EVENT VISUALISATION FOR THE ATLAS EXPERIMENT - THE TECHNOLOGIES INVOLVED

MAXA Z., COUCHMAN J., DROHAN J., KONSTANTINIDIS N. (University College London)
DREVERMANN H., PETRUSCA D. (CERN)
LU Q., THOMAS J., WATKINS P. (University of Birmingham)
TAYLOR G. (University of California at Santa Cruz)
HAAS A. (University of Columbia)
JANSEN E., KLOK P., TIMMERMANS C. (University of Nijmegen)

Abstract

We describe the design of Atlantis, an event visualisation program for the ATLAS experiment at CERN, and the other supporting applications within the visualisation project, mainly focusing on the technologies employed. The ATLAS visualisation consists of several parts with Atlantis being the central application. The main purpose of Atlantis is to help visually investigate and intuitively understand complete ATLAS events.

Atlantis is a stand-alone graphical application written entirely in Java, using Java/Swing 2D API, XML parsers and Apache/XMLRPC for network communication with Athena, the ATLAS software framework. The event data, in XML format, is produced by a dedicated interface called JiveXML running within the Athena framework. Atlantis reads in the data either from files (offline mode) or via a network connection in the online mode of JiveXML. In the online mode, the data is transferred on request from a C++ XMLRPC server running within JiveXML to Atlantis acting as a XMLRPC client.

The Atlantis user is also able to steer the Athena framework over a network connection directly from Atlantis. Atlantis makes remote calls to a XMLRPC Python server started at the interactive Athena Python prompt. This server receives the Athena commands and executes them as if typed locally.

INTRODUCTION

Atlantis is an event display for the ATLAS experiment at CERN. When the LHC starts running in 2007, it will collide protons with a center of mass energy of 14TeV at a bunch crossing rate of 40MHz. At the design luminosity of $1034\,cm^{-2}\,s^{-1}$ there will be approximately 23 interactions per bunch crossing. Atlantis aims to provide easy, fast, error free visual investigation and physical understanding of these complicated events. Furthermore, Atlantis is supposed to be instrumental for development and debugging of reconstruction algorithms and an important tool displaying data during commissioning of the detector subsystems.

Atlantis, written entirely in Java language, is a stand-alone application which employs a variety of 2D data-oriented projections and uses simplified geometry of the ATLAS detector.

ATLANTIS VISUALISATION STRUCTURE

Picture 1 shows the dataflow from the ATLAS detector to the Atlantis event display. Currently, as the real data is not available, the offline software framework Athena takes on input full simulation data (digi files) or reduced-size AOD or ESD files containing already certain reconstructed objects. Since Atlantis is a stand-alone application it needs an
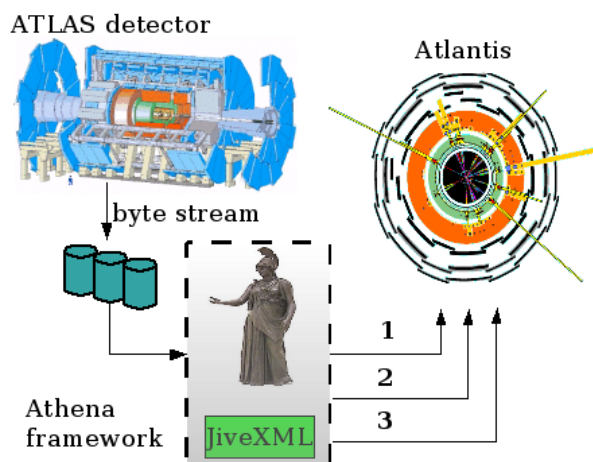

Figure 1: Atlantis visualisation structure.

interface in the framework, this task performs the JiveXML package which runs in Athena. JiveXML converts event data from the transient event store of Athena into XML format which Atlantis understands. There are 3 modes of communication between the Athena framework and the Atlantis event display:

1. offline mode

2. online mode

3. interactive mode

They are all covered in the next section.

Atlantis Program Overview

Atlantis is a Java application employing the Java 2D libraries for graphics and Swing libraries for the Graphical User Interface (GUI). Both DOM[1] and SAX[2] XML parsers are used to process input XML files (event data, ATLAS geometry and Atlantis configuration). XMLRPC[3] technology is utilised for the network communication between Atlantis and the Athena framework. The JavaHelp package maintains the online help system in the application.

