The NIX file format for data and metadata: Acquire together, file together

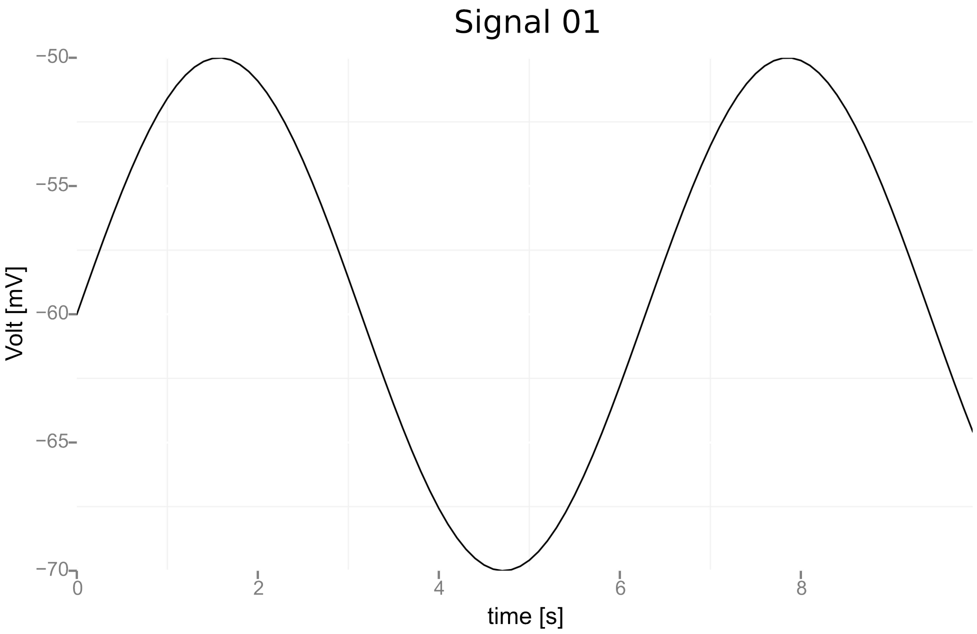
Adrian Stoewer
Christian Kellner

Introduction

- Why another file format?
  - Many formats are proprietary and not open source
  - Often poorly documented
  - Existing open source formats do not support complex metadata
- About the NIX project
  - Started at the hackathon after the 2012 INCF Congress
  - Project goals:
    - A flexible format for neuroscience data and metadata based on HDF5
    - Development of a data model that can also be used in other back-ends
    - Development of a reference implementation in C++

Talking about data:
what do we need to store?

Sampled Data

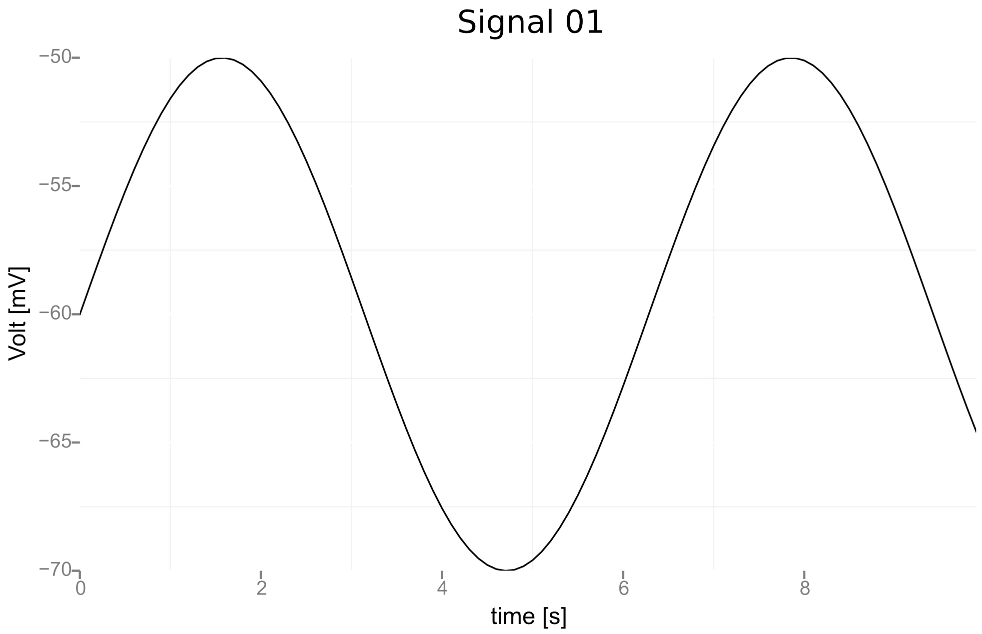


What do we need to store?

- Name
- 1D Data

  $[s_1, \ldots, s_n]$
- label (y-axis)
- unit (y-axis)
- label (x-axis)
- sampling unit (x-axis)
- sampling interval (x-axis)

Irregularly Sampled Data



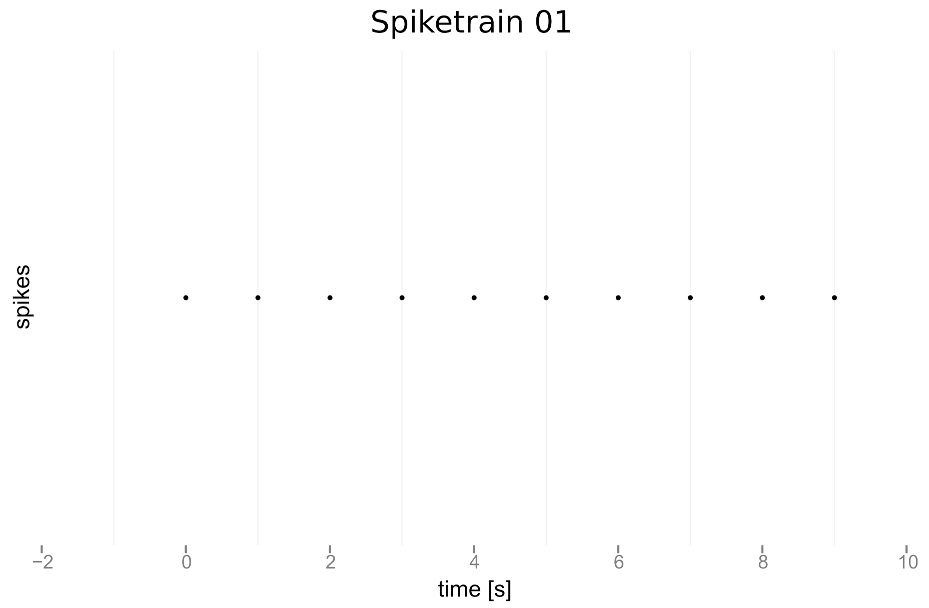
What do we need to store?

- Name
- 1D Data

  `[s₁, …, sₙ]`
- 1D Data (ticks)

  `[t₁, …, tₙ]`
- label (y-axis)
- unit (y-axis)
- label (x-axis)
- unit (x-axis)

Event Data

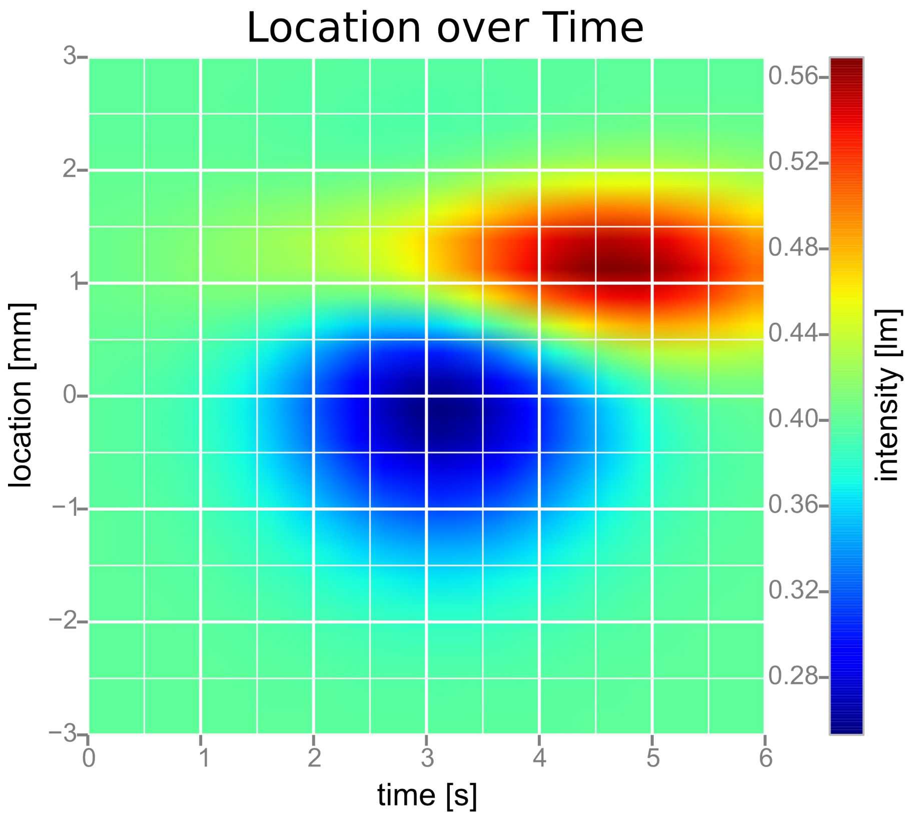


What do we need to store?

- Name
- 1D Data

  `[`e_1`, …, `e_n`]`
- label (y-axis)
- label (x-axis)
- unit (x-axis)

Multiple Dimensions



What do we need to store?

- Name
- 2D Data

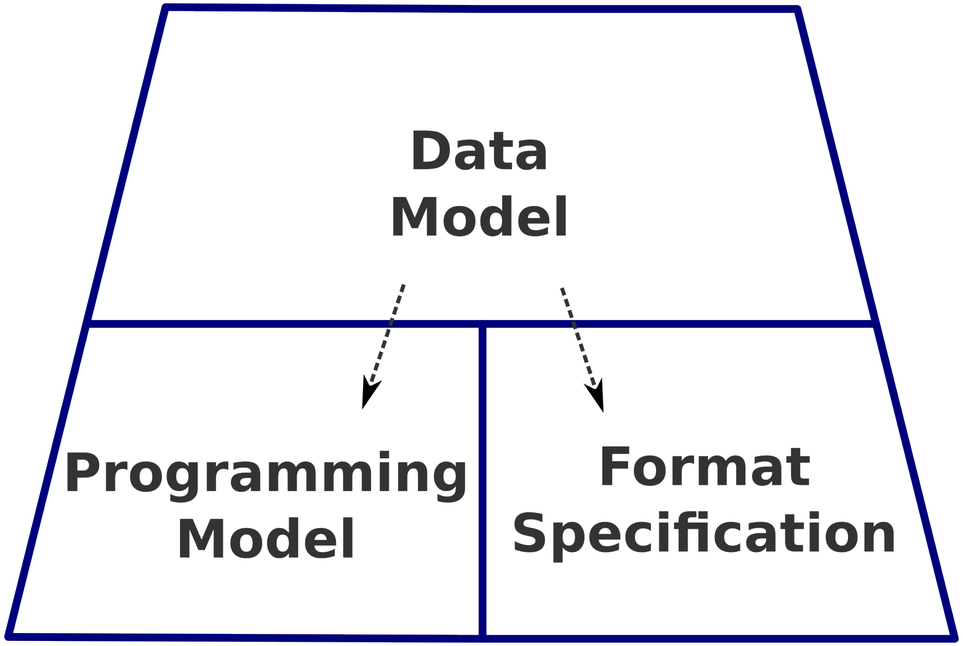
  `[[s`$_{1,1}$`, …, s`$_{n,m}$`]]`
- label (z-axis)
- unit (z-axis)
- label (x- / y-axis)
- sampling unit (x- / y-axis)
- sampling interval (x- / y-axis)

Data Models

Data Models

Why is a data model is useful?

