A Versatile Scientific Programming Environment for Cardiovascular Data Analysis

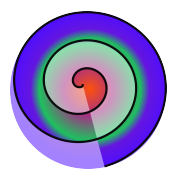
A. Schlemmer

T. Hajek, U. Parlitz, S.E. Lehnart, N. Wessel, H. Malberg and S. Luther

Max Planck Research Group Biomedical Physics,
Max Planck Institute for Dynamics and Self-Organization, Göttingen

http://www.bmp.ds.mpg.de/
alexander.schlemmer@ds.mpg.de

April 13, 2010



Aims

- Platform for cardiovascular data analysis in multi-disciplinary environments

Aims

- Platform for cardiovascular data analysis in multi-disciplinary environments
- Intuitive graphical user interface

Aims

- Platform for cardiovascular data analysis in multi-disciplinary environments
- Intuitive graphical user interface
- High level of interactivity

Aims

- Platform for cardiovascular data analysis in multi-disciplinary environments
- Intuitive graphical user interface
- High level of interactivity
- Performant application of algorithms

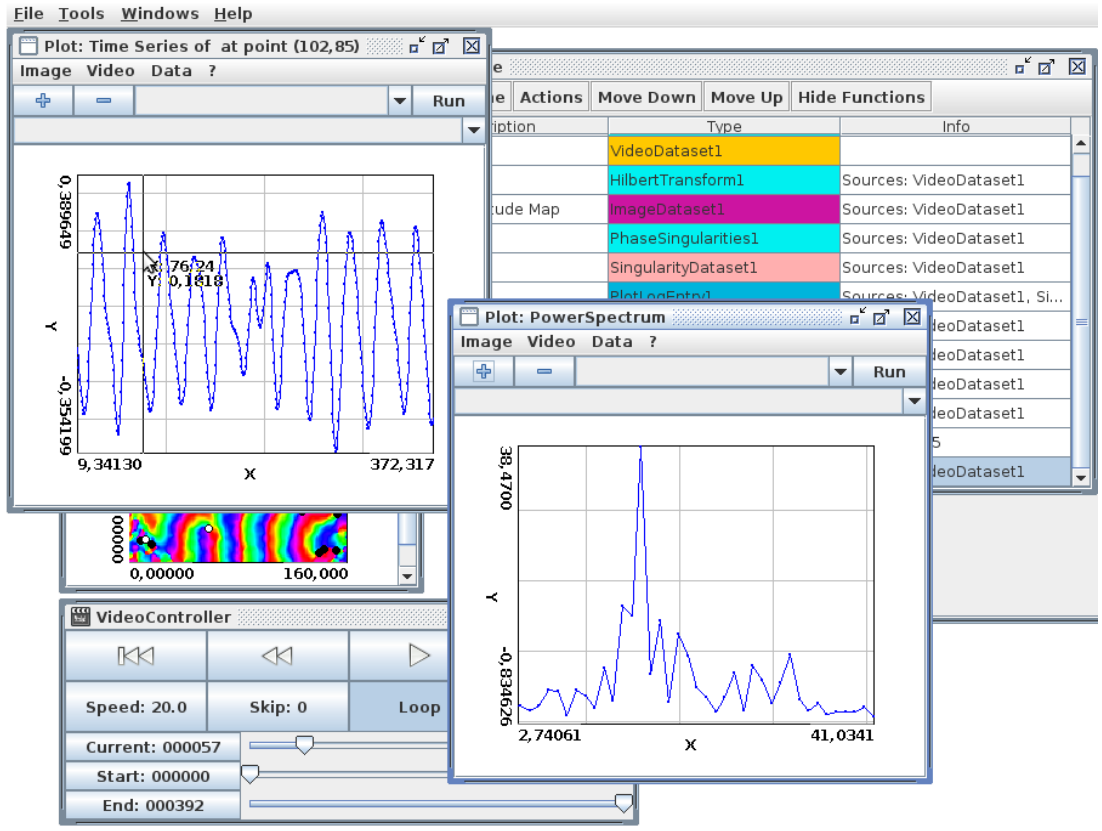
Aims

- Platform for cardiovascular data analysis in multi-disciplinary environments
- Intuitive graphical user interface
- High level of interactivity
- Performant application of algorithms
- Efficient and standardized development of algorithms

Aims

- Platform for cardiovascular data analysis in multi-disciplinary environments
- Intuitive graphical user interface
- High level of interactivity
- Performant application of algorithms
- Efficient and standardized development of algorithms
- Reproducability of results

OpenAnalyser



OpenAnalyser is an open source-licensed (GNU GPL) Java application for scientific data analysis with a graphical user interface (GUI) and integrated plugin system.

