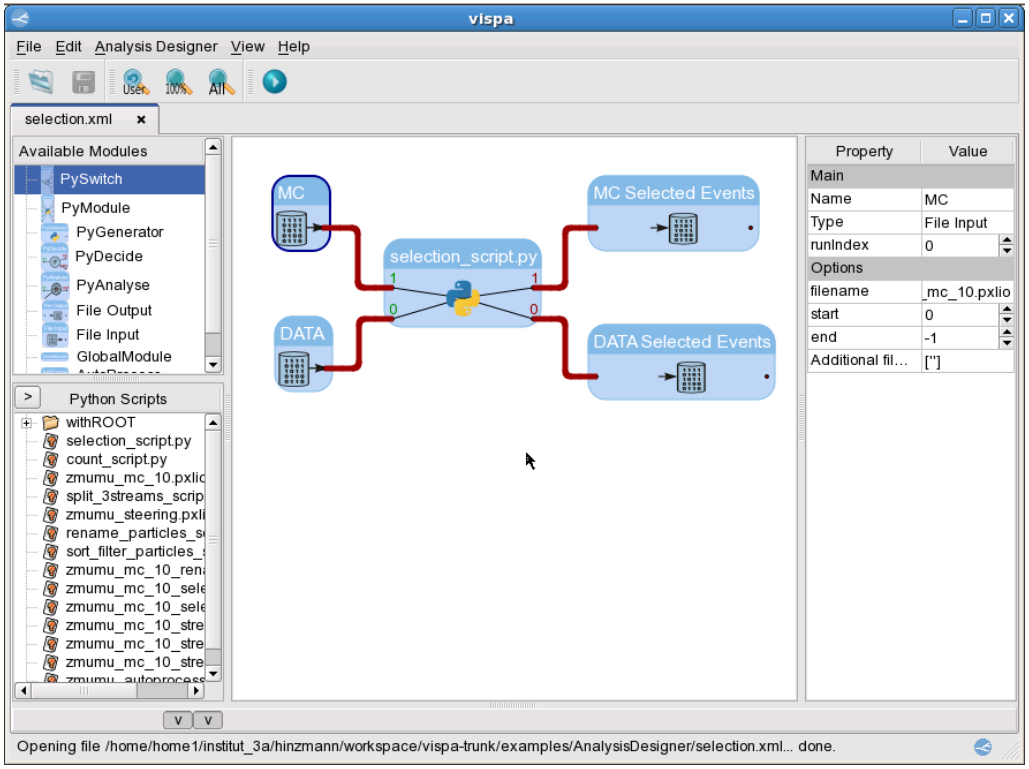
Visual Physics Analysis (VISPA)

A graphical development environment for physics data analysis



GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung

M.Brodski, M.Erdmann, R.Fischer, **Andreas Hinzmann**, T.Klimkovich, D.Klingebiel, M. Komm, G.Müller, T.Münzer, J.Steggemann, T.Winchen

PHYSICS AT THE TERA SCALE Helmholtz Alliance

Introduction

- Common graphical tools for physics analysis

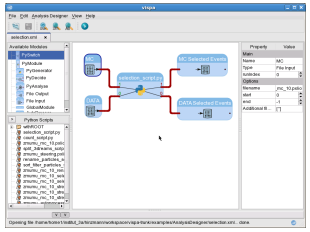
IDE for code development (e.g. Eclipse)

Data browsing (e.g. Event displays, ROOT-TBrowser)

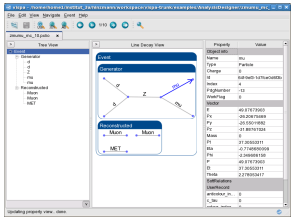
- Visual Physics Analysis (VISPA):