- Allows early evaluation of different use cases
- Captures the most important concepts of the format
- Format specification can be derived from the model
- API or programming model can be derived from the model



Conventional Data Models

- Data model development process
  - Isolation of data objects the model should be able to describe
  - Define entities reflecting the data objects
  - Define properties and relationships of the entities
- Advantages
  - Model is easy to understand
  - Entities have a strong semantic meaning
- Disadvantages
  - Can result in a very large number of entities

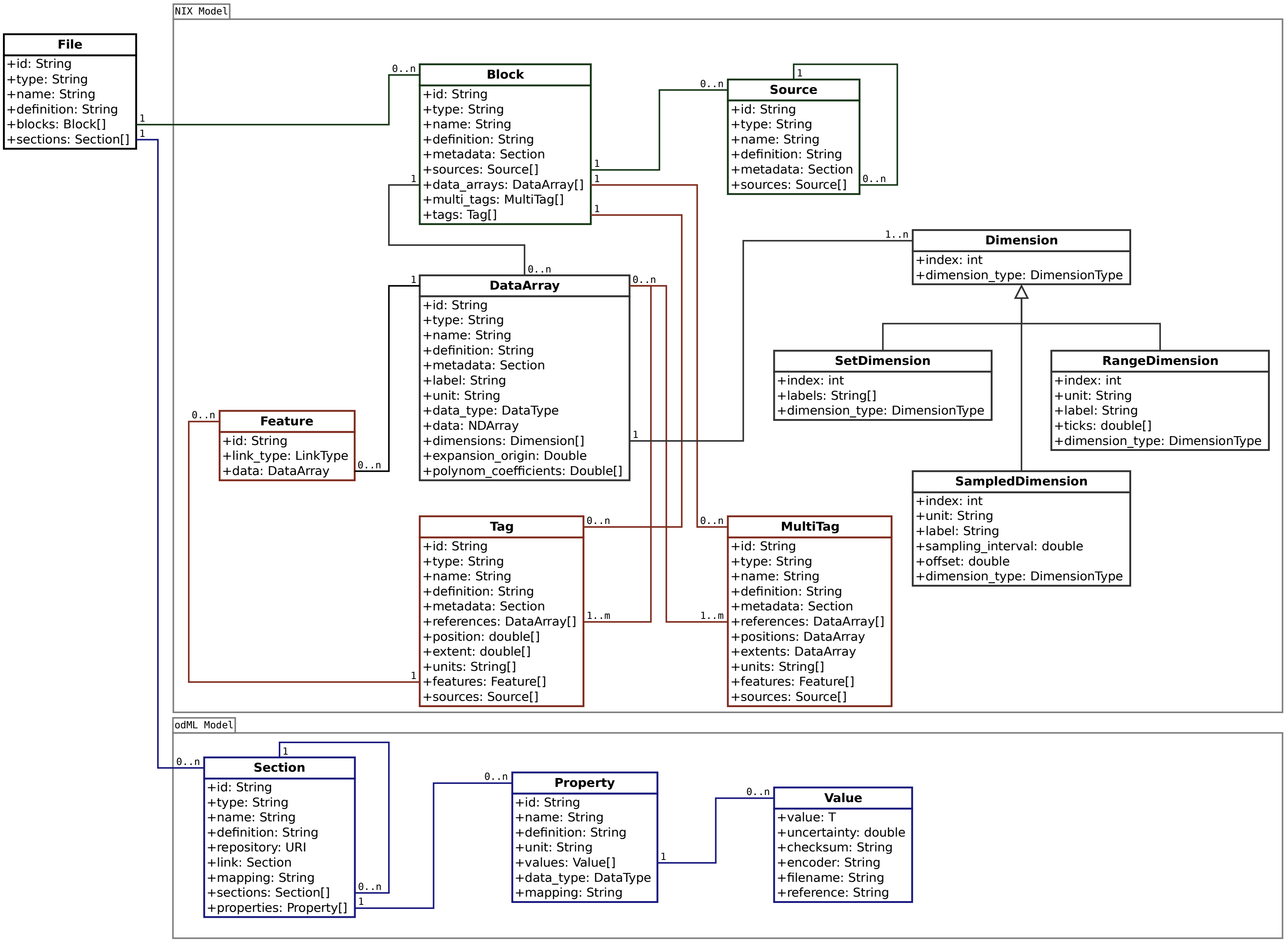
Generic Data Models

- Data model development process
  - Isolation of data objects the model should be able to describe
  - Identify common properties and concepts of data objects
  - Design generic entities that can represent those common concepts
  - Introduce generic properties that can describe differences between data objects
- Advantages
  - Model is much smaller
- Disadvantages
  - May be harder to understand and to process
  - Entities have less semantic meaning

The NIX approach

- Model for data
  - Take entities from existing formats such as NEO
  - Extract common concepts from these entities
  - Generalize the concepts as far as possible
- Model for metadata
  - Use the odML data model
  - Connect both data models

NIX Model: Overview



File
+id: String
+type: String
+name: String
+definition: String
+blocks: Block[]
+sections: Section[]

Block
+id: String
+type: String
+name: String
+definition: String
+metadata: Section
+sources: Source[]
+data_arrays: DataArray[]
+multi_tags: MultiTag[]
+tags: Tag[]

Source
+id: String
+type: String
+name: String
+definition: String
+metadata: Section
+sources: Source[]

Dimension
+index: int
+dimension_type: DimensionType

DataArray
+id: String
+type: String
+name: String
+definition: String
+metadata: Section
+label: String
+unit: String
+data_type: DataType
+data: NDArray
+dimensions: Dimension[]
+expansion_origin: Double
+polynom_coefficients: Double[]

SetDimension
+index: int
+labels: String[]
+dimension_type: DimensionType

RangeDimension
+index: int
+unit: String
+label: String
+ticks: double[]
+dimension_type: DimensionType

Feature
+id: String
+link_type: LinkType
+data: DataArray

SampledDimension
+index: int
+unit: String
+label: String
+sampling_interval: double
+offset: double
+dimension_type: DimensionType

Tag
+id: String
+type: String
+name: String
+definition: String
+metadata: Section
+references: DataArray[]
+position: double[]
+extent: double[]
+units: String[]
+features: Feature[]
+sources: Source[]

MultiTag
+id: String
+type: String
+name: String
+definition: String
+metadata: Section
+references: DataArray[]
+positions: DataArray
+extents: DataArray
+units: String[]
+features: Feature[]
+sources: Source[]

Section
+id: String
+type: String
+name: String
+definition: String
+repository: URI
+link: Section
+mapping: String
+sections: Section[]
+properties: Property[]

Property
+id: String
+name: String
+definition: String
+unit: String
+values: Value[]
+data_type: DataType
+mapping: String

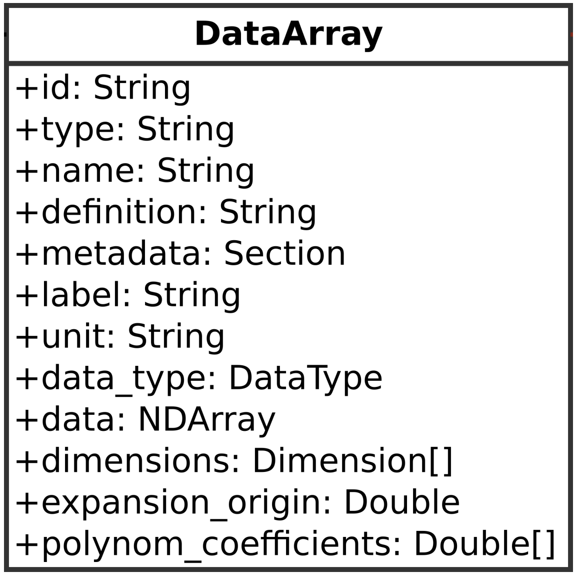
Value
+value: T
+uncertainty: double
+checksum: String
+encoder: String
+filename: String
+reference: String

NIX Model: Common Entity Properties

- id:
  - Randomly generated uuid
  - Low collision probability
- name:
  - Human readable user defined identifier
- type:
  - Provides domain specificity to the generic entities
  - The type may be defined in a terminology or ontology
- definition:
  - User defined description of the entity

NIX Model: DataArray

- Main entity for storing data
- Stores data in n-dimensional array
- Provides unit, data type and label for the stored values
- Has a dimension descriptor for each dimension of the data array
- Additional metadata via link to an odML section
- Value scaling