The Java Platform

- Platform independence

The Java Platform

- Platform independence
- Powerful API including GUI-toolkit

The Java Platform

- Platform independence
- Powerful API including GUI-toolkit
- Prevalence

The Java Platform

- Platform independence
- Powerful API including GUI-toolkit
- Prevalence
- Robustness

The Java Platform

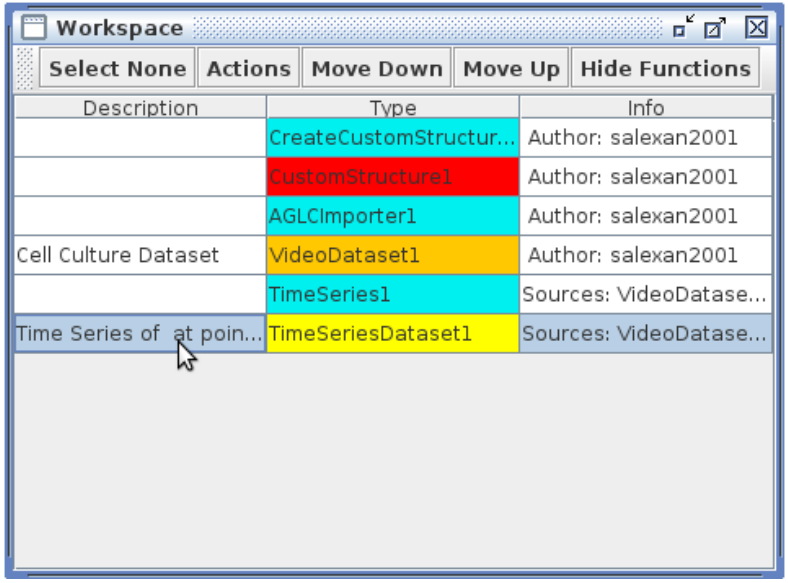
- Platform independence
- Powerful API including GUI-toolkit
- Prevalence
- Robustness
- Modularity and dynamic features