Integrated Development Environment (IDE) for Physics analysis

Analysis development

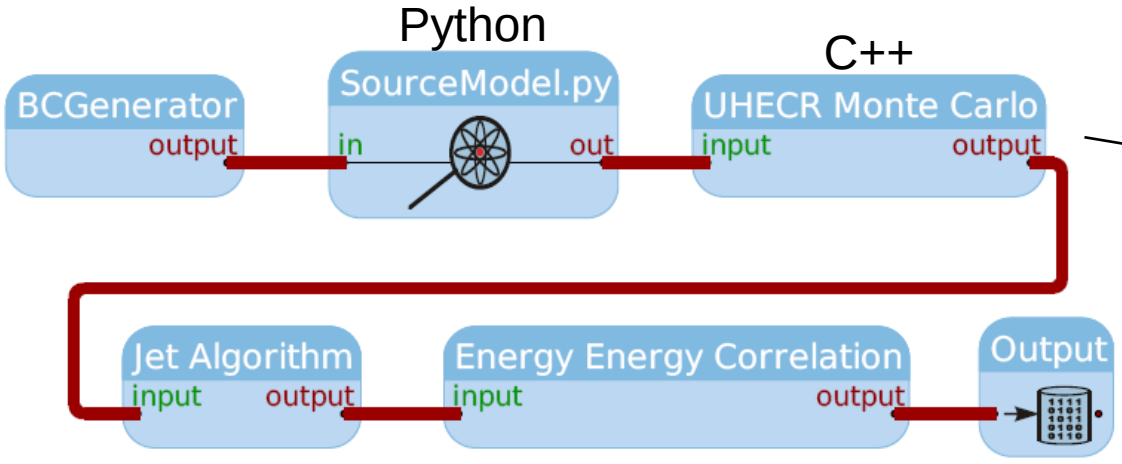


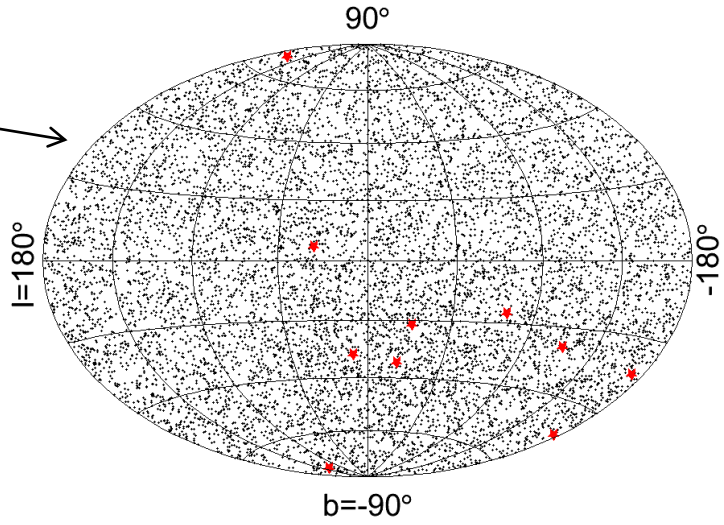
Data browsing

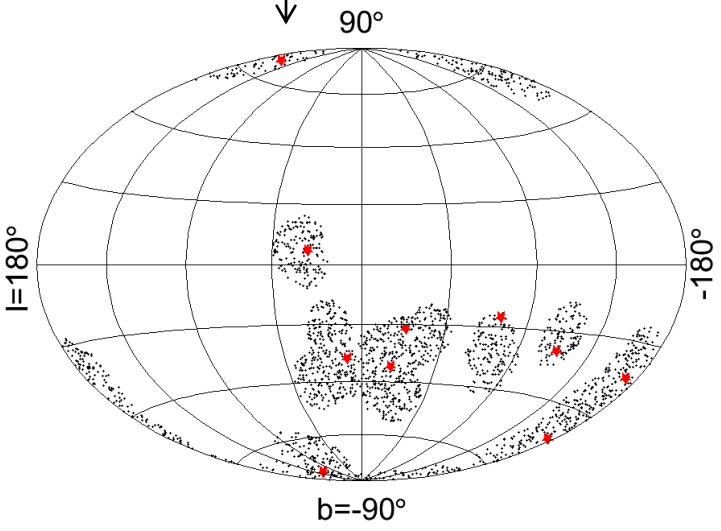


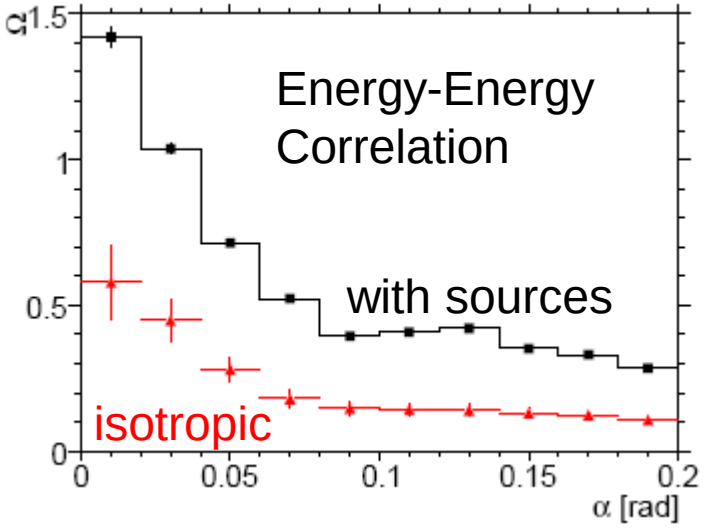
Example from Astroparticle Physics

Measuring Cosmic Magnetic Fields with Ultra High Energy Cosmic Ray Data









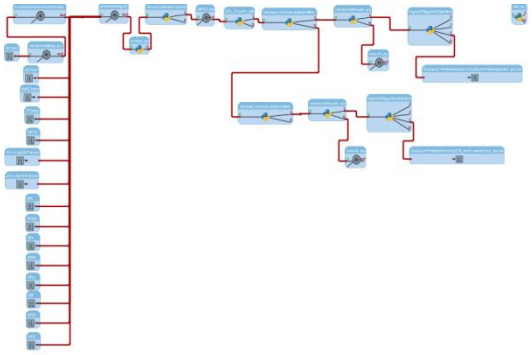
MC study

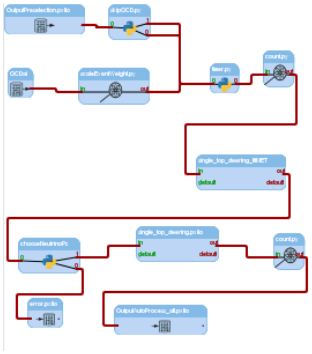
M. Erdmann, P. Schiffer
arXiv:0904.4888
accepted
Astroparticle Physics

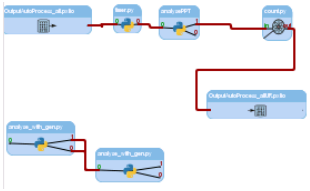
Example from High-Energy Physics

Single top at CMS

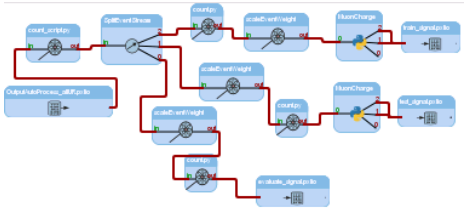
Data/MC → Detector reconstruction → Preselection → Single top reconstruction → Control plots →