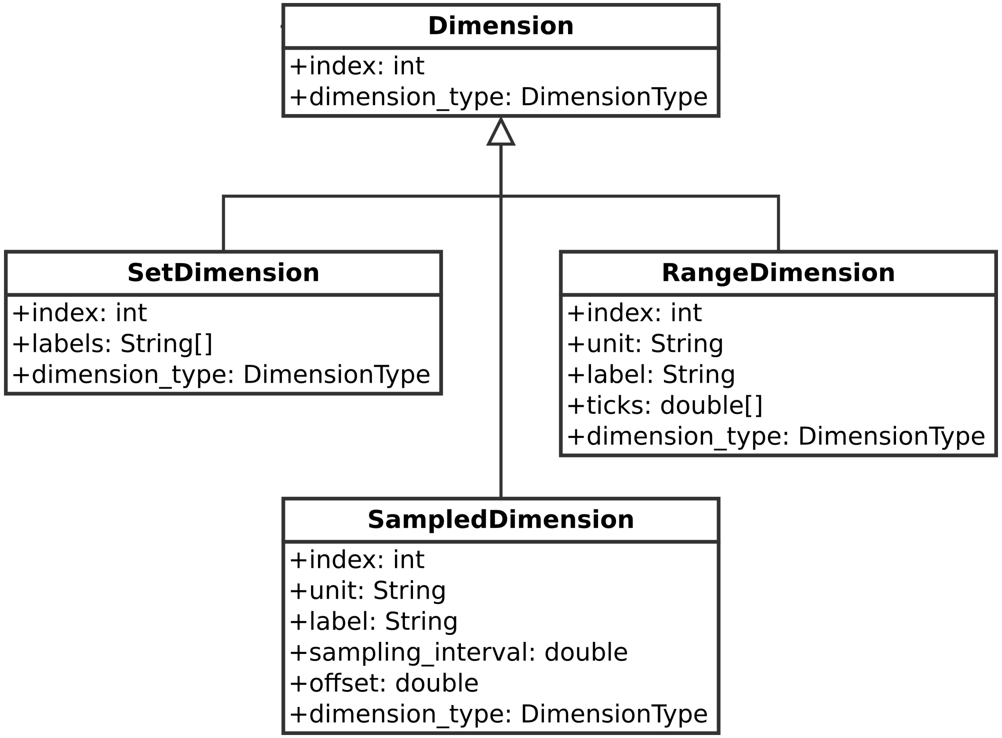


DataArray

+id: String
+type: String
+name: String
+definition: String
+metadata: Section
+label: String
+unit: String
+data_type: DataType
+data: NDArray
+dimensions: Dimension[]
+expansion_origin: Double
+polynom_coefficients: Double[]

NIX Model: Dimensions

- Describe the dimensions of data in a DataArray entity
- Three different kinds of dimensions
  - Sampled
  - Range
  - Set



Dimension

+index: int
+dimension_type: DimensionType

SetDimension

+index: int
+labels: String[]
+dimension_type: DimensionType

RangeDimension

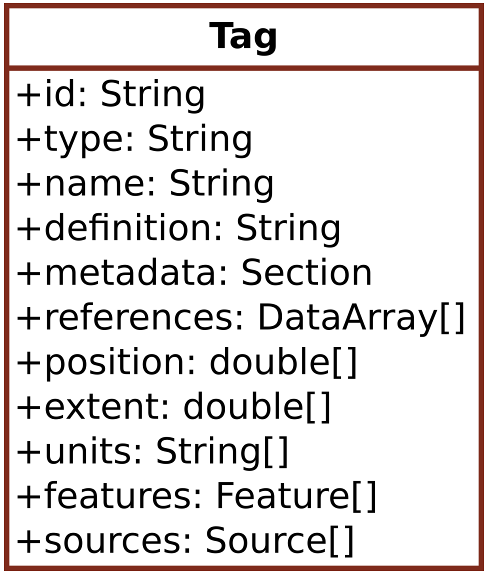
+index: int
+unit: String
+label: String
+ticks: double[]
+dimension_type: DimensionType

SampledDimension

+index: int
+unit: String
+label: String
+sampling_interval: double
+offset: double
+dimension_type: DimensionType

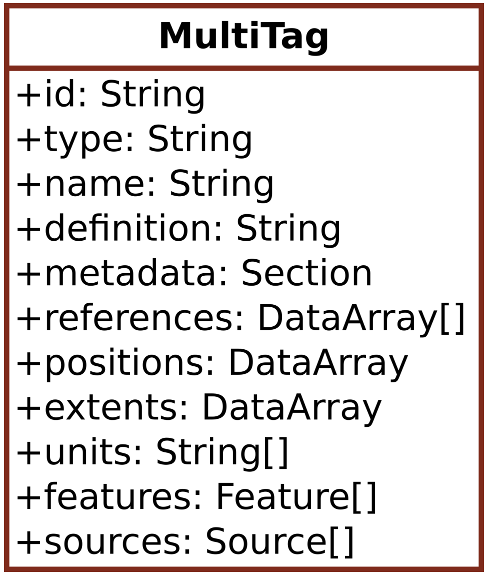
NIX Model: Tags

- Can define points or regions of interest
- Examples
  - Events
  - Spikes
  - Epochs
- MultiTag can use DataArray entities to define positions and extents



Tag

+id: String
+type: String
+name: String
+definition: String
+metadata: Section
+references: DataArray[]
+position: double[]
+extent: double[]
+units: String[]
+features: Feature[]
+sources: Source[]

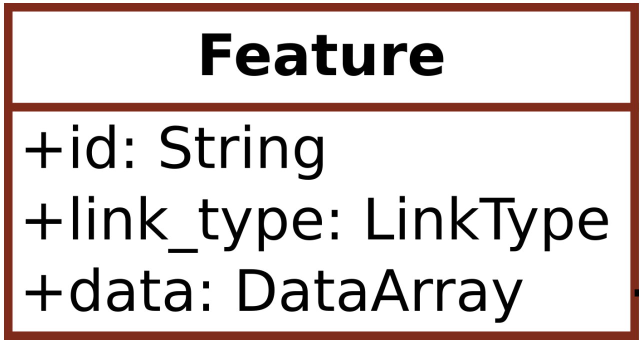


MultiTag

+id: String
+type: String
+name: String
+definition: String
+metadata: Section
+references: DataArray[]
+positions: DataArray
+extents: DataArray
+units: String[]
+features: Feature[]
+sources: Source[]

NIX Model: Feature

- Used to attach additional data to a tag entity
- Example: attach a waveform to a spike

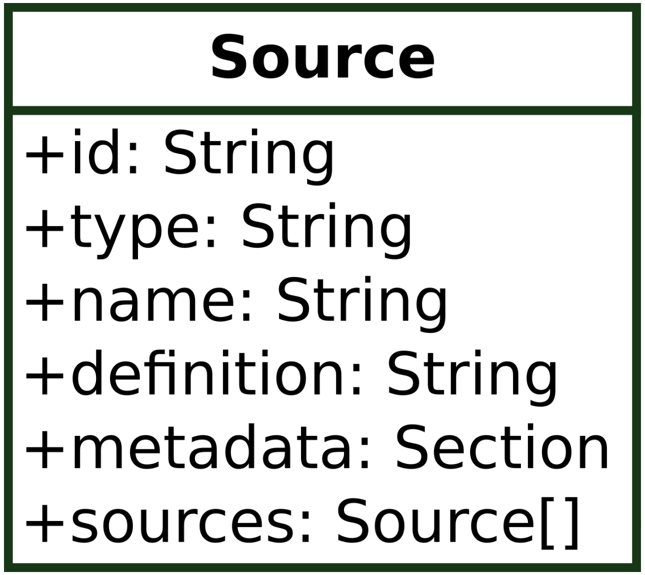


Feature

+id: String
+link_type: LinkType
+data: DataArray

NIX Model: Source

- Used to describe the provenance of other entities
- Can have child sources
- Example
  - Recording Channel
  - Electrode
  - Analysis

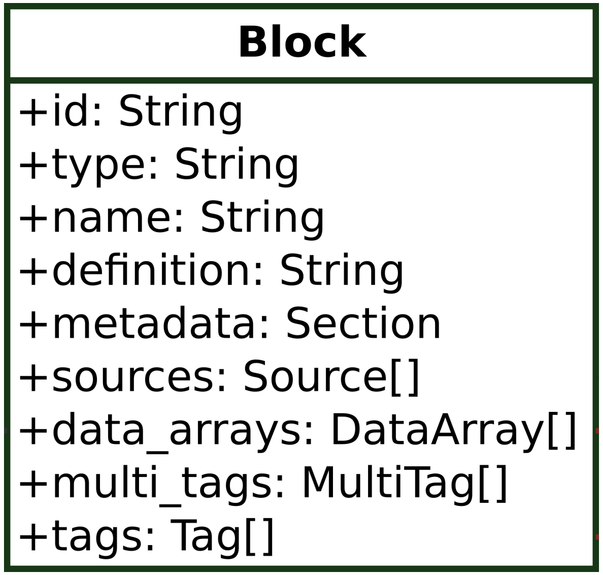


Source

+id: String
+type: String
+name: String
+definition: String
+metadata: Section
+sources: Source[]

NIX Model: Block

- Group other entities such as sources, tags and data arrays

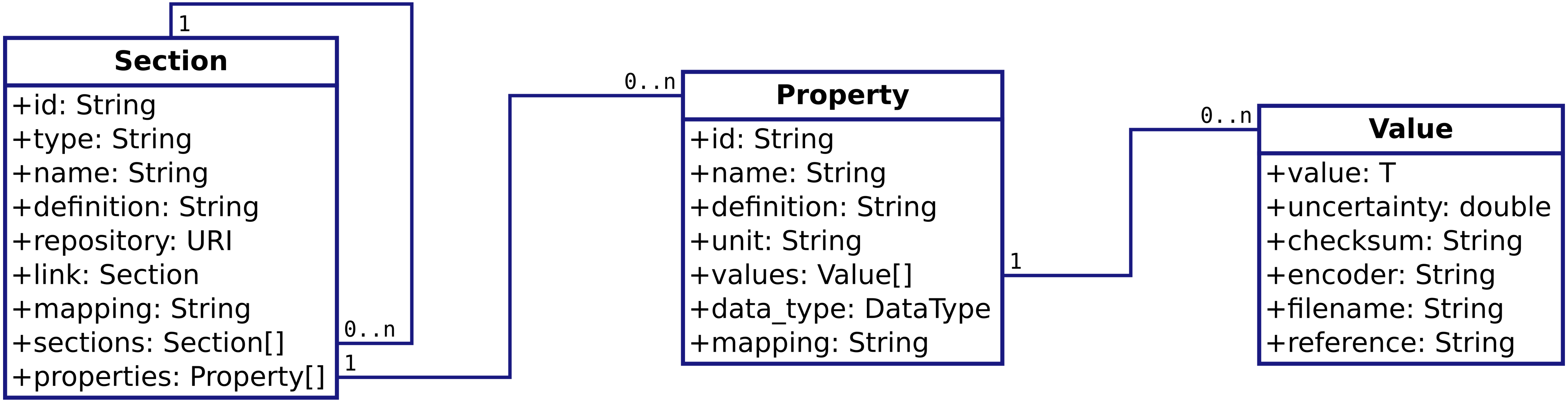


Block

+id: String
+type: String
+name: String
+definition: String
+metadata: Section
+sources: Source[]
+data_arrays: DataArray[]
+multi_tags: MultiTag[]
+tags: Tag[]

odML Model

- Flexible model for metadata (Grewe et al. 2011)
- Stores hierarchically grouped key value pairs



Section

+id: String
+type: String
+name: String
+definition: String
+repository: URI
+link: Section
+mapping: String
+sections: Section[]
+properties: Property[]