The Atlantis application consists of two main windows. The canvas (figure 2) is a layered panel, allowing multiple projections to be displayed on the screen at once (canvas is divided into different subwindows), with varying size and position. These displays can be adjusted within the GUI (the control window). The two upper subwindows show
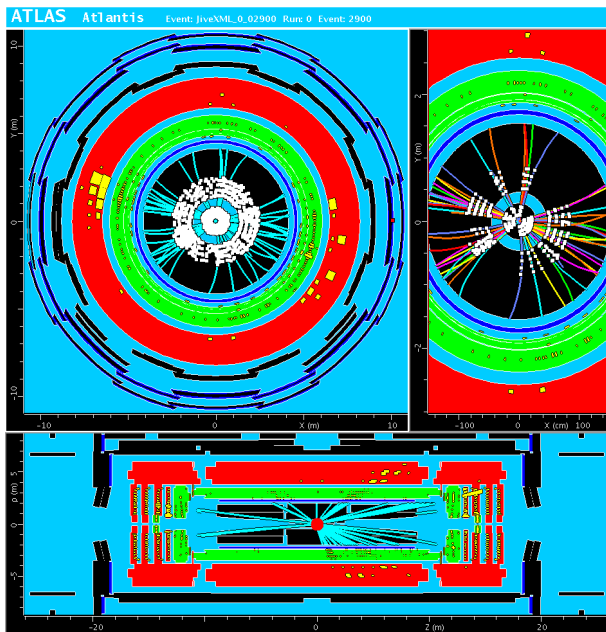

Figure 2: Atlantis canvas with the event data displayed.

XY projection, the right subwindow is zoomed into the inner detector area. The left subwindow as well as the bottom one, which shows ρZ projection, have fish-eye transformation applied.

COMMUNICATION WITH THE ATHENA FRAMEWORK

Offline mode

Athena and the Atlantis event display run independently of each other. JiveXML in Athena produces XML event files, the user transfers them to the Atlantis location and browse them in the event display.

Online mode

The Athena framework and Atlantis run simultaneously, presumably on different machines. The Atlantis user or multiple Atlantis users are able to connect to the running instance of Athena and retrieve event data over network, provided that some reconstructed event is available at Athena. Generally, this is 1-to-many communication model. XMLRPC technology is used for network communication which takes place between XMLRPC C++ server running in a posix thread in JiveXML package and Atlantis which acts as Java XMLRPC client. ULXMLRPCPP[4] implementation of XMLRPC specification is used on the C++ JiveXML side and Apache Java XMLRPC is employed on the Atlantis side.

There is a possibility of retrieving the event data periodically after setting up a timer in Atlantis. New event data is requested repeatedly after a specified delay.

This scenario gets rid of the intermediate step of producing many event files in the offline mode. Nonetheless, if the user spots an interesting event, it's possible to save the obtained event data locally into the file. Online mode of running JiveXML - Atlantis already proved useful for monitoring purposes.

Interactive mode

The Athena framework is set up and accessed exclusively by a particular user directly from Atlantis in this 1-to-1 communication. The basic facility allowing interactive communication between Athena and Atlantis is the interactive Python prompt of Athena. This prompt is avail-
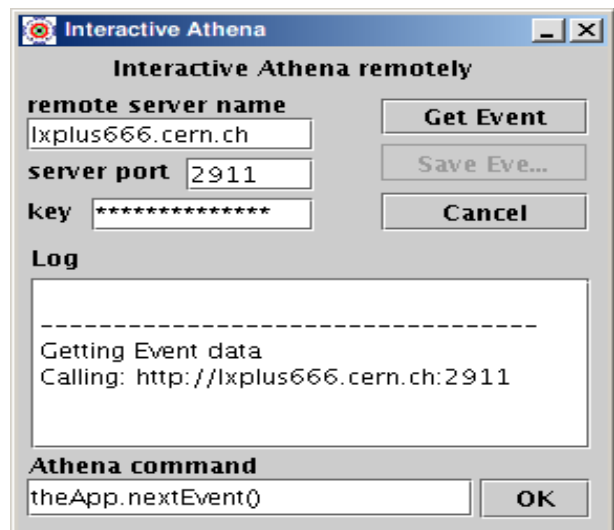

Figure 3: Interactive Athena dialog in Atlantis.

able when the user runs the Athena framework in the interactive mode. Athena commands can be typed on this prompt permitting to steer the framework on command by

---

[1] Document Object Module
[2] Simple API for XML
[3] eXtensible Markup Language Remote Procedure Call
[4] Ultra Lightweight XMLRPC for C++

command basis. Atlantis - interactive Athena communication is merely extension of this Athena feature into the Atlantis dialog shown in the figure 3. `InteractiveServer` is a Python script executed on the Athena prompt and it is in fact an interface between this prompt and the Atlantis dialog (shown in 3). `InteractiveServer` functions as a Python XMLRPC server listening for Athena commands and event data requests from the Atlantis dialog. The Athena commands are executed at the Athena side as if they were typed locally on the interactive prompt. Results are forwarded back to the Atlantis user.

This feature allows the user to perform the interactive analysis directly from the Atlantis event display. The Atlantis user commands to reconstruct next event, retrieve the event data to Atlantis, execute the algorithms in Athena, etc. For example, the user sees a few reconstructed tracks which seem to originate from a common vertex. After selecting these tracks, their indices are sent to Athena where the vertex fitter is executed to calculate the vertex. If found, the reconstructed secondary vertex is shipped along with the event data to Atlantis to be examined.

CONCLUSION

Atlantis is an event display tailored for the visual investigation of complete ATLAS events. Atlantis reads in XML event data produced by JiveXML algorithm running in the ATLAS offline software framework Athena. The event data may be obtained as XML files for offline use