

VISPA - Visual Physics Analysis on Linux, Mac OS X and Windows

Michael Brodski, Martin Erdmann, Robert Fischer, Andreas Hinzmann, Tatsiana Klimkovich, Gero Müller, Thomas Münzer, Jan Steggemann, Tobias Winchen

III. Physikalisches Institut A, RWTH Aachen University, Germany

E-mail: winchen@physik.rwth-aachen.de

The VISPA programm supports particle physicists and astroparticle physicists in their data analysis projects. VISPA combines elements of graphical and textual programming to enable fast development cycles of physics analyses.

*European Physical Society Europhysics Conference on High Energy Physics, EPS-HEP 2009,
July 16 - 22 2009
Krakow, Poland*

1. Introduction

A physics analysis can be regarded as an iterative task consisting of three steps. The analysis has to be prototyped, executed and afterwards verified. Based on the results, the analysis is modified and the process starts again in a new iteration. Today these steps are mostly performed by programming source code (prototyping), running the software on the laptop or computer clusters (executing) and checking the results of the analysis (verifying).

For supporting scientists in each step of this process, we developed VISPA, VISual Physics Analysis [1]. VISPA is designed to combine the convenience of graphical user interfaces with the flexibility of textual programming – a combination which has already been successfully implemented in software for hardware control (e.g. [2]). To account for the variety of computers and operating systems nowadays used for individual tasks, we developed VISPA as platform independent software running on Microsoft Windows, Mac OS X and Linux.

2. The Physics eXtension Library (PXL)

For experiment independent data handling VISPA uses the PXL C++ libraries and their python interface[3, 4]. PXL has been continuously developed since 2006 as the successor of the PAX toolkit [5]. The PXL libraries provide necessary C++ classes and templates for experiment independent high-level physics analysis. It features classes for the representation of physics objects, physics interpretation and also container classes. Using PXL as underlying data structure, VISPA can be used in any experiment with serial data flow. In particular, VISPA can be connected to any high energy physics experiment.

In addition to the established features for High Energy Physics, support for Astroparticle Physics has been included in version 2.1 released in the beginning of 2009. The support of Astroparticle Physics includes classes representing astronomic objects as well as specialized classes representing, e.g., cosmic rays.

3. Analysis Designer and PXL Browser

An analysis on the data is performed in several steps, each represented by an individual module. While modules e.g. for file input and output are already provided, additional modules can be written by the user with the C++ or the Python languages. From the list of individual modules, the analysis is designed by graphically placing and connecting the modules on the workspace using drag and drop. The connections of the modules represent the data flow in the analysis. From the Analysis Designer workspace, the source code of the Python modules and the parameters of the pre-compiled C++ modules are available from within the GUI.

Allowing to use both languages in one analysis, VISPA supports fast prototyping of the analysis using the Python language and subsequently replacing time critical parts with fast code written in C++. This modular concept assists the user in fast prototyping of the analysis and improves the reusability of written source code. The execution of the analysis can be performed directly from the GUI, or on any supported computer in batch mode. Therefore the analysis can be easily transported from the laptop to other machines.

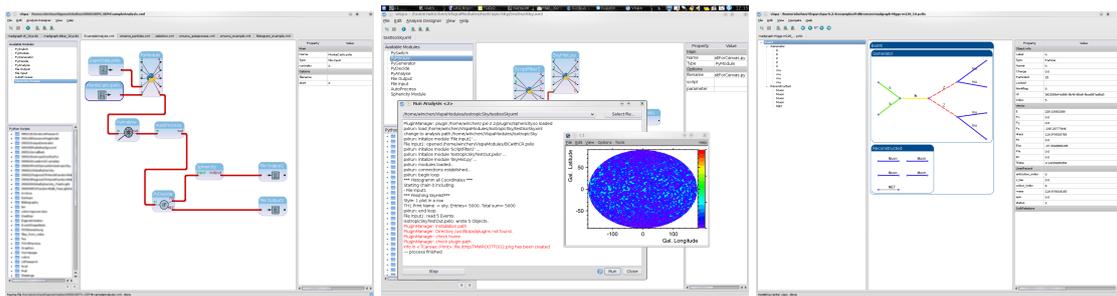


Figure 1: Screenshots of the VISPA GUI. **Left:** In the workspace of the analysis designer individual modules can be arranged to a physics analysis. **Center:** The analysis can be executed directly from the GUI. During execution graphical output can be generated using plotting frameworks like ROOT. **Right:** The data in- and output can be browsed using the PXL Browser component.

The recently improved GUI of VISPA is based on a plug-in mechanism. Besides components for the development and execution of physics analyses, additional plug-ins are available for the visualization of e.g. the structure of high energy physics events, or the properties of cosmic rays in an astroparticle physics analysis. Furthermore plug-ins have been developed to display and edit configuration files of individual experiments from within the VISPA GUI [6].

4. Conclusion

With VISPA and PXL we presented powerful and convenient to use software tools for high-level physics analyses. They allow fast development of physics analysis in High-Energy and Astroparticle Physics featuring modular connection of analysis modules written in C++ and Python.

5. Acknowledgements

We are very grateful for financial support of the Ministerium für Innovation, Wissenschaft, Forschung und Technologie des Landes Nordrhein-Westfalen, the Bundesministerium für Bildung und Forschung (BMBF), the Deutsche Forschungsgemeinschaft (DFG) and the Helmholtz Alliance “Physics at the Terascale”.

References

- [1] *VISPA (Visual Physics Analysis)*, <http://vispa.sourceforge.net>
- [2] *LabVIEW*, <http://www.ni.com/labview/>
- [3] *PXL (Physics Extension library)*, <http://pxl.sourceforge.net>
- [4] O. Actis et al., Proceedings of the XII Conference on Advanced Computing and Analysis Techniques in Physics Research, November 3-7 2008, Erice, Italy
- [5] S. Kappler et al., IEEE Trans. Nucl. Sci. 53 (2006) 506
- [6] M. Erdmann et. al., International Conference on Computing in High Energy and Nuclear Physics (CHEP 2009), To be published in Journal of Physics: Conference Series (JPCS)