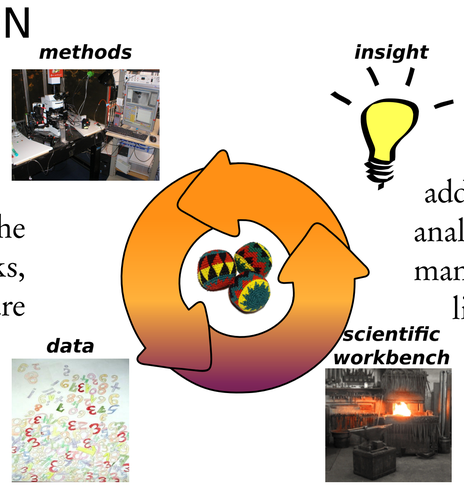
Datajongleur – controlling data objects and data models with Python at the neuroscientist's workbench

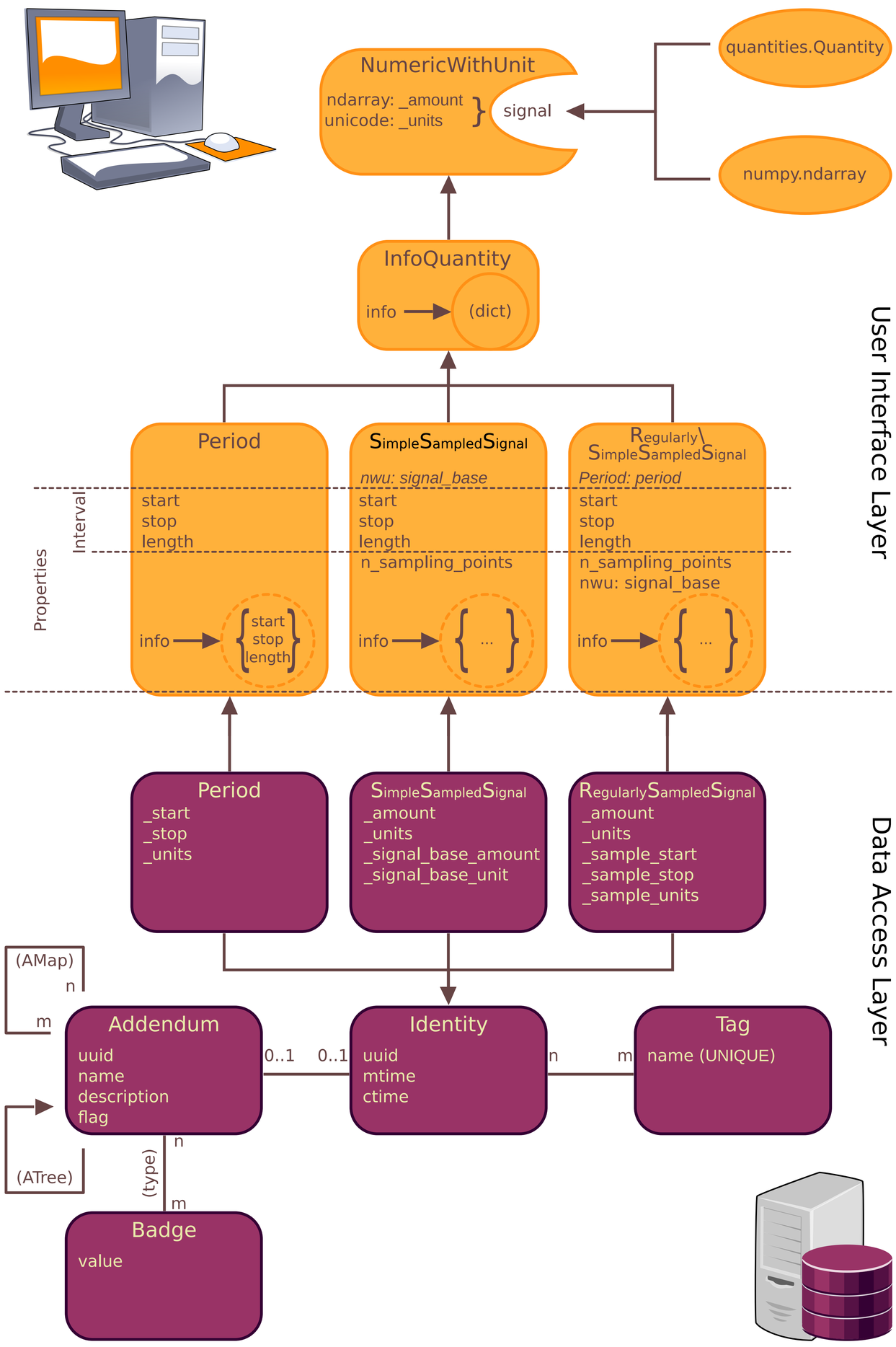
Philipp L. Rautenberg, Andrey Sobolev, Michael Stransky, and Thomas Wachtler
Ludwig-Maximilians-Universität München, Department of Biology, G-Node, Planegg-Martinsried, Germany