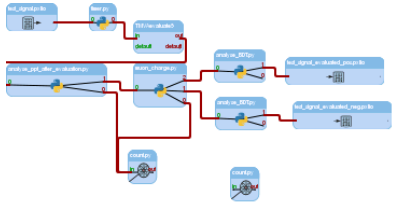


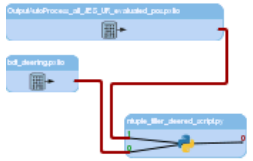


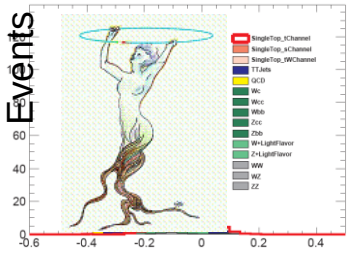


→ Sample splitting → Boosted decision tree training → Boosted decision tree evaluation → Output and plots









BDT output

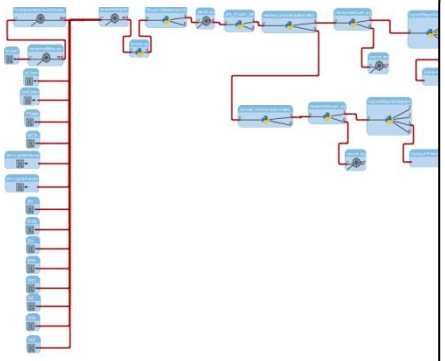
For CMS members: http://indico.cern.ch/contributionDisplay.py?contribId=5&confId=84275

Example from High-Energy Physics

Single top at CMS

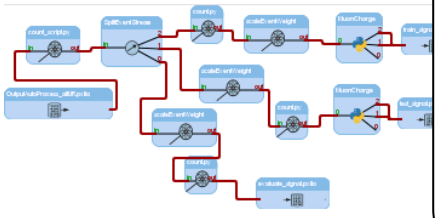
Data/MC → Detector reconstruction

Preselection



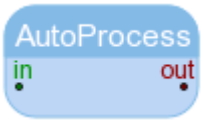
Sample splitting

Boosted decision tree tr

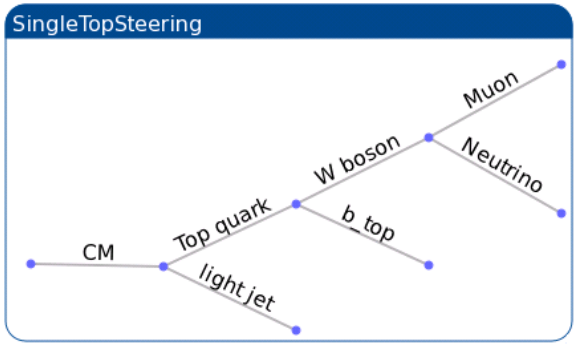


Single top reconstruction

Autoprocess module

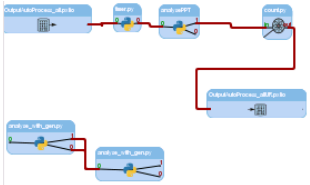


Reconstruct decay tree using steering tree

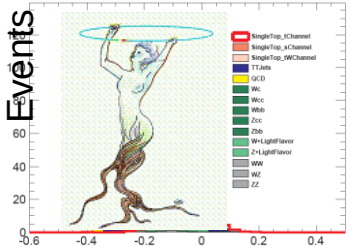


Multiple versions for reconstruction ambiguities

Control plots →



tput and plots



BDT output

For CMS members: http://indico.cern.ch/contributionDisplay.py?contribId=5&confId=84275

