Journal of Physics: Conference Series **396** (2012) 052015 doi:10.1088/1742-6596/396/5/052015

A Server-Client-Based Graphical Development Environment for Physics Analyses (VISPA)

H.-P. Bretz, M. Erdmann, R. Fischer, A. Hinzmann, D. Klingebiel, M. Komm, G. Müller, M. Rieger, J. Steffens, J. Steggemann, M. Urban and T. Winchen

III. Physikalisches Institut A, RWTH Aachen University, 52056 Aachen, Germany

E-mail: vispa@lists.rwth-aachen.de

Abstract. The Visual Physics Analysis (VISPA) project provides a graphical development environment for data analysis. It addresses the typical development cycle of (re-)designing, executing, and verifying an analysis. We present the new server-client-based web application of the VISPA project to perform physics analyses via a standard internet browser. This enables individual scientists to work with a large variety of devices including touch screens, and teams of scientists to share, develop, and execute analyses on a server via the web interface.

1. Introduction

The use of distributed resources for various applications experienced a rapid development in the past decades. Networks connect substantial computing power and large scale data storage. Mostly pre-defined algorithms such as search engines are accessible worldwide to satisfy individual user requests.

In advanced physics data analyses, pre-defined algorithms are not sufficient. For successful work of scientists, development of individual algorithms for data treatment is necessary beyond access to adequate computing resources. A convenient worldwide access to data analyses which are hosted and executed on a server, and are programmable through a web interface constitutes a new approach.

In this contribution, we present the alpha-version of a system enabling the typical analysis development cycle of (re-)designing, executing, and verifying results in a standard internet browser. This new approach is based on the desktop version of the Visual Physics Analysis (VISPA) project which has been successfully used in various high-energy and astroparticle physics data analyses [1].

2. The Visual Physics Analysis Project

The VISPA project presents a graphical development environment for physics analyses. As the underlying C++ analysis toolkit the Physics eXtension Library (PXL) [2] is used.

Basic guidance to the project is given by several software paradigms. In his development, the user combines graphical representations of modules to create the analysis structure. The modules are textually implemented using object-oriented programming for data treatment based on the objects provided by the PXL toolkit.

On top of the application programming interface of PXL, a Python [3] interface is provided so that analyses can be developed using both C++ and Python code. For graphical representation of results, e.g. histograms and sky maps, external packages such as ROOT [4] and matplotlib [5] can thus be easily connected to VISPA.

3. Technical Realization of the VISPA Server-Client System

For the realization of the server-client approach we use technology typically found in Web 2.0 applications. On the client side, a standard internet browser is required which supports JavaScript and HTML5 capabilities. For displaying the VISPA specific graphical user interface, the DOJO framework [6] is used which supports standard screens as well as touch devices. The corresponding files are provided by the server.

On the server side, the new VISPA platform is based on the Python web framework CherryPy [7]. The key ingredients of the platform incorporate multi-user capability, a virtual file system, and a system to efficiently perform user actions and to execute analyses using multiple processes. The individual server components are described in the following sections.

For the communication between the client and the server, the Asynchronous JavaScript And XML (AJAX) [8] concept is used.

Note that although the server-client system is intended to run on separate computers, the possibility to install the program locally on a single computer is preserved. After installation of the packages [2, 3, 4, 5, 7, 9, 10, 11, 12], and download of the Python based VISPA server software, the server process can be started from a standard local Python installation. Access to the user analysis is realized by the local internet browser using a local address (e.g. http://localhost:4282).

3.1. User Interface for Physics Analysis Development

For performing an analysis development cycle, the current user interface consists of i) a designer for implementing the analysis structure, ii) an editor for programming algorithms, and iii) a browser to inspect the content of every object contained in a data file.

The various aspects of the user interface have been implemented as extensions, and are registered at the VISPA platform accordingly. The analysis designer and the data file browser are based on the C++ physics analysis toolkit PXL.

In Fig. 1, we show the analysis structure of a jet selection procedure in particle collision processes. Alternatively simulated and measured data are passed to the input of the module "Jet Selection". In every collision event, the jet with the largest transverse momentum p_T passes or fails a threshold of $p_T > 50$ GeV. For verification of the selection procedure, all resulting output streams are stored. In the property view of the "Jet Selection" module on the right hand side, the value of the p_T threshold can be varied directly.

In order to inspect and modify the source code of a module, an editor based on the JavaScript library CodeMirror [13] is provided which includes features like syntax highlighting (Fig. 2).

To inspect the contents of a data file, a browser is provided. In Fig. 3, a simulated Z-boson event from a proton-proton collision is shown. The data file contains generator information on the production and decay of the Z-boson as well as reconstructed properties of the final state measured in a detector. The figure shows the property view with the particle kinematics after selecting the generated Z-boson.

3.2. Core Components of the VISPA Platform

The analysis focused extensions mentioned above access several core components of the VISPA platform which provide functionalities necessary for the server-client approach.

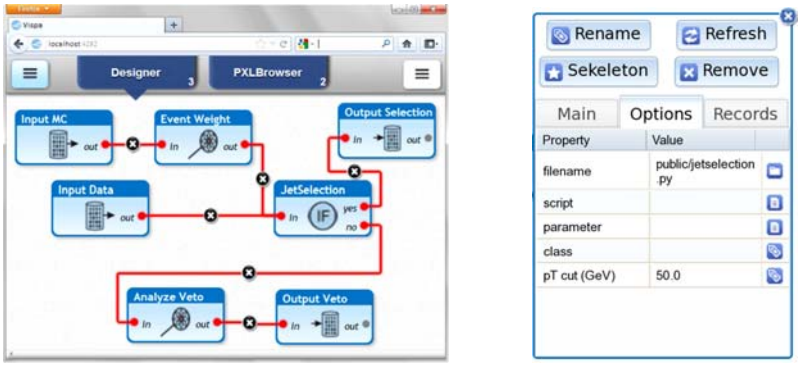

Figure 1. Analysis designer in an internet browser presenting an exemplary data selection procedure. The configuration dialog on the right hand side refers to the module "Jet Selection".