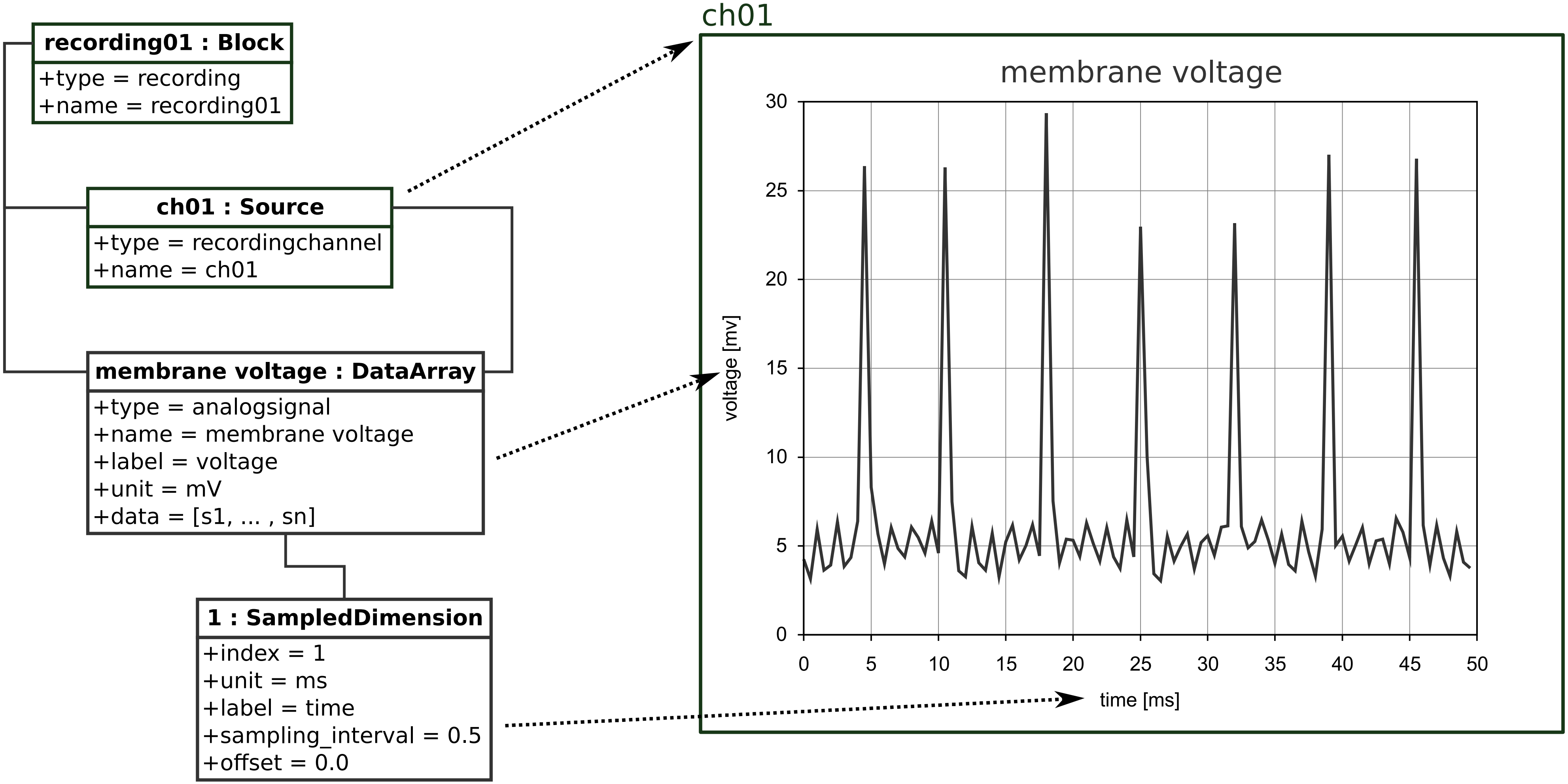
Property

+id: String
+name: String
+definition: String
+unit: String
+values: Value[]
+data_type: DataType
+mapping: String

Value

+value: T
+uncertainty: double
+checksum: String
+encoder: String
+filename: String
+reference: String

Example: Analog Signal



recording01 : Block

+type = recording
+name = recording01

ch01 : Source

+type = recordingchannel
+name = ch01

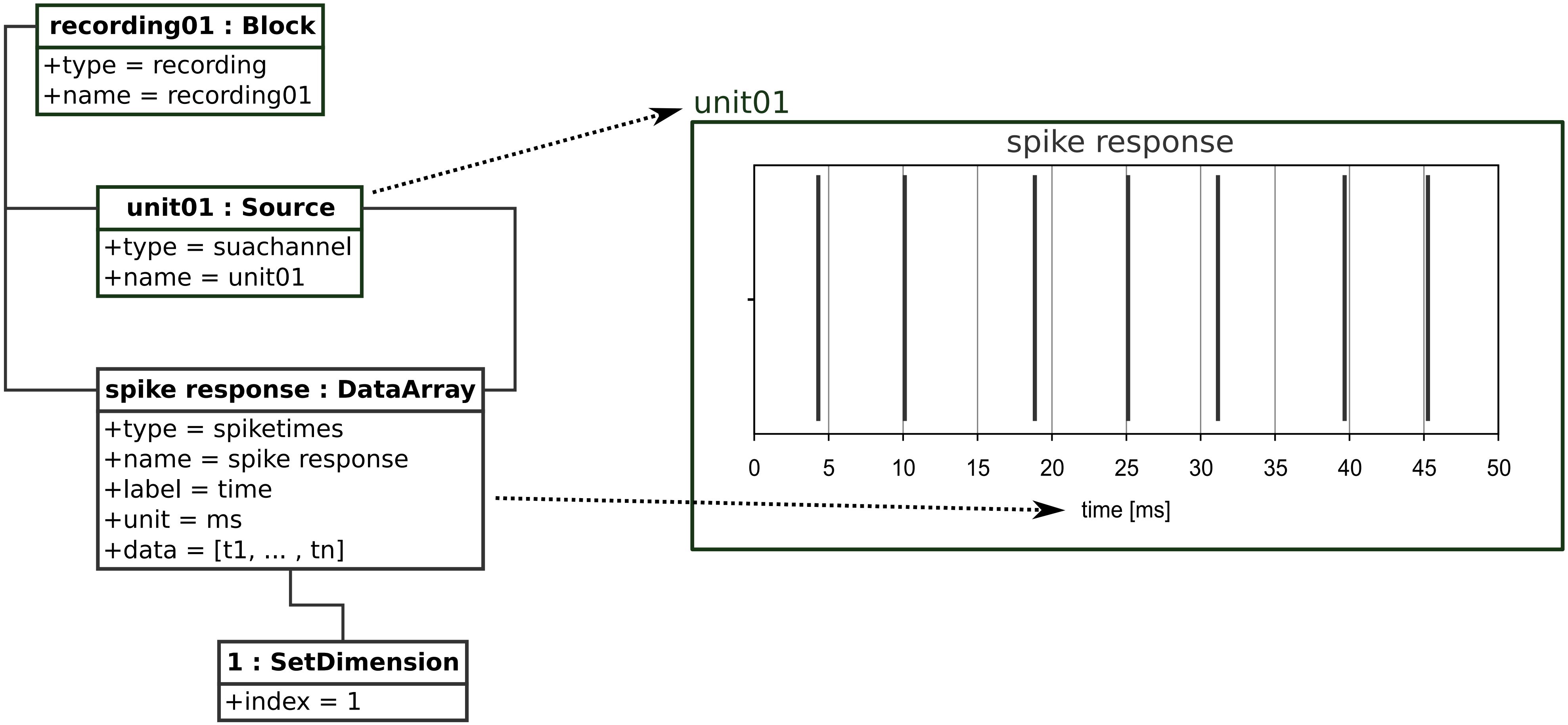
membrane voltage : DataArray

+type = analogsignal
+name = membrane voltage
+label = voltage
+unit = mV
+data = [s1, ... , sn]

1 : SampledDimension

+index = 1
+unit = ms
+label = time
+sampling_interval = 0.5
+offset = 0.0

Example: Spikes



recording01 : Block

+type = recording
+name = recording01

unit01 : Source

+type = suachannel
+name = unit01

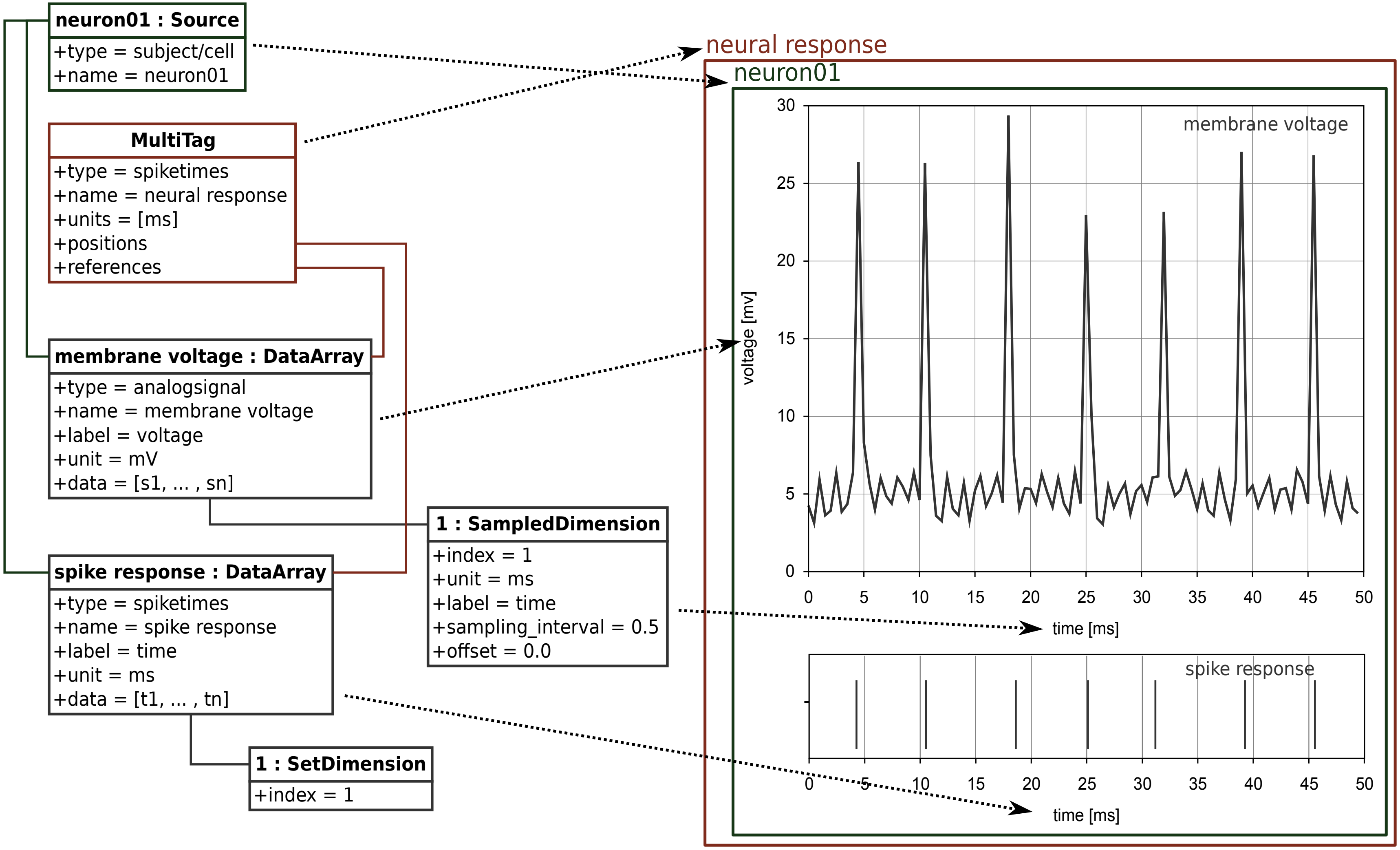
spike response : DataArray

+type = spiketimes
+name = spike response
+label = time
+unit = ms
+data = [t1, ... , tn]