The Java Platform

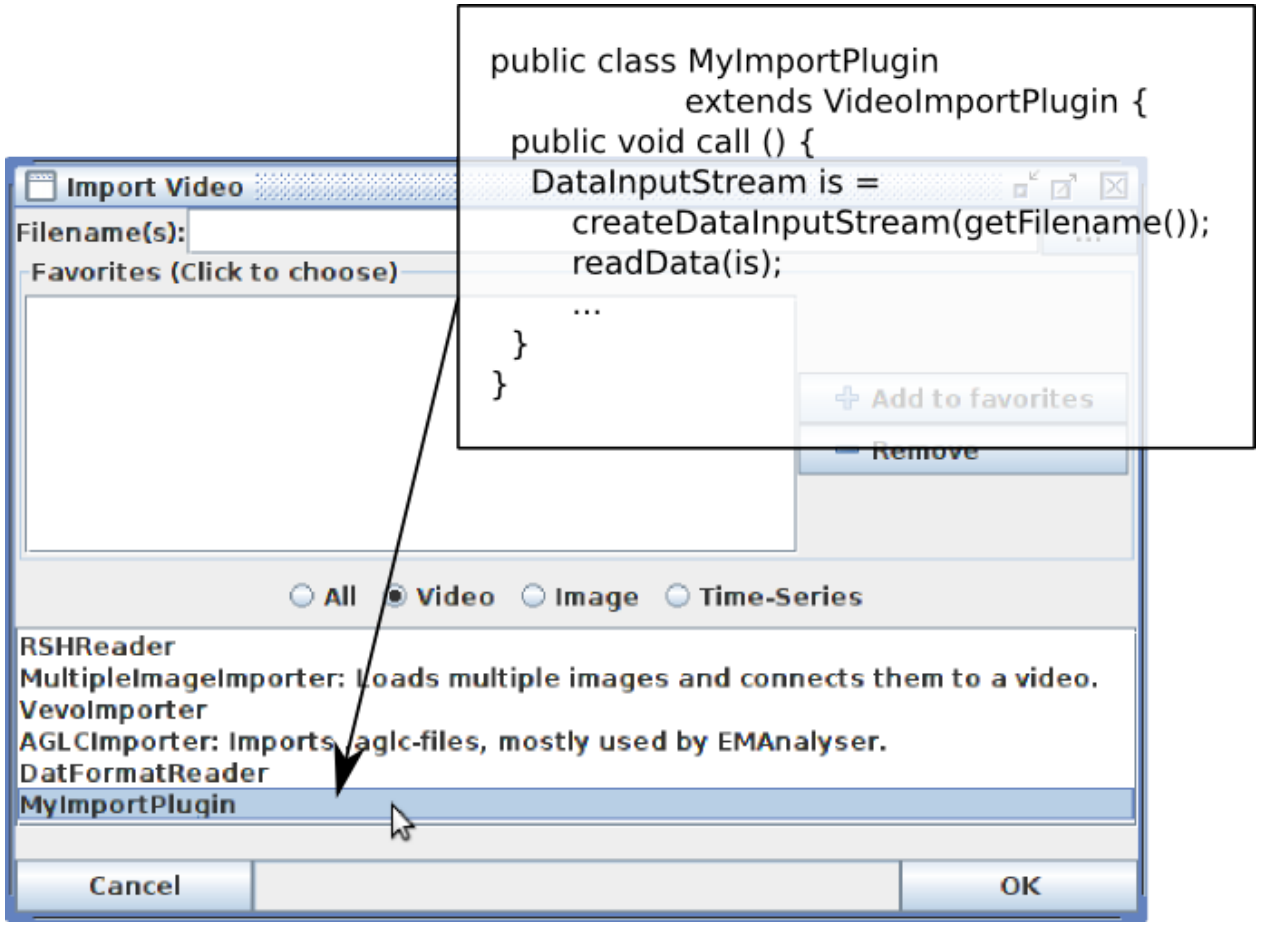
- Platform independence
- Powerful API including GUI-toolkit
- Prevalence
- Robustness
- Modularity and dynamic features
- Integration of other programming languages, e.g. Python, C++ or FORTRAN

The Workspace

- Overview over data, analysis steps and results
- Visualization and analysis algorithms applicable via context menus