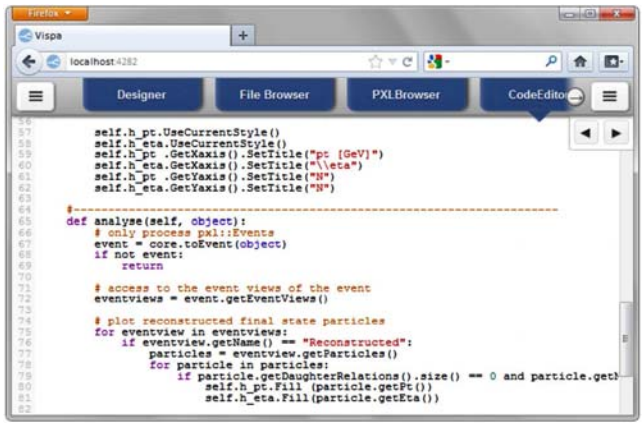

Figure 2. Editor in an internet browser exhibiting exemplary code for particle data analysis.

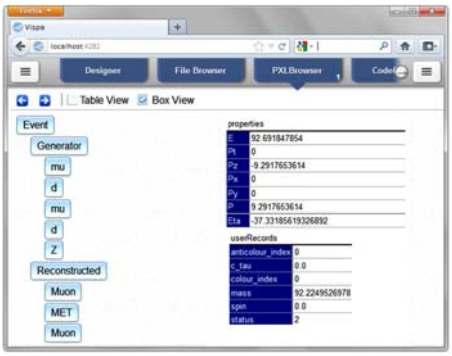

Figure 3. Data inspection in an internet browser showing a simulated event of a Z-boson production and decay.

User Management. For handling multi-user access to the system, a mandatory user registration process is provided. The user data are stored in an SQL data base containing default permissions to develop and execute an analysis.

File System. Users have access to a virtual file system providing i) a directory containing experimental data and example analyses, ii) a personal directory visible to the individual user only, and iii) a personal directory for sharing analyses and results with colleagues. Data files, Python scripts, and resulting plots can be downloaded from the server, or uploaded respectively.

Multiple Processes. In order to account for the computing demands of multiple users in terms of security, scalability, and user separation, the frontend web server creates additional processes for executing analyses based on the Remote Procedure Call (RPC) system [14]. Communication is performed via the XML-RPC protocol. Therefore, these additional processes can run either on the same machine as the web server, or on any other machine within a network. This is intended to provide sufficient computing power for user actions and analysis execution depending on the usage of the system. For these machines no additional software is required compared to the installation on a single computer described above.

4. Conclusions

We have presented the alpha version of a program enabling scientists to develop and execute their individual physics data analysis in a standard internet browser on a remote server machine. While all necessary features already exist to perform analysis work in this client-server system, the beta-version will further improve the user experience in physics analysis development.

The potential of the presented server-client approach is comprehensive. For example, scientists are able to share their analysis concepts and algorithms with colleagues through the web interface. A new level of transparency may be offered in publications by providing web access to the corresponding analysis. Students may get access to analysis examples exploring public experiment data with professional tools that are standard in today's physics analyses.

Acknowledgments

We wish to thank the organizers of the CHEP2012 conference for their kind support of a live demonstration of the VISPA server-client system. This work is supported by the Ministerium für Wissenschaft und Forschung, Nordrhein-Westfalen, the Bundesministerium für Bildung und Forschung (BMBF), and the Helmholz Alliance Physics at the Terascale. T. Winchen gratefully acknowledges funding by the Friedrich-Ebert-Stiftung.

References

[1] Bretz H-P *et al.* 2012 A Development Environment for Visual Physics Analysis, *JINST* **7** T08005, doi:10.1088/1748-0221/7/08/T08005, arXiv:1205.4912v2, http://vispa.physik.rwth-aachen.de
[2] Erdmann M *et al.* 2012 Physics eXtension Library (PXL), http://vispa.physik.rwth-aachen.de/Documentation
[3] Python Software Foundation 2012 http://www.python.org
[4] Brun R and Rademakers F 1996 *Nucl. Inst. & Meth. in Phys. Res.* A **398** 81-86, http://root.cern.ch
[5] Hunter J 2012 matplotlib, http://matplotlib.sourceforge.net
[6] Russell M 2008 *Dojo: The Definitive Guide* (O'Reilly Media, Sebastopol, CA, USA), http://dojotoolkit.org
[7] CherryPy Team 2012 CherryPy - A Minimalist Python Web Framework, http://www.cherrypy.org/
[8] Garrett J J 2005 AJAX: A New Approach to Web Applications, http://www.adaptivepath.com/ideas/ajax-new-approach-web-applications
[9] Bayer M 2012 Mako Templates for Python, http://www.makotemplates.org/
[10] Pointer R 2011 Paramiko: Python SSH module, http://www.lag.net/paramiko/
[11] Hipp D R 2012 SQLite, Hwaci - Applied Software Research, http://www.sqlite.org
[12] SQLAlchemy authors and contributors 2012, http://www.sqlalchemy.org
[13] Haverbeke M 2011 CodeMirror, http://codemirror.net/
[14] Thurlow R 2009 RPC: Remote Procedure Call Protocol Specification Version 2, http://tools.ietf.org/html//rfc5531