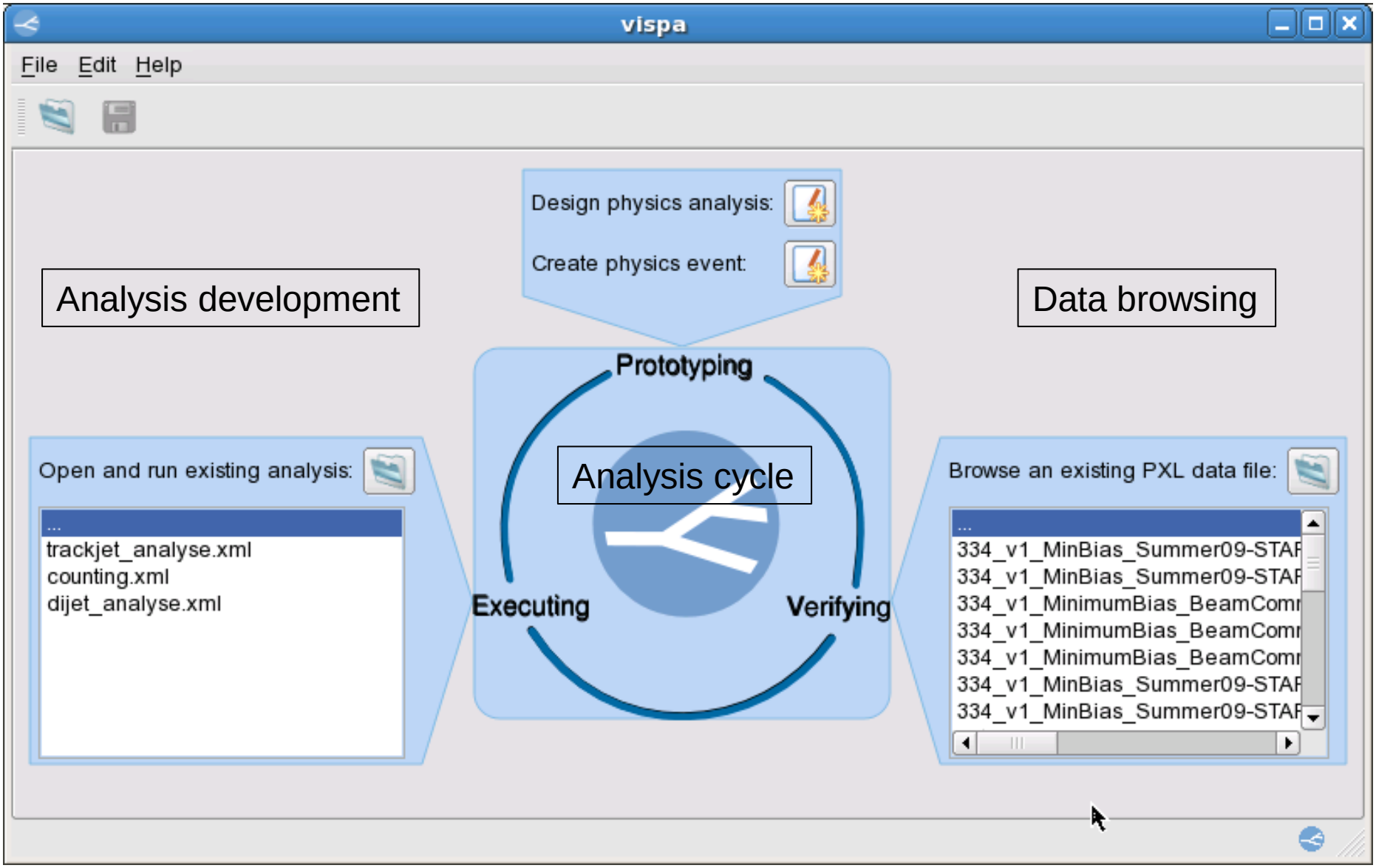
Fields of application

- Expert analyses
  - High flexibility of analyses concept and data format
  - Implement analyses of any complexity
  - **Spend more time on physics problem than on implementation**

- Student level analyses
  - Minimal time of learning
  - For easy analysis little knowledge of programming needed
  - Clear structuring of analyses
  - Clear understanding of data

- Teaching
  - Quick and easy implementation of simple problems
  - Used in “Elementary particles” (4th year students) hands-on exercises at RWTH Aachen University (2009)

IDE for Physics analysis

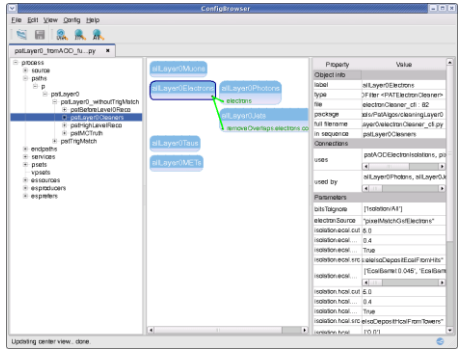


IDE for Physics analysis

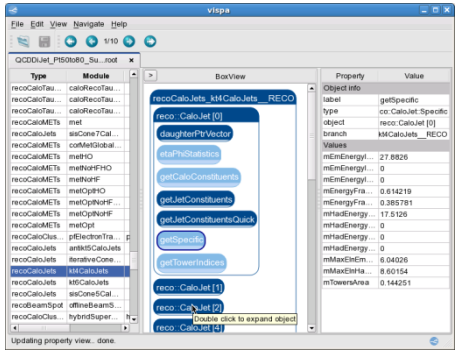
- Develop entire analyses in a single IDE
  - **Experiment independent** and **experiment specific** analysis steps

- Major redesign of the Graphical User Interface codebase
  - Easy integration with experiment specific software
  - **Graphical platform based on a plugin mechanism**