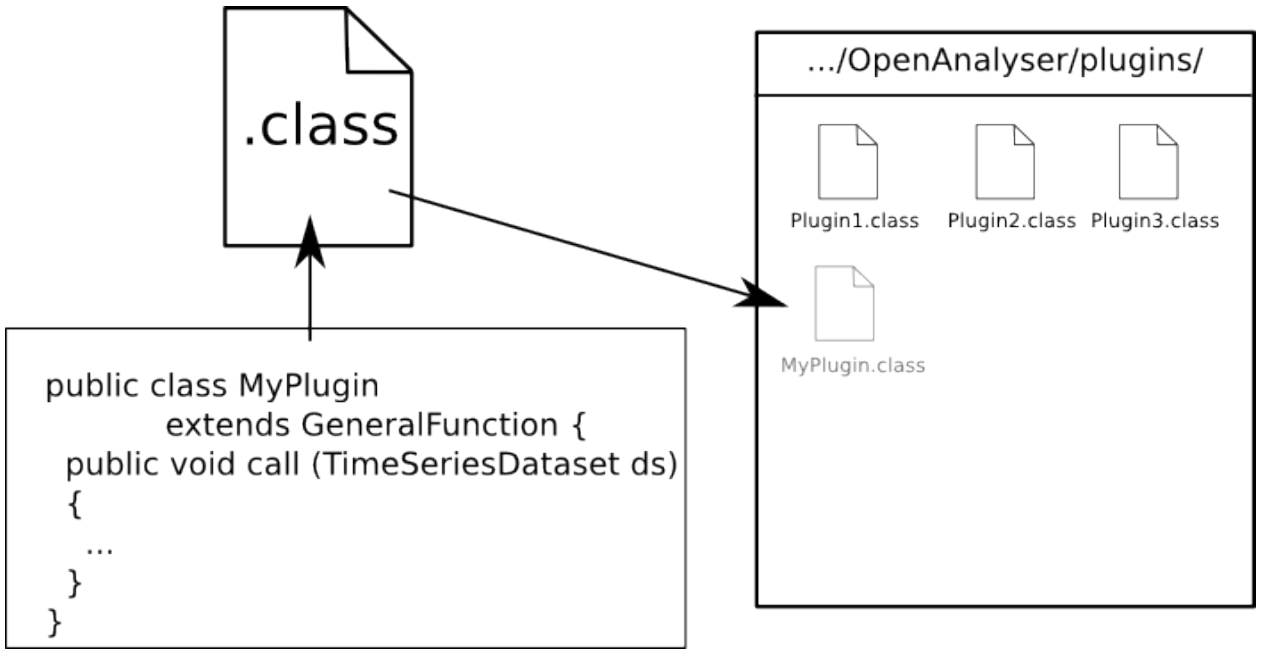


Data Import



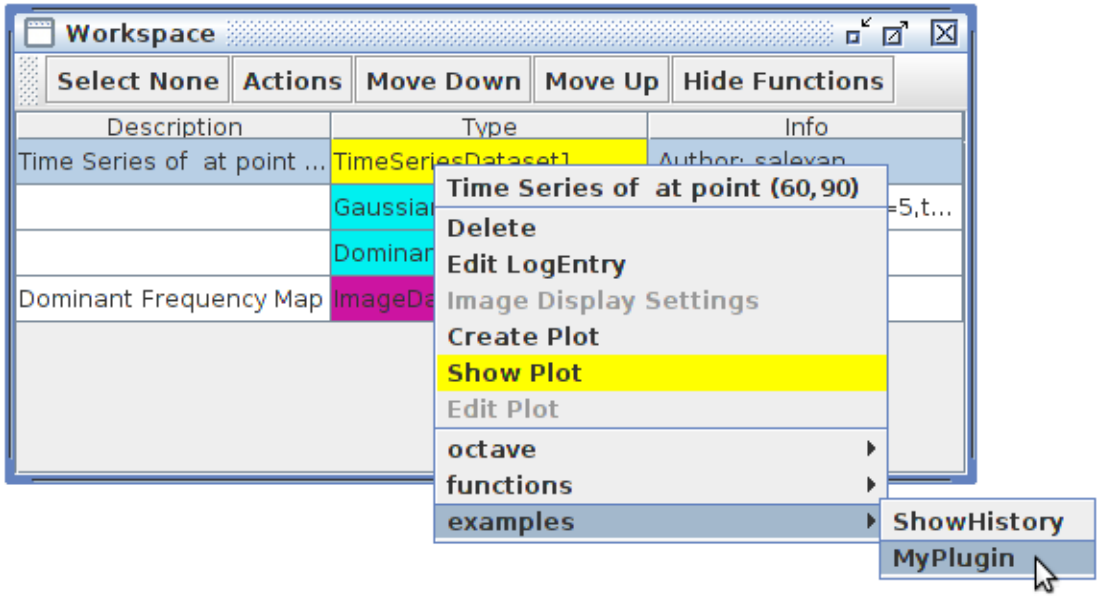
- Import plugins facilitate the integration of data formats.

