.1) was produced during summer 2010 as part of the rewriting from scratch of OpenElectrophy.

To respond to the needs of new projects interested in using Neo, some discussions took place in February 2011 for the specification of Neo 0.2.

The main improvements introduced in Neo 0.2 are the possibility to handle physical quantities and the possibility to represent data items individually or grouped in arrays for improved processing efficiency and better handling of multichannel data.

ARCHITECTURE



Neo 0.2 architecture

The core objects available in Neo 0.2 are:

- **Block**: Main container gathering all the data for a given setup, which can be viewed as a list of Segments.
- **Segment**: Container for heterogeneous data objects sharing a common time base, generally acquired during the same time period.
- **AnalogSignal**: A regular sampling of a continuous signal.
- **AnalogSignalArray**: A regular sampling of a multichannel continuous signal.
- **IrregularlySampledSignal**: An irregular sampling of a continuous signal.
- **RecordingChannel**: A physical location identifying the recorded data (for example the position of the electrode).
- **RecordingChannelGroup**: A set of physical locations identifying the recorded data (for example the positions of the electrodes).
- **SpikeTrain**: A set of action potential times (with optional waveforms) emitted by the same cell in a given time period.
- **Spike**: One action potential characterized by its time and waveform.
- **Unit**: A Unit gathers all the SpikeTrain objects within a common Block, across several Segments, that have been emitted by the same cell.
- **Event**: Point-like time event (triggers, stimulus,...)
- **EventArray**: A set of Events
- **Epoch**: Similar to Event but with a duration (period, state of a subject,...)
- **EpochArray**: A set of Epochs

INPUT/OUTPUT MODULES

Neo provides of a set of IO modules to convert data from/to many file formats used for multichannel recordings of electrophysiological data. The IO module API also enables access to databases for the storage of Neo objects.

Compared to other projects, Neo has the specificity to provide these converters as free and open-source pure Python code, which ensures transparent multiplatform use and upgradability.

The file formats currently supported in Neo 0.2 are: ASCII, Axon, EEGLAB, Elan, HFD5, Micromed, NeuroExplorer, Plexon, PyNN, raw, Spike2, TDT, WinEDR, WinWCP, Neuroshare (win32 only).

Some new IO modules are currently in development or planned (AlphaOmega, Brainvision, EDF, Elphy), and others will be added in the future.

PERSPECTIVES

Neo 0.2 is currently in development and will be released towards the end of this summer.

Neo is a young project which is still open to evolution and welcomes new contributions. You can join the Neo initiative by:

- Participating in the Neural Ensemble online discussion group (groups.google.com/group/neuralensemble) and shaping the evolution of Neo.
- Using Neo classes in your projects.
- Implementing IO modules for new data file formats and databases.

PROJECTS USING NEO