1 : SetDimension

+index = 1

Example: Signal and Spikes



neuron01 : Source
+type = subject/cell
+name = neuron01

MultiTag
+type = spiketimes
+name = neural response
+units = [ms]
+positions
+references

membrane voltage : DataArray
+type = analogsignal
+name = membrane voltage
+label = voltage
+unit = mV
+data = [s1, ... , sn]

1 : SampledDimension
+index = 1
+unit = ms
+label = time
+sampling_interval = 0.5
+offset = 0.0

spike response : DataArray
+type = spiketimes
+name = spike response
+label = time
+unit = ms
+data = [t1, ... , tn]

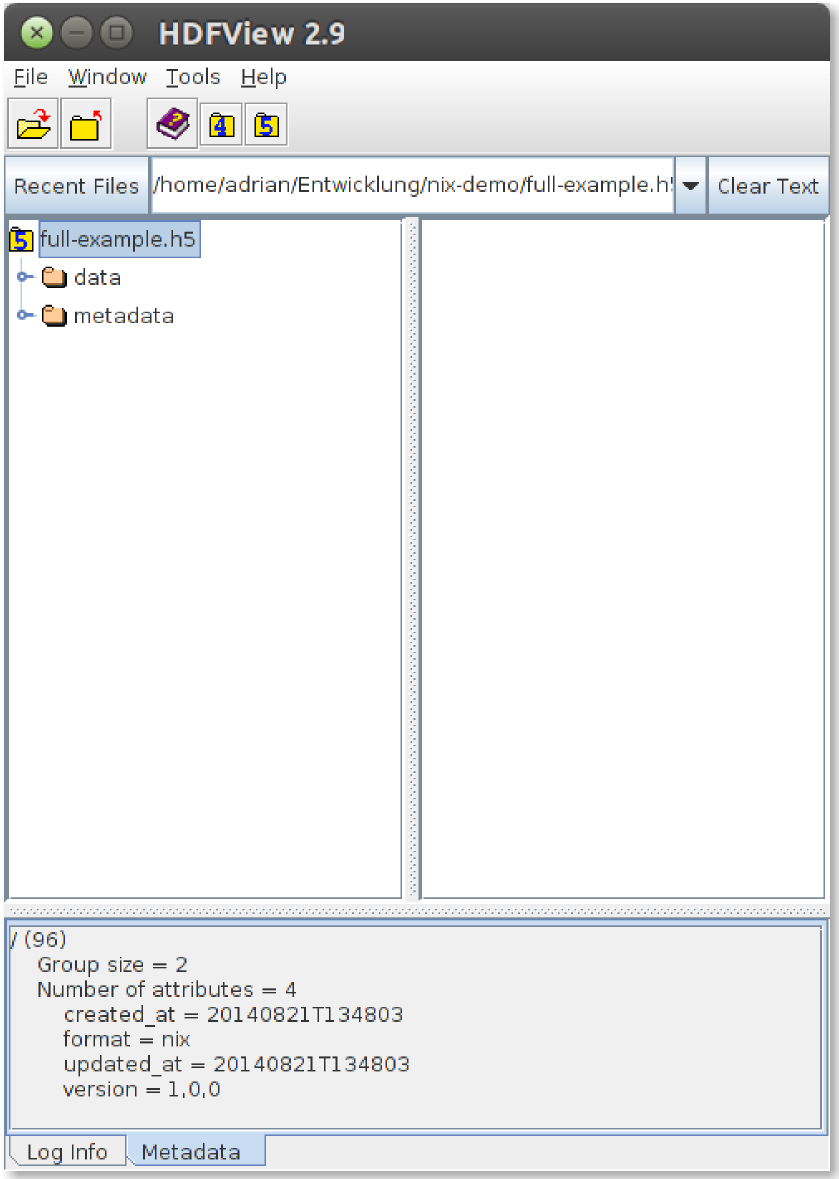
1 : SetDimension
+index = 1

HDF5 Format: General Concepts

- Most entities are implemented as HDF5 groups
  - The entity name is the group name
- Each entity type resides at a specific location in the file
- Entity properties are HDF5 attributes
- Relationships are implemented as nested subgroups or as hard links to other groups

HDF5 Format: File Root

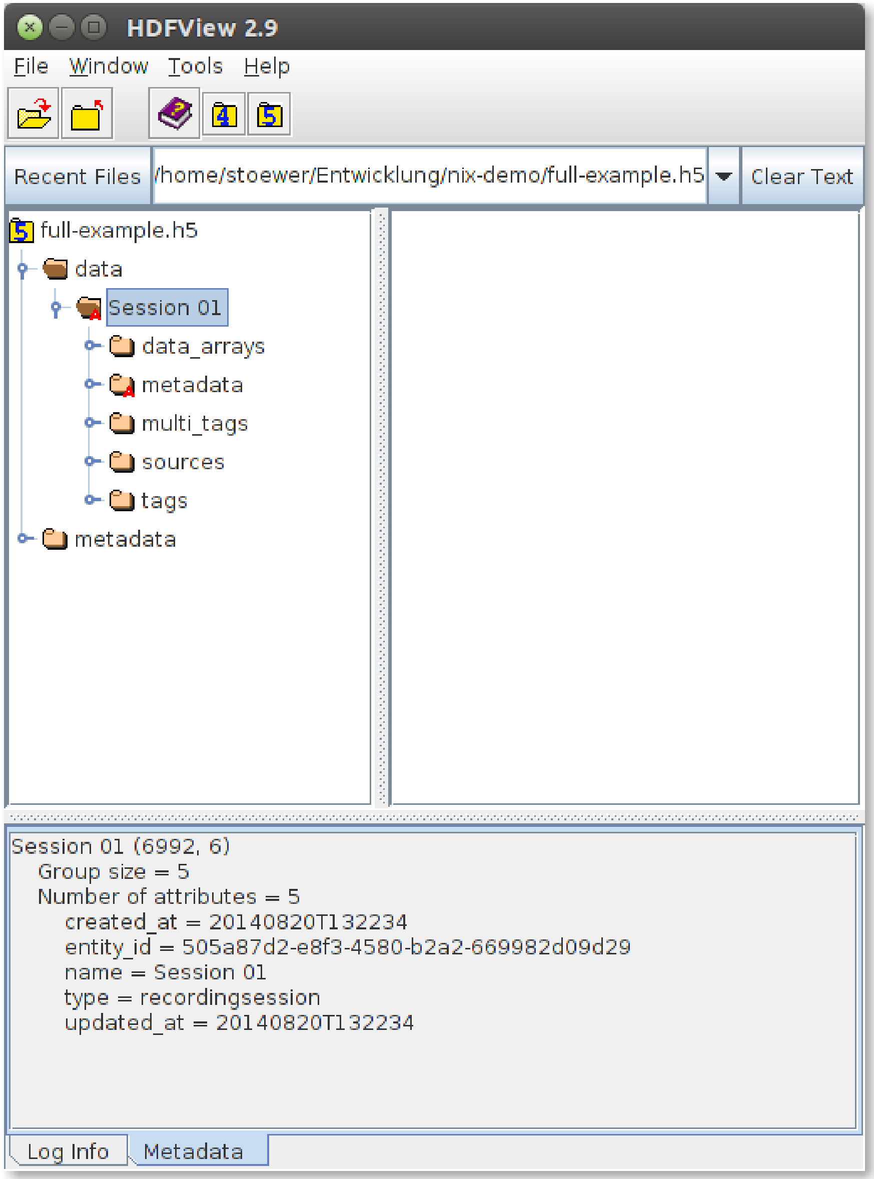
- File root attributes
  - Format hint
  - Format version
  - Update and creation time
- Subgroup *metadata*
  - Contains all odML root sections
- Subgroup *data*
  - Contains block entities



```
/ (96)
   Group size = 2
   Number of attributes = 4
      created_at = 20140821T134803
      format = nix
      updated_at = 20140821T134803
      version = 1,0,0
```