Plugin-System



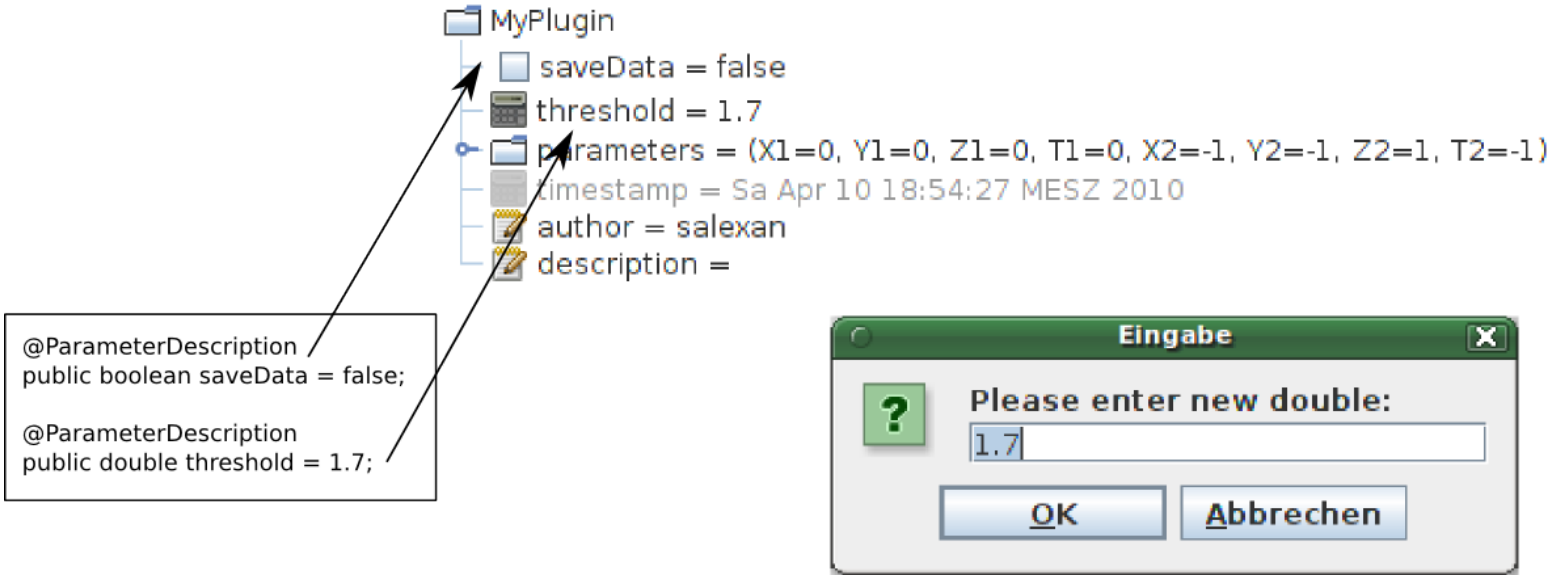
- Simple, yet powerful and flexible plugin-system for development of algorithms for OpenAnalyser
- Permits users to add their own algorithms

Integration of Algorithms



- New algorithms are integrated into the GUI and can even be reloaded at runtime

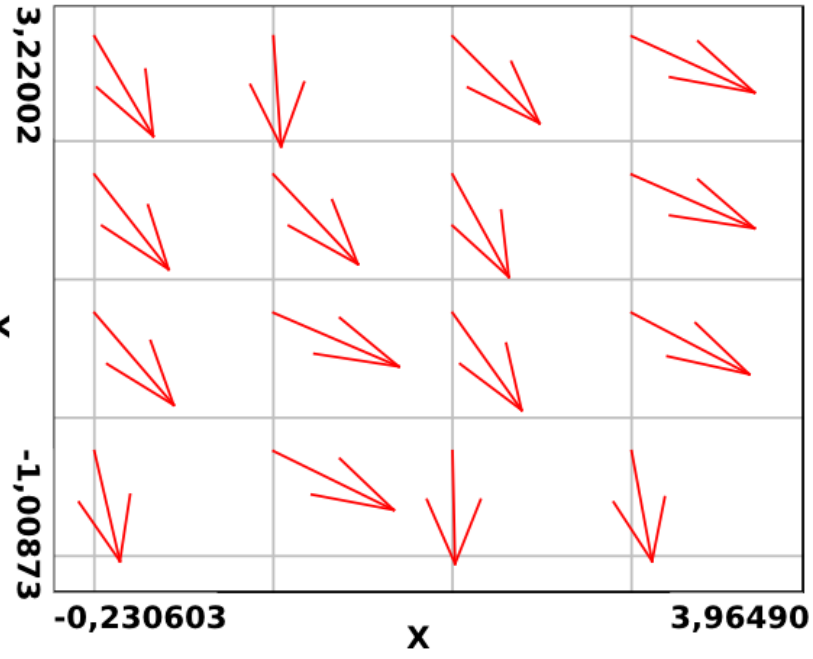
Parameters



- Simple declaration of parameters needed by plugins
- Automatic integration of parameters into user-interface