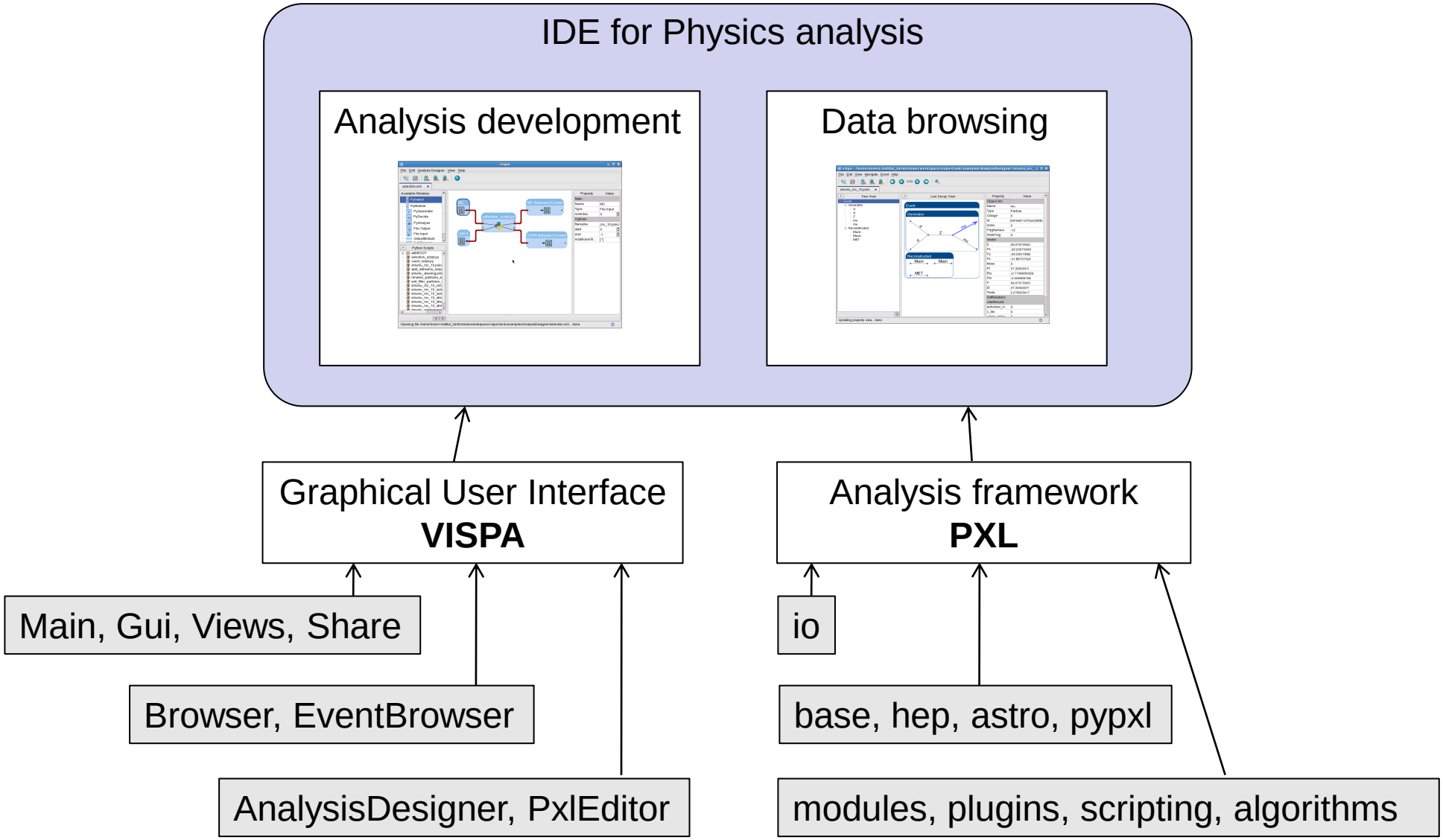
- Examples:
  - Editor for CMS configuration files (CHEP 2009)
    edmConfigEditor



  - Browser for CMS data files (current development)
    edmBrowser



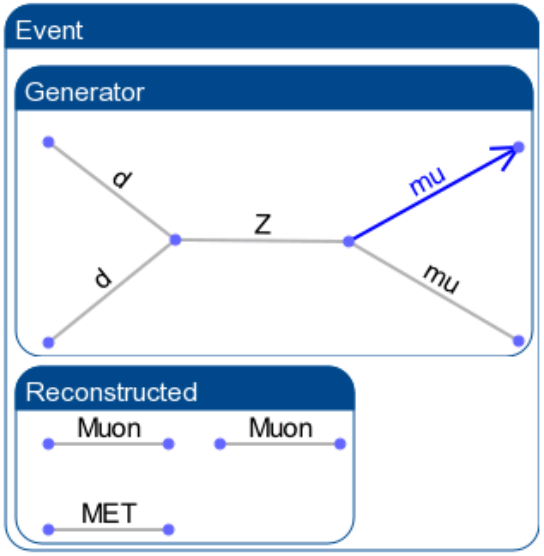
The ingredients of VISPA



PXL (Physics eXtension Library)

- C++ toolkit for physics analysis
- Analysis framework for VISPA

- Interfaces for physics analysis: base, hep, astro
  - Physics objects: pxl::Particle, pxl::UHECR
  - Containers: pxl::Event, pxl::BasicContainer
  - Object relations: pxl::Relations (e.g. of pxl::Particle)
  - User data: pxl::UserRecord (e.g. of pxl::Event)

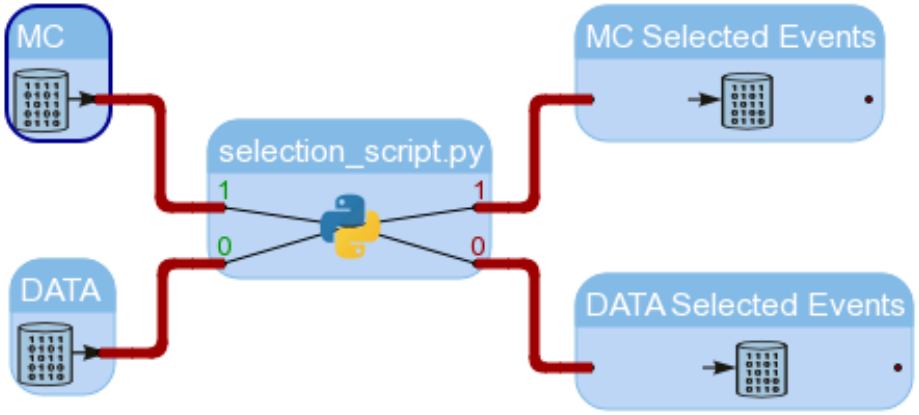


- PyPXL: pypxl
  - All classes available in Python
  - Use SWIG for automatic generation of Python extension

- I/O format: io
  - All physics objects are pxl::Serializables
  - Fast and highly flexible
  - ZLIB for compression