HDF5 Format: Block

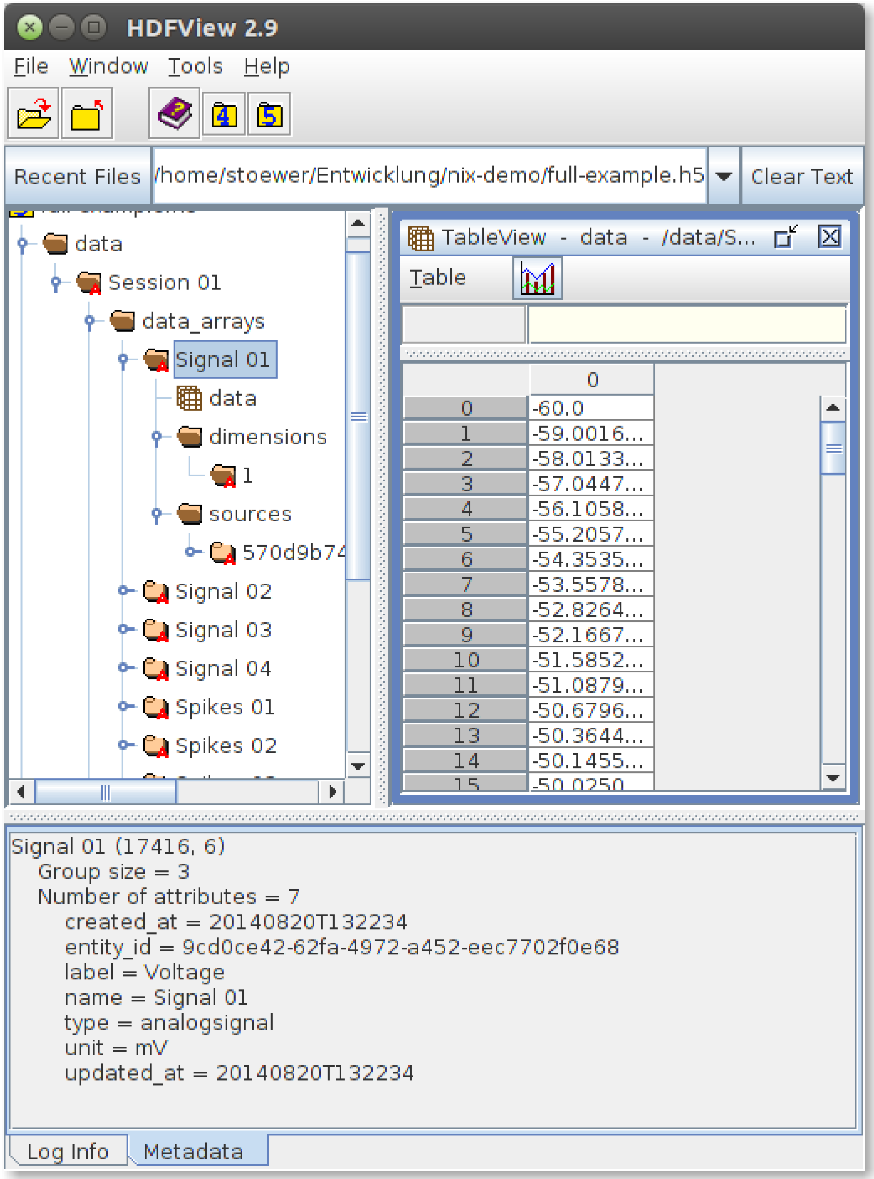
- Subgroups for:
  - DataArrays
  - Tags
  - MultiTags
  - Sources



```
Session 01 (6992, 6)
    Group size = 5
    Number of attributes = 5
        created_at = 20140820T132234
        entity_id = 505a87d2-e8f3-4580-b2a2-669982d09d29
        name = Session 01
        type = recordingsession
        updated_at = 20140820T132234
```

HDF5 Format: DataArray

- HDF5 dataset for the actual data
- Subgroup for dimension descriptors
- Subgroup for links to Source entities


HDFView 2.9
File Window Tools Help
Recent Files
/home/stoewer/Entwicklung/nix-demo/full-example.h5
Clear Text
TableView - data - /data/S...
Table
Log Info
Metadata

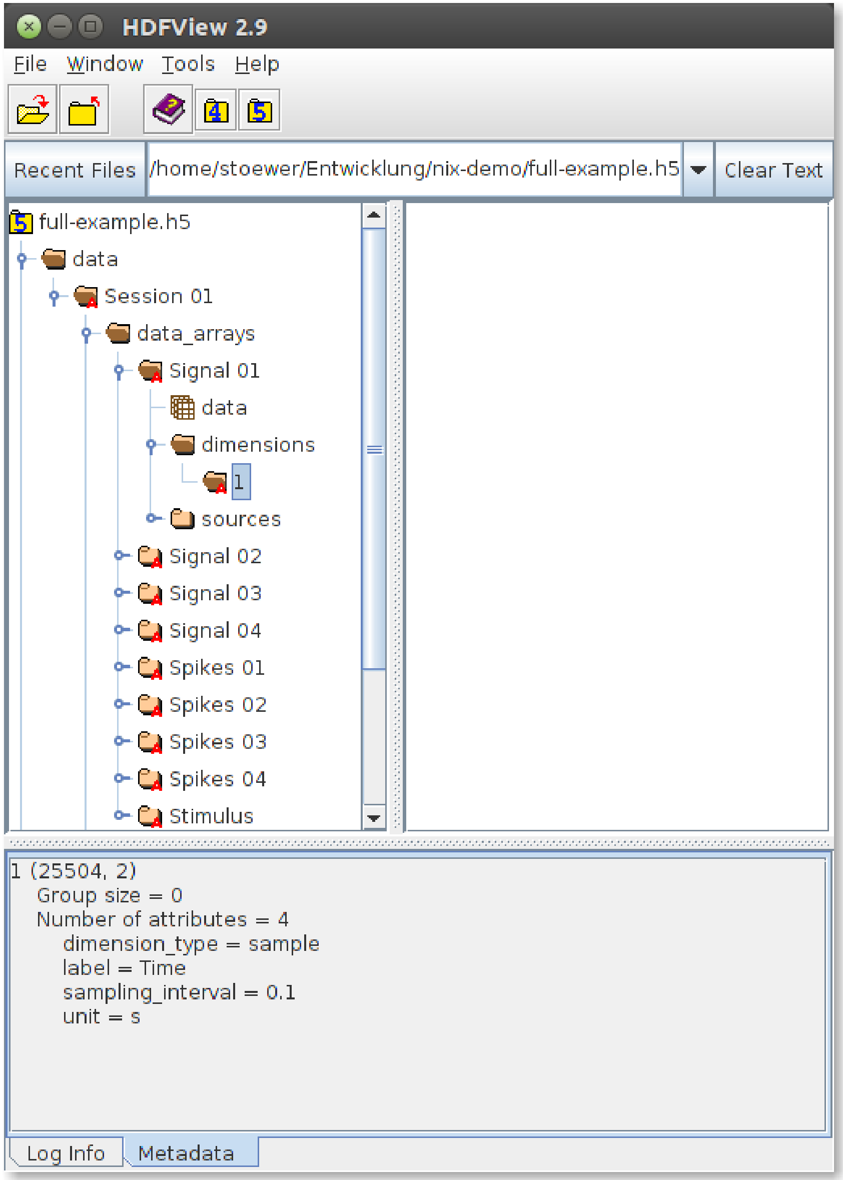
```
data
  Session 01
    data_arrays
      Signal 01
        data
        dimensions
          1
        sources
          570d9b74
      Signal 02
      Signal 03
      Signal 04
      Spikes 01
      Spikes 02
```

	0
0	-60.0
1	-59.0016...
2	-58.0133...
3	-57.0447...
4	-56.1058...
5	-55.2057...
6	-54.3535...
7	-53.5578...
8	-52.8264...
9	-52.1667...
10	-51.5852...
11	-51.0879...
12	-50.6796...
13	-50.3644...
14	-50.1455...
15	-50.0250

```
Signal 01 (17416, 6)
    Group size = 3
    Number of attributes = 7
        created_at = 20140820T132234
        entity_id = 9cd0ce42-62fa-4972-a452-eec7702f0e68
        label = Voltage
        name = Signal 01
        type = analogsignal
        unit = mV
        updated_at = 20140820T132234
```

HDF5 Format: Dimensions

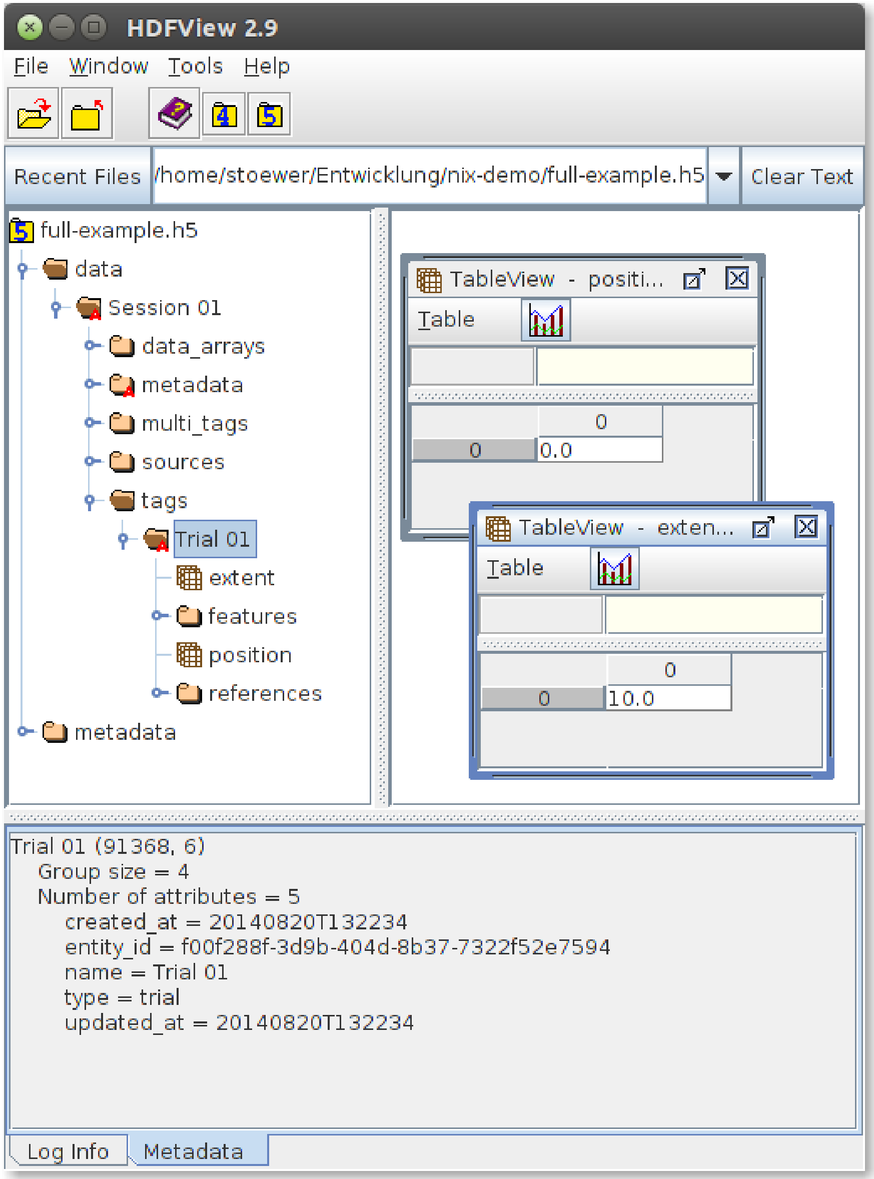
- The group name is the dimension index
- RangeDimension entities contain an HDF5 dataset for the ticks



```
1 (25504, 2)
    Group size = 0
    Number of attributes = 4
        dimension_type = sample
        label = Time
        sampling_interval = 0.1
        unit = s
```