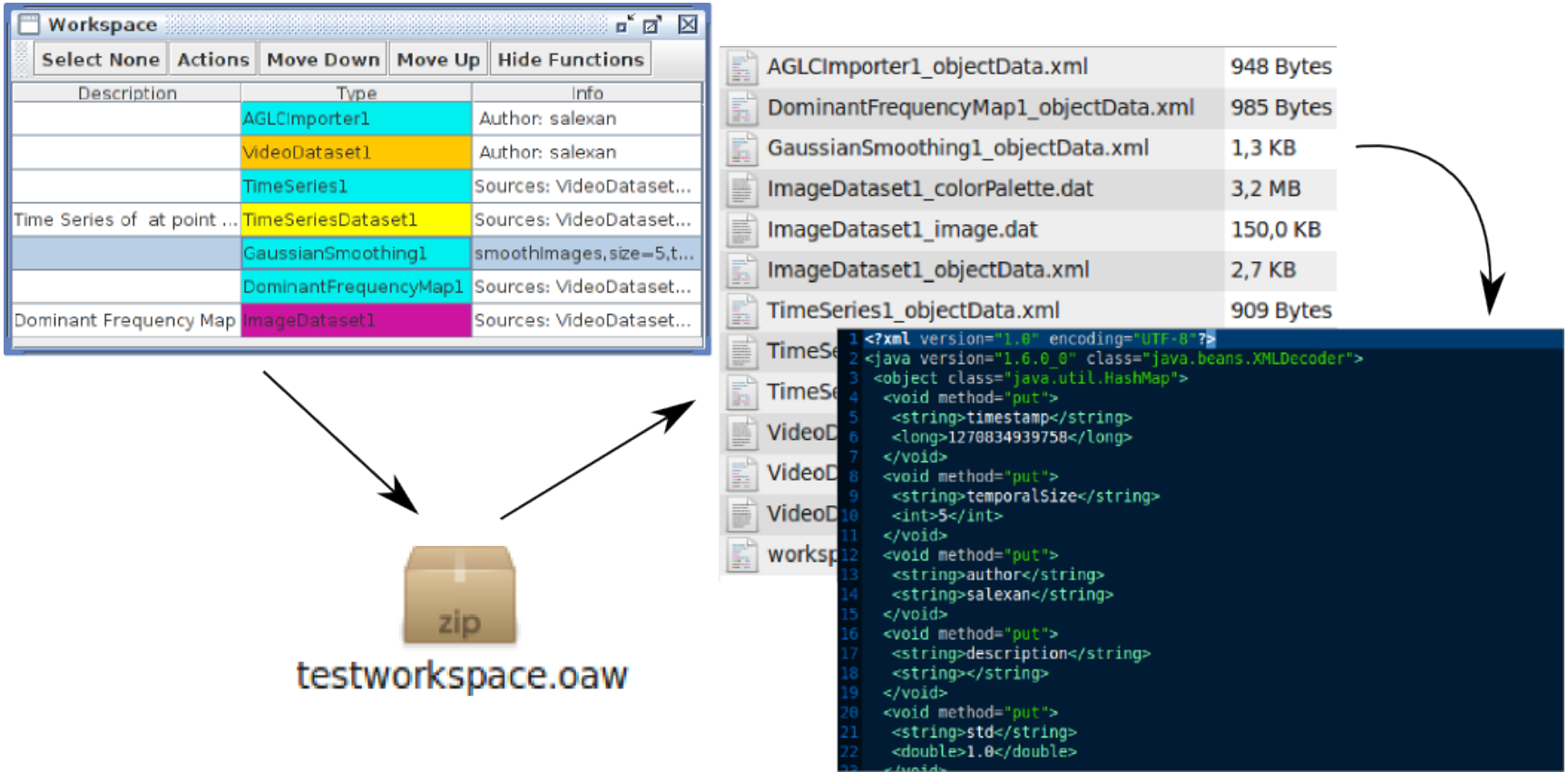
Visualization-System

```
public class VectorVisualizer
                extends GeneralVisualizer {

    private VectorDataset dataset;

    public void paint(Graphics2D g) {
        for (int i = 0; i < dataset.vectors.size(); i++) {
            Arrow vector = dataset.vectors.get(i);
            DoubleXY pos = vector.pos;

            ...

            }
    }
}
```