Analyses in PXL

- Modular physics analyses with multiple sinks and sources



- Dataflow:
  - visualized by connection lines from left to right
  - starting from input/generator module
  - **Common interface for all modules**:
    HEP: pxl::Event , Astro: pxl::BasicContainer , Or: any pxl::Serializable
- Load/save in XML format

- Run → Interactively in VISPA
  → On batch using pxlrun

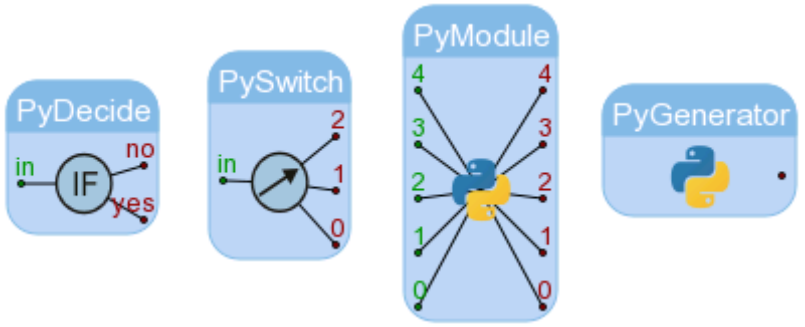
Modules in PXL

- **C++ and Python modules in same analysis**
  - C++ for performance-sensitive analysis modules
  - Python for fast-prototyping and analysis logic