1. INTRODUCTION & MOTIVATION

Scientific progress depends increasingly on data management efforts that involve storing and structuring data, querying and analyzing data, exchange of data, and re-analysis of previously recorded data. To gain *insight* into the information processing in neurons and networks, electrophysiological techniques (*methods*) are indespensable. Progress in recording techniques and methodological approaches lead to ever increasing data volumes (*data*) and complexity of data. This causes a major barrer to fully exploit the scientific potential of experimental data. In addition, the increasing complexity of the experiments lead to an increasing amount of additional information (metadata). In order to make data analysis, re-analysis, and data sharing efficient, (meta-)data management has to start at the local *scientific workbench* closely linked to the analysis methods used by the scientist. We address these needs to facilitate data handling by providing an appropriate package "Datajongleur" for the Python language controlling standardized data objects and data models, and linking them to a persistent storage.

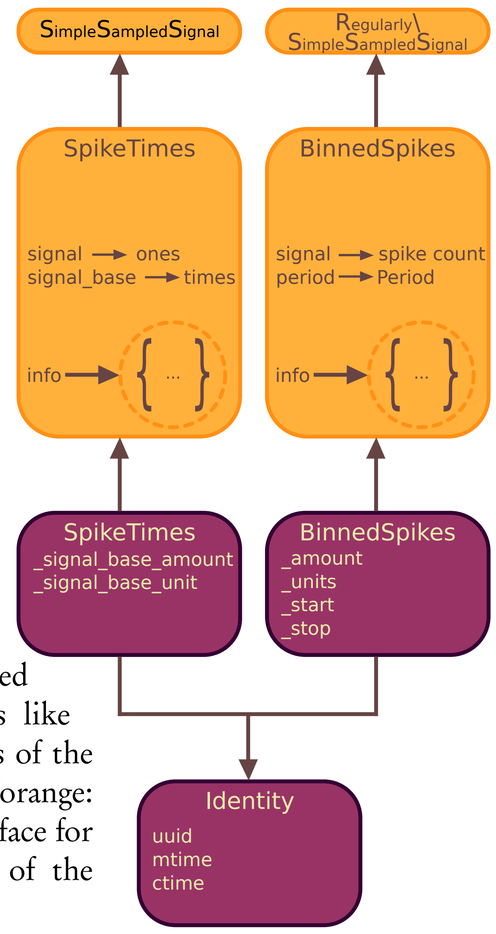


2. DATAJONGLEUR and its CORE STRUCTURE



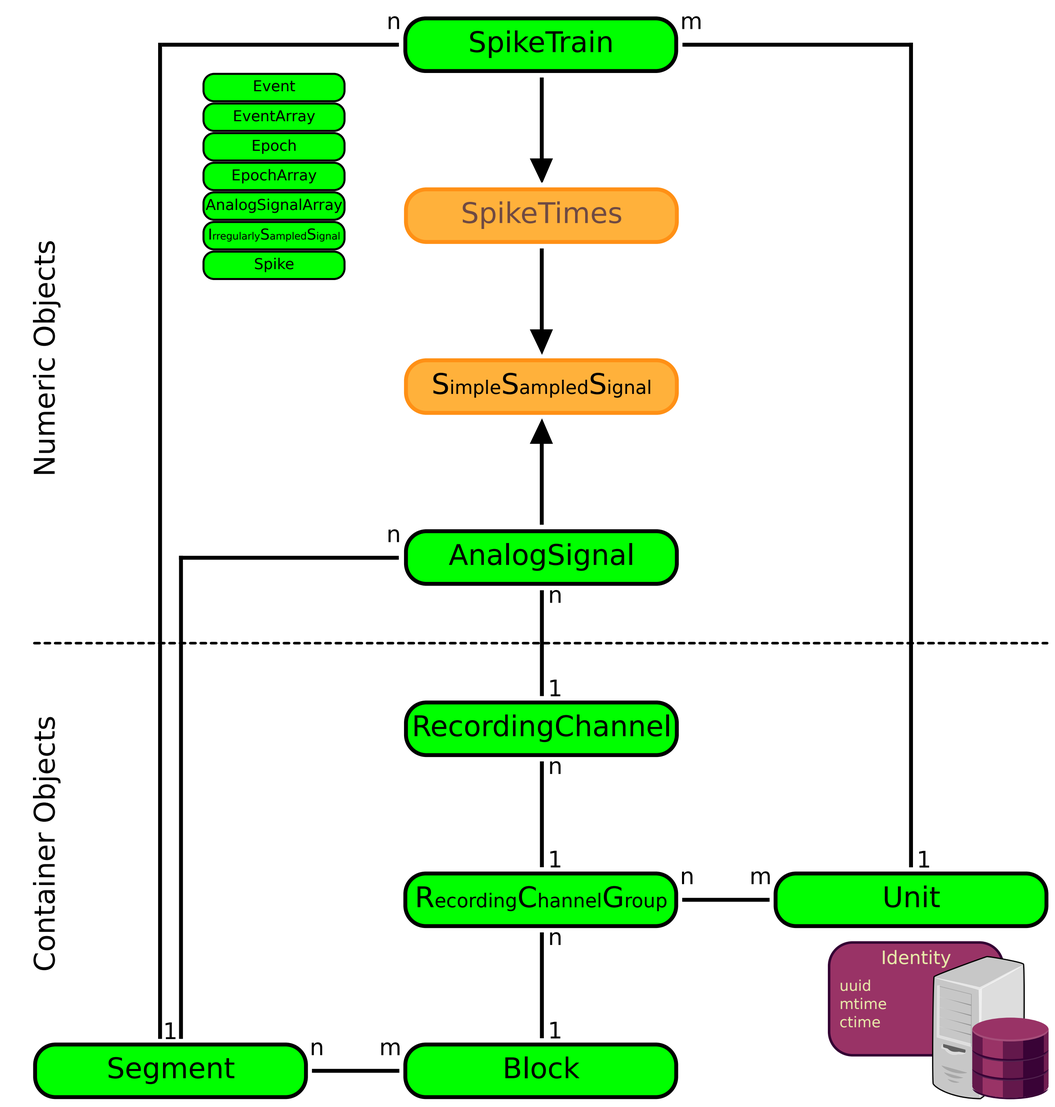
Datajongleur is a Python package that supports the scientist in controlling single data objects when writing scripts and software for analyzing scientific data. It provides predefined single objects ("beanbags") representing scientific data, such as recorded signals, including corresponding units. These objects can be loaded from, changed, and stored within a relational database like PostgreSQL [1] (server based) or SQLite [2] (file based). Furthermore, single data objects can be arranged via *Addendum* within domain specific data models which inherit on the data access layer from *Identity*, thus adding information about the relations between data objects. (Graphics adapted from [5] and [6]; classes are implemented within an user interface layer (orange) and within a data access layer (purple); for class-specifications see [7]).

3. SPECIFIC DOMAIN



The set of "beanbags" of the core structure can be extended by domain specific beanbags like SpikeTimes and BinnedSpikes of the domain of neuroscience (orange: classes optimized as user interface for the scientist; purple: classes of the data access layer).

4. EXTENDING TOWARDS EXISTING DATAMODELS



Single data objects can be arranged to define specific data models, thus adding information about the relations between data objects. Datajongleur handles domain specific data models as extensions. Some recently evolving data models are already implemented at a corresponding stage of development, such as neo [3] which is a data model with objects for the domain of electrophysiology. It organizes numeric object like *SpikeTrain* and *AnalogSignal* with container objects like *Block* or *Segment*. In addition, scientists can write their own extensions if necessary. All classes of the existing data model (green/black) are implemented according to the user interface layer (orange) and data access layer (purple).

5. CONCLUSION

Datajongleur bridges a gap between persistent data storage realized with relational database technology on the one side and single data objects that can be arranged as data models on the other side. This facilitates data management for the scientist on the level of scripting and programming where data analysis takes place.

ACKNOWLEDGEMENTS
This work is supported by the Federal Ministry of Education and Research (BMBF, grant 01GQ0801 and grant 01GQ1116).

CONTACT
philipp.rautenberg@g-node.org

REFERENCES

[1] http://www.postgresql.org/
[2] http://www.sqlite.org/
[3] http://neuralensemble.org/trac/neo
[4] https://github.com/NeuralEnsemble/libNeuroML/
[5] Computer-aj_aj_ashton_01.svg, 7.8.2012, http://en.wikipedia.org/
[6] database-server.svg, 7.8.2012, http://www.clipart-rahmen.de/datenbankserver/
[7] Open Source Pojecte Datajonlegur, https://github.com/G-Node/Datajongleur