HDF5 Format: Tag

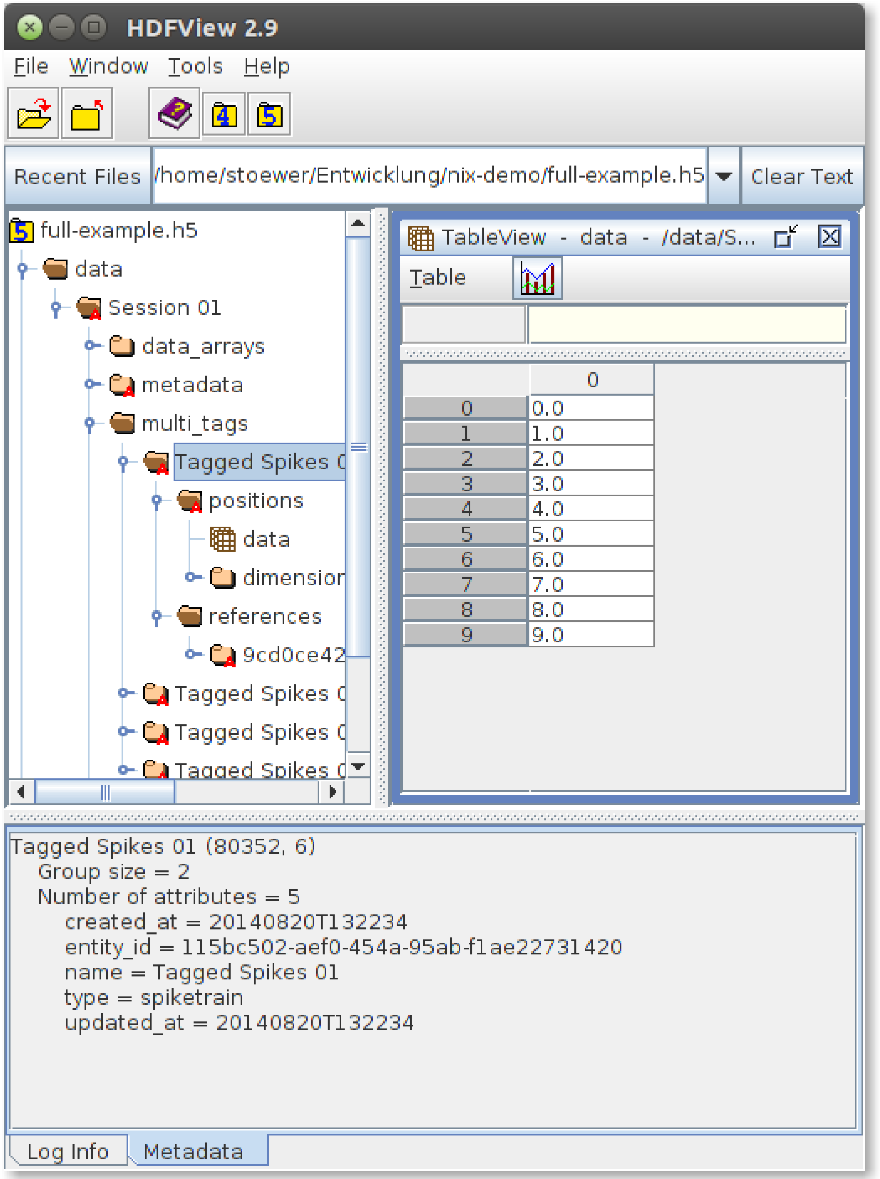
- HDF5 datasets for the position and extent vectors
- Subgroup for attached features
- Subgroup for links to referenced data



```
Trial 01 (91368, 6)
    Group size = 4
    Number of attributes = 5
        created_at = 20140820T132234
        entity_id = f00f288f-3d9b-404d-8b37-7322f52e7594
        name = Trial 01
        type = trial
        updated_at = 20140820T132234
```

HDF5 Format: MultiTag

- Link to DataArray entities representing the positions and extents

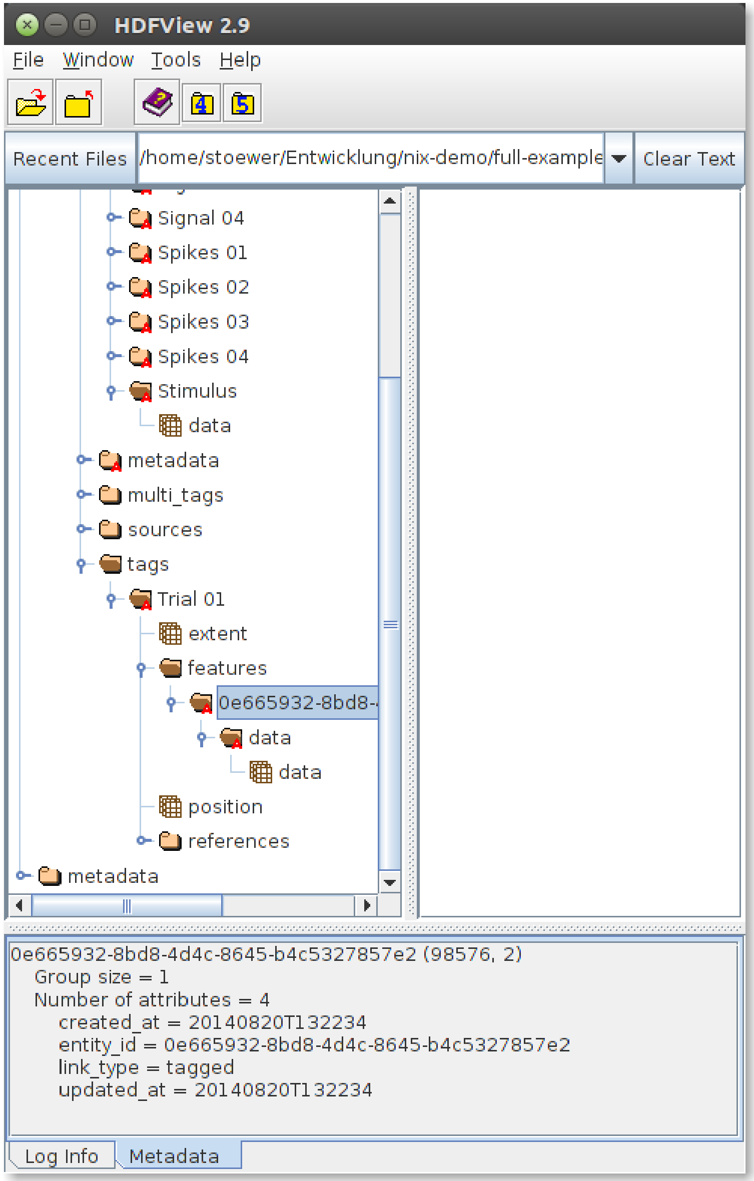


	0
0	0.0
1	1.0
2	2.0
3	3.0
4	4.0
5	5.0
6	6.0
7	7.0
8	8.0
9	9.0

```
Tagged Spikes 01 (80352, 6)
   Group size = 2
   Number of attributes = 5
      created_at = 20140820T132234
      entity_id = 115bc502-aef0-454a-95ab-f1ae22731420
      name = Tagged Spikes 01
      type = spiketrain
      updated_at = 20140820T132234
```

HDF5 Format: Feature

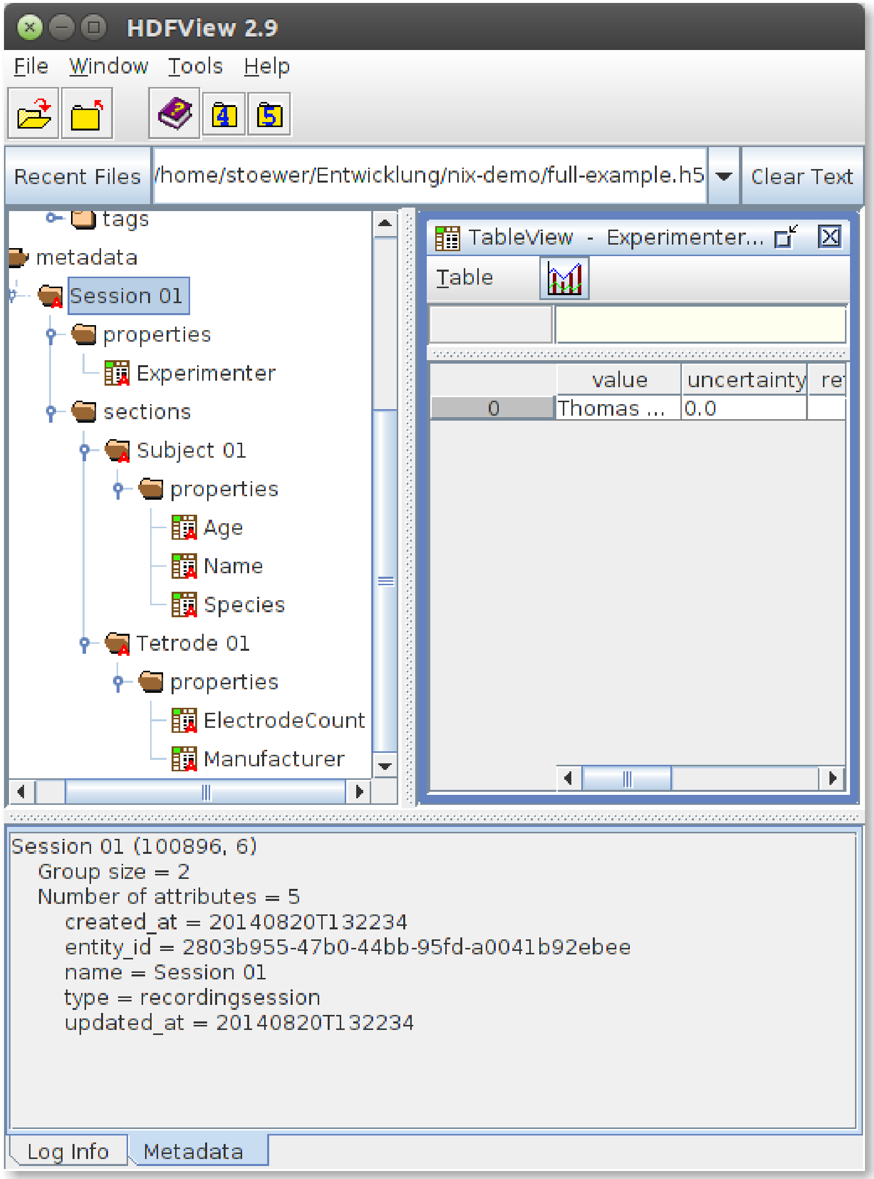
- Link to the attached DataArray entity



```
0e665932-8bd8-4d4c-8645-b4c5327857e2 (98576, 2)
   Group size = 1
   Number of attributes = 4
      created_at = 20140820T132234
      entity_id = 0e665932-8bd8-4d4c-8645-b4c5327857e2
      link_type = tagged
      updated_at = 20140820T132234
```

HDF5 Format: Metadata

- Subgroup for child sections
- Subgroup for properties
- Properties and values are implemented as HDF5 datasets



```
Session 01 (100896, 6)
    Group size = 2
    Number of attributes = 5
        created_at = 20140820T132234
        entity_id = 2803b955-47b0-44bb-95fd-a0041b92ebee
        name = Session 01
        type = recordingsession
        updated_at = 20140820T132234
```