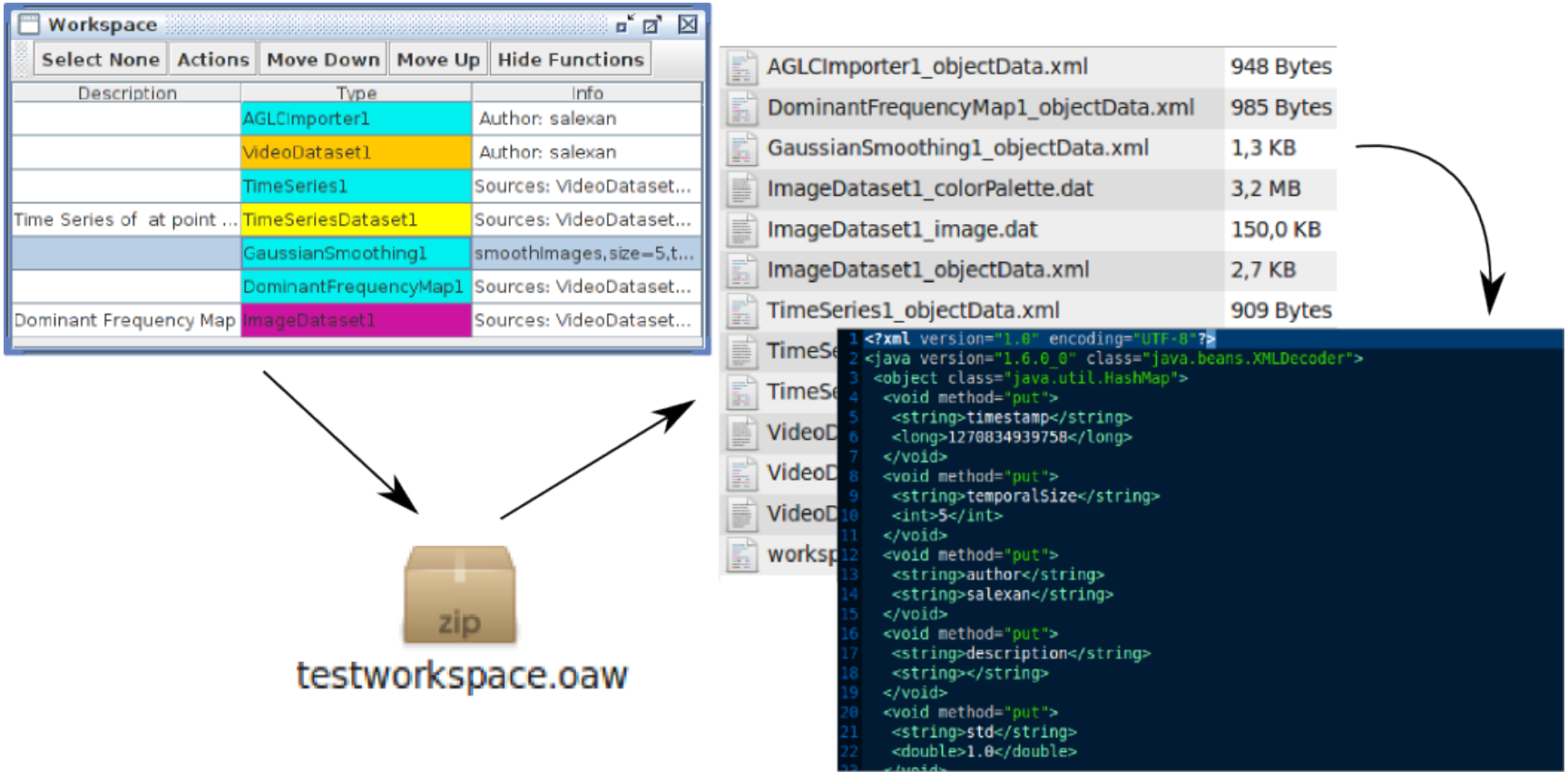
- OpenAnalyser offers basic visualization functionality (e.g. xy-plots, video display, ...)
- Support for custom visualization methods (e.g. for custom datatypes) via visualization plugins