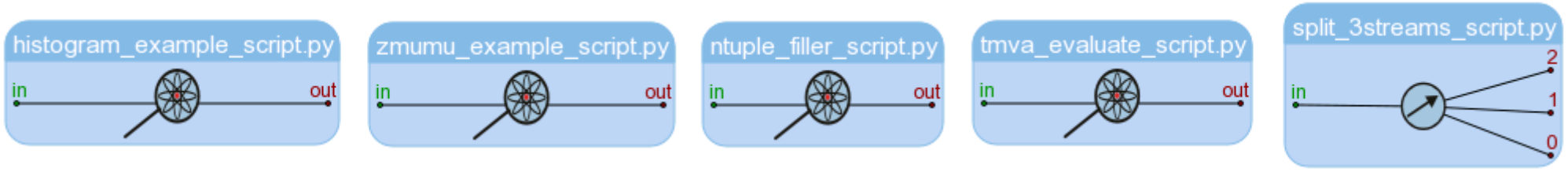
modules, plugins, scripting, algorithms

Standard modules

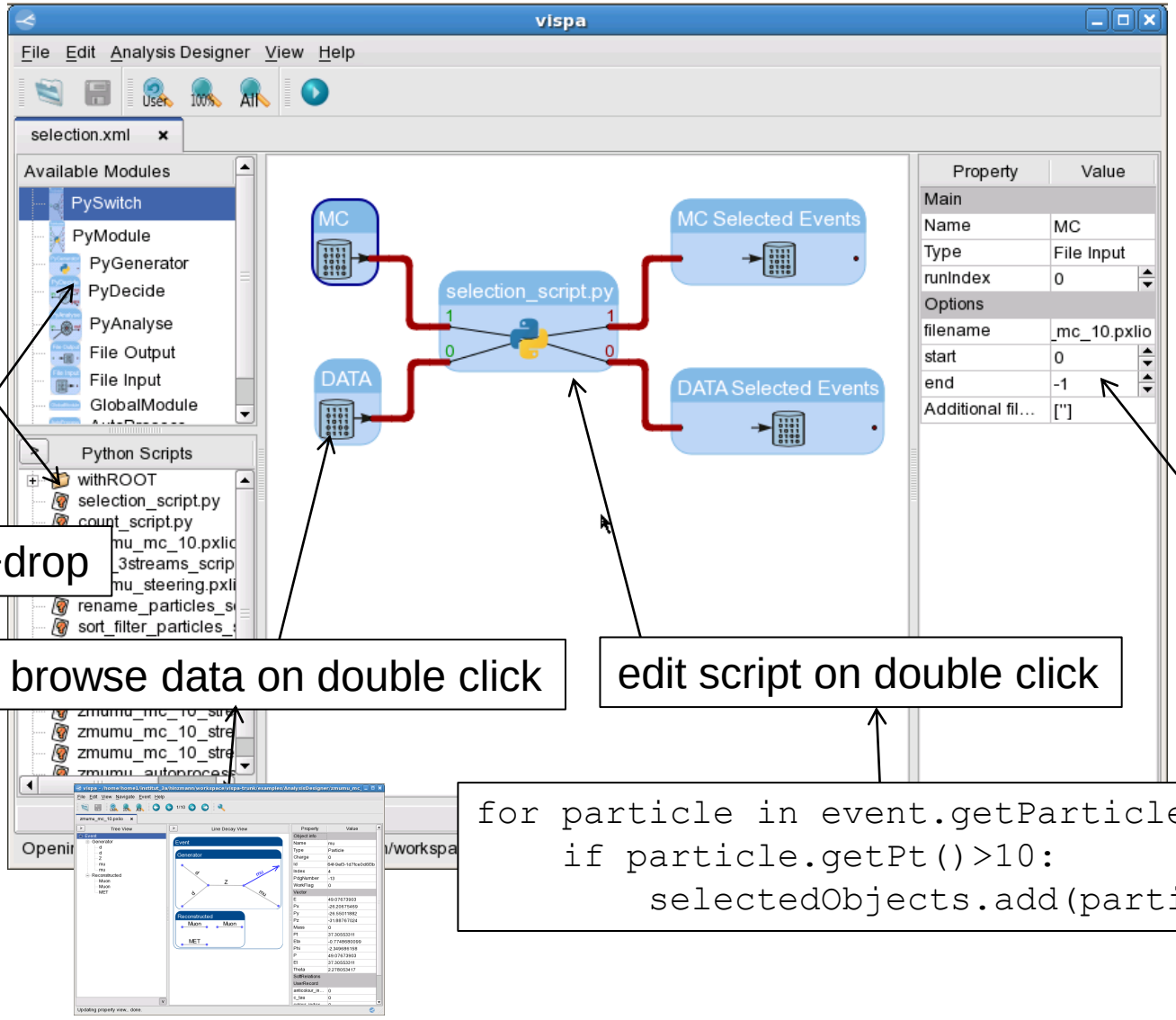
Logical modules



- VISPA delivers a set of examples
  - Explain access to common tools: e.g. plotting using PyROOT



Visual development of analyses



available
C++ and Python
modules

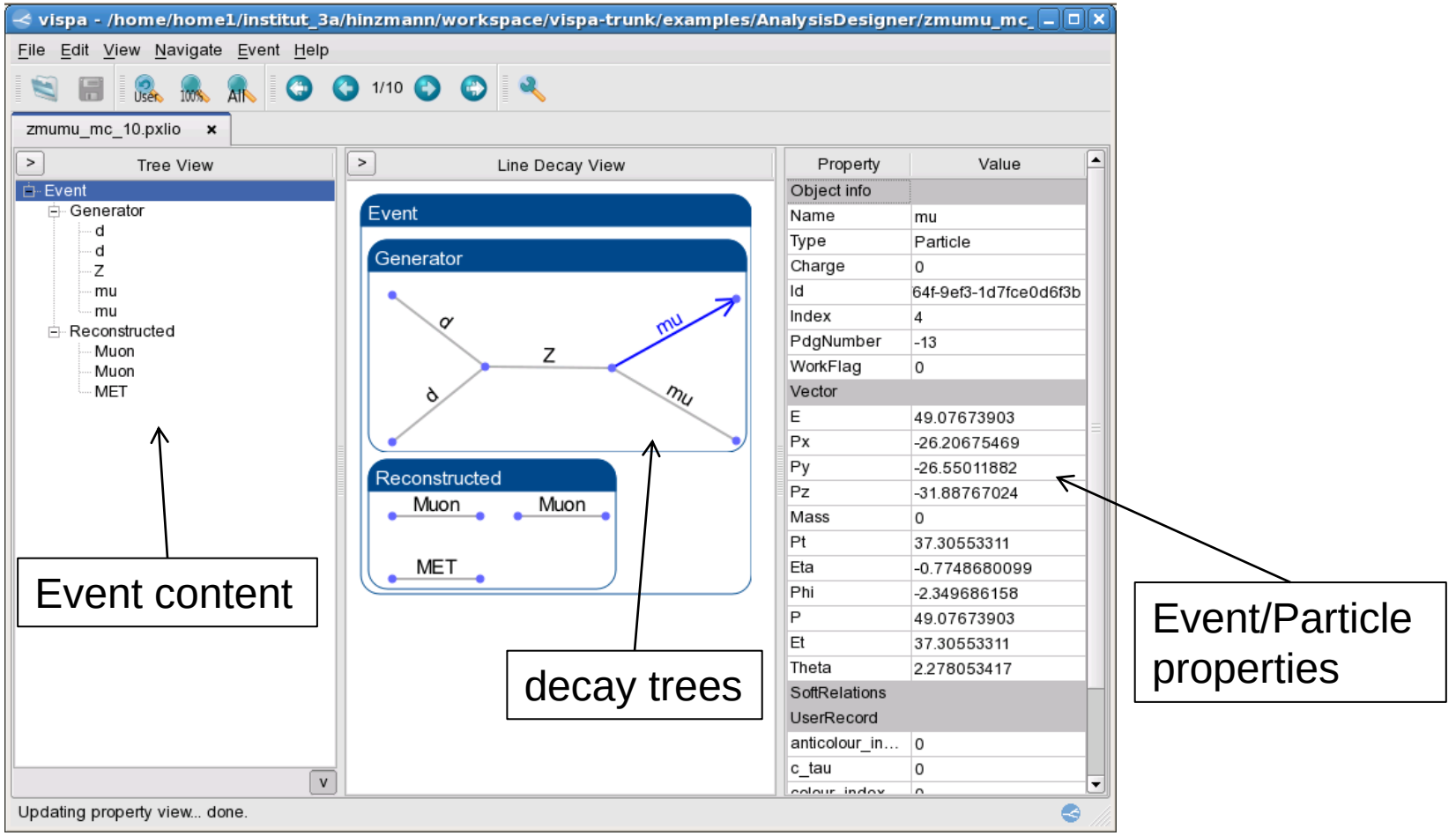
insert using drag+drop

parameters

browse data on double click

edit script on double click

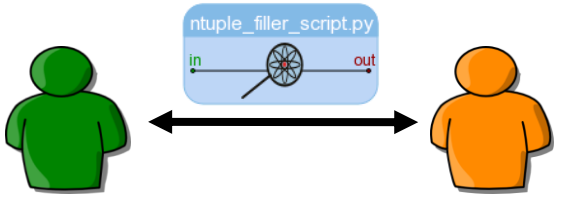
Visual representation of data



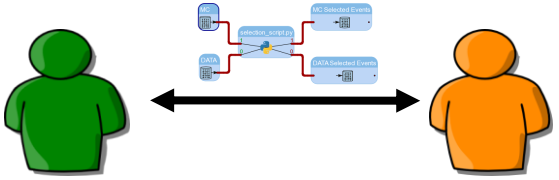
- Browse every single object/parameter in file, ≠ event display, ≠ TBrowser
- Event editor for templates / steering files