C++ API

- Reflects directly the concepts of the data model
- Multi compiler and multi platform
  - g++, clang, MSVC
  - Linux, OSX, Windows
- Designed to support multiple back-ends
- HDF5 back-end
  - Classes operate directly on the file
  - Allows partial reading and writing of data
- Memory back-end
  - Planned
- Hosted on GitHub
  - https://github.com/G-Node/nix

Python API

- Implemented as bindings to the native C++ API
- Numpy support
- Pythonic interface
  - Reading and writing works similar to h5py
- Also hosted on GitHub
  - https://github.com/G-Node/nixpy

License

- NIX is open source
- Published under BSD License
  - Allows use and modification of the source code
  - Can be used in commercial non open source products
  - Even modified versions of NIX can be distributed without source code

Project Status

- File format is stable
- API is stable
- Beta 1 release in preparation
  - Packages for Debian via Launchpad

    https://launchpad.net/~gnode/+archive/ubuntu/nix
  - Python package on PyPi
  - Compiled version for Windows

Tools using NIX

- RELACS
  - A user interface for data aquisition
  - It is controllable and customizable to specific experimental setup by C++ based plugins
- Guppy
  - A small video recorder tool
  - Stores videos grabbed from open CV devices either as avi or to the NIX file format

Outlook

- Improve model, format and API
- Implement other back-ends
  - In-memory back-end
- More language bindings
  - Java
  - Matlab
- Think about RDF support
  - Metadata could be stored directly as triples
  - NIX type could be interpreted as rdf:type
- Make NIX a community project

Acknowledgments

Programming and Data Model:

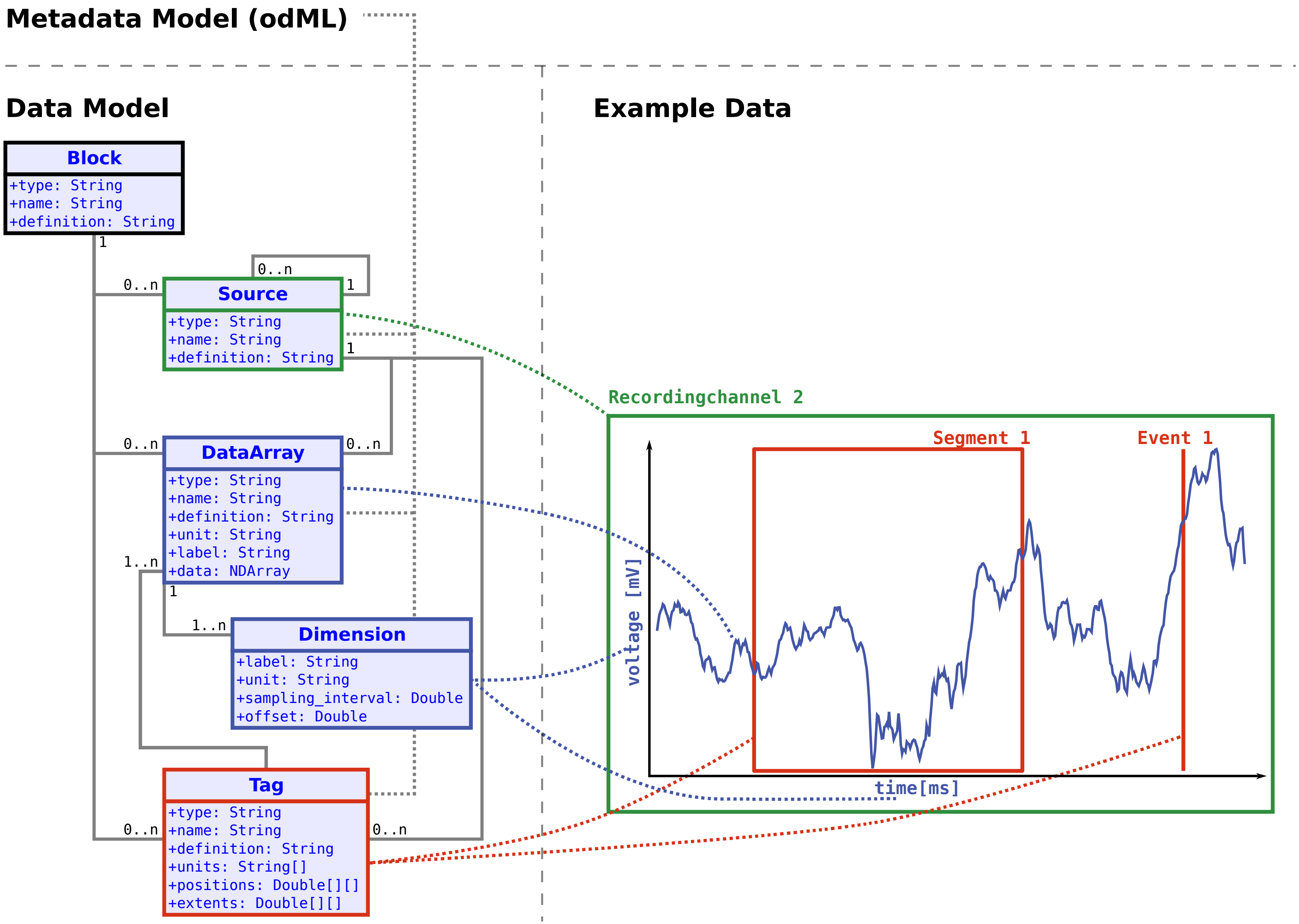
Adrian Stoewer
Christian Kellner
Jan Grewe
Balint Morvai
Andrey Sobolev

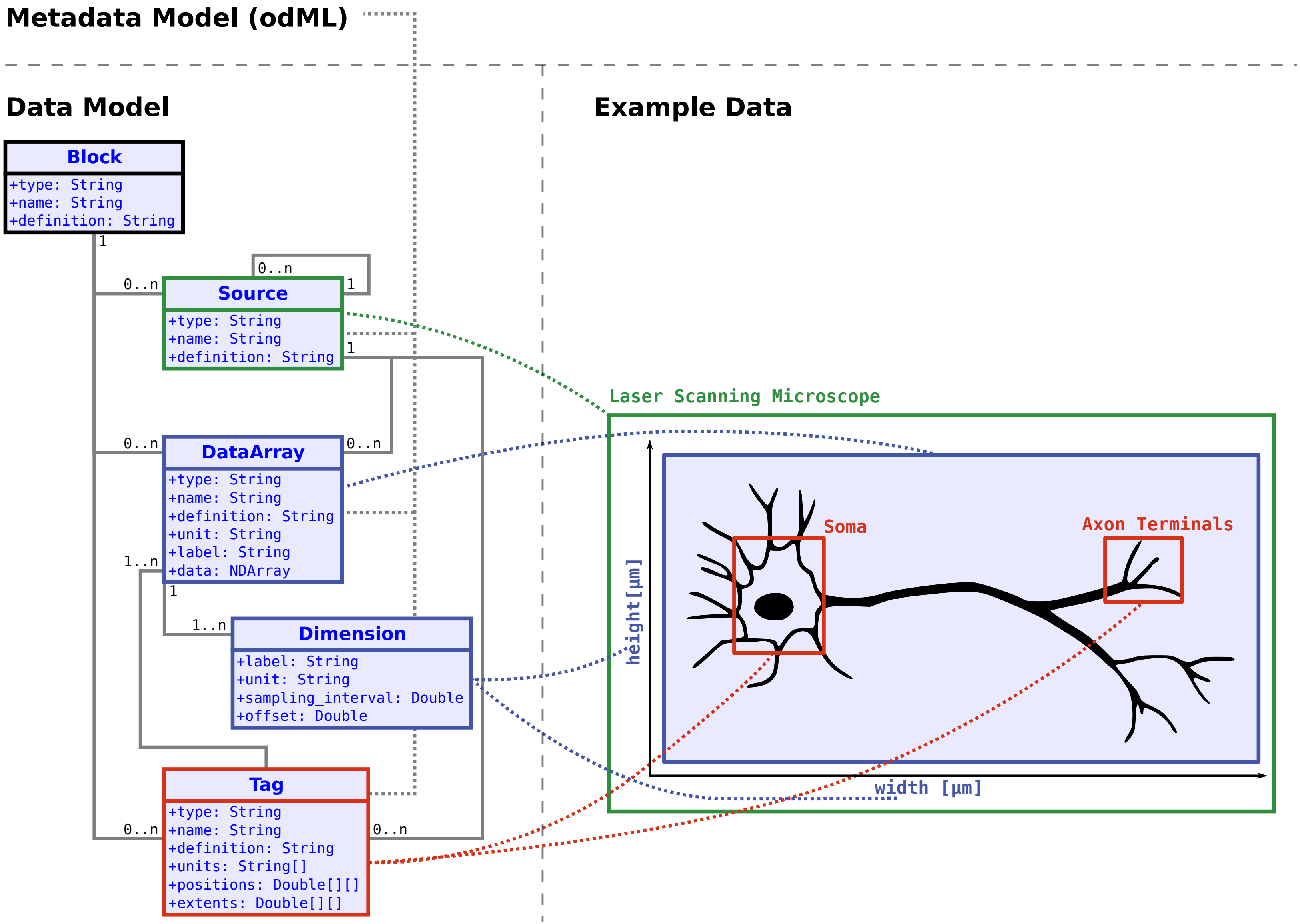
Project Lead:

Thomas Wachtler
Jan Grewe

Funding:

BMBF, grant 01GQ1302


Metadata Model (odML)
Data Model
Example Data
Block
+type: String
+name: String
+definition: String
1
0..n
Source
+type: String
+name: String
+definition: String
0..n
1
1
0..n
DataArray
+type: String
+name: String
+definition: String
+unit: String
+label: String
+data: NDArray
0..n
1..n
1
1..n
Dimension
+label: String
+unit: String
+sampling_interval: Double
+offset: Double
0..n
Tag
+type: String
+name: String
+definition: String
+units: String[]
+positions: Double[][]
+extents: Double[][]
0..n
Recordingchannel 2
Segment 1
Event 1
voltage [mV]
time[ms]


Metadata Model (odML)
Data Model
Example Data
Block
+type: String
+name: String
+definition: String
1
0..n
Source
+type: String
+name: String
+definition: String
0..n
1
1
0..n
DataArray
+type: String
+name: String
+definition: String
+unit: String
+label: String
+data: NDArray
0..n
1..n
1
1..n
Dimension
+label: String
+unit: String
+sampling_interval: Double
+offset: Double
Tag
+type: String
+name: String
+definition: String
+units: String[]
+positions: Double[][]
+extents: Double[][]
0..n
0..n
Laser Scanning Microscope
Soma
Axon Terminals
height[µm]
width [µm]