Workspace storage and logging mechanism



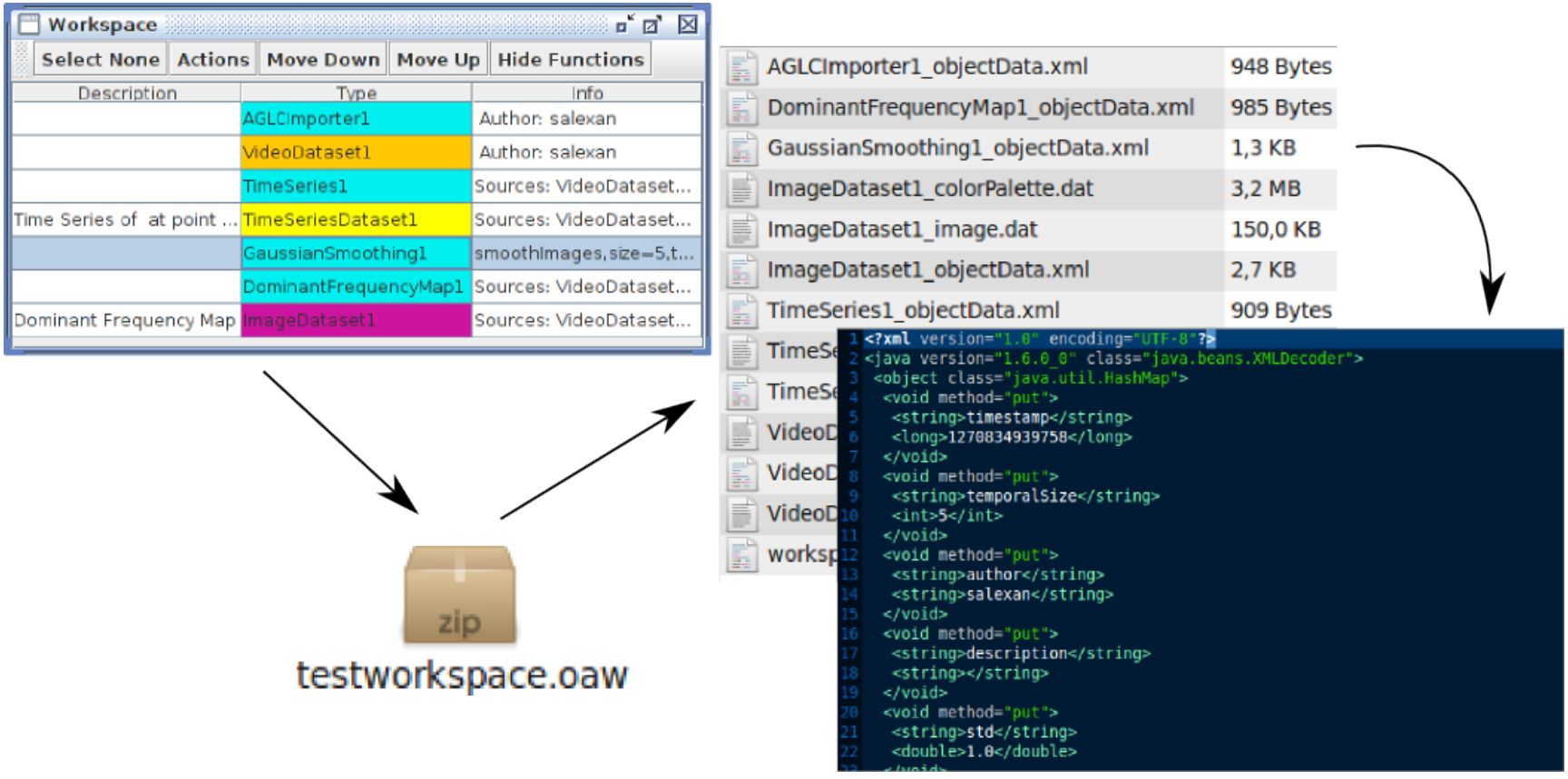
- Workspace objects are mapped to xml- and binary-data

Workspace storage and logging mechanism



- Workspace objects are mapped to xml- and binary-data
- Storage in container-files or databases possible

Workspace storage and logging mechanism



- Workspace objects are mapped to xml- and binary-data
- Storage in container-files or databases possible
- Each step in the analysis procedure can be reproduced

Conclusion

- OpenAnalyser is designed for collaborative data analysis in large-scale, cross-disciplinary research networks
- OpenAnalyser combines an intuitive GUI with a powerful prorgamming framework for algorithm development and application
- Users can integrate existing code easily via versatile plugin system
- OpenAnalyser is work in progress. Your suggestions, comments and contributions are welcome!

Thank you!

Thank you for your attention!

Download and more Information about OpenAnalyser:

http://www.bmp.ds.mpg.de/software.html

The project is supported by:

- BMBF FKZ 01EZ0905/6
- European Community FP7 EUTrigTreat (HEALTH-F2-2009-241526)