Portability of analyses

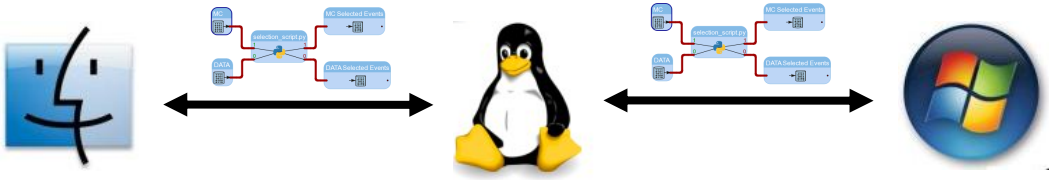
- Exchange of modules:
  - Well defined module interface allows reuse
  - Share common modules within group
  - Outlook: Central database of modules on the web (“Module-store”)



- Exchange of analyses:
  - VISPA allows automatic tar-ball creation



- Exchange between platforms: Linux, Windows and MacOS
  - Graphical platform redesigned in PyQt4 application framework
  - VISPA+PXL downloads for all platforms

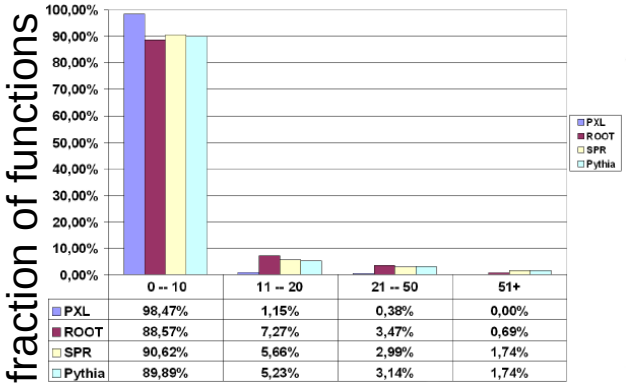


Software quality management

Software quality and performance review in collaboration
with Institute for Software Engineering, RWTH Aachen University

Bug tracking, revision log:
PXL trac
VISPA trac

McCabe metric:
measure for function complexity
98% of functions < 10 in PXL
easy maintainability, low error rate

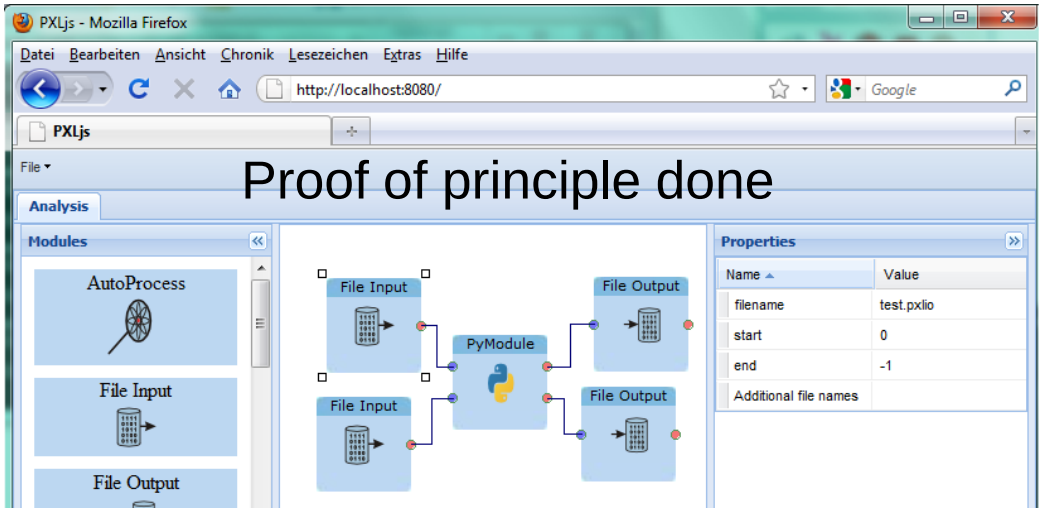


	0 -- 10	11 -- 20	21 -- 50	51+
PXL	98,47%	1,15%	0,38%	0,00%
ROOT	88,57%	7,27%	3,47%	0,69%
SPR	90,62%	5,66%	2,99%	1,74%
Pythia	89,89%	5,23%	3,14%	1,74%

VISPA-Workshop 2008
VISPA-Workshop 2009
VISPA-Workshop 2010

Outlook: VISPA@WEB

- VISPA analyses using web browser
  - No installation needed
  - Modules and data centrally maintained: on the web or within institute
  - Analysis performed on server
  - Good solution for teaching



Proof of principle done


Client (Browser)
Javascript
(draw2d, Extjs, Mootools)


Ajax-Request
JSON


HTTP-Server
Python
PXL

- Development of security concept for user data and modules

Conclusions

- Website: Download, Documentation, Tutorial
  - http://vispa.sourceforge.net
  - http://pxl.sourceforge.net

- Where VISPA is today:
  - Well tested graphical development environment for Physics analysis

- Where VISPA will go:
  - Combination of local physics analysis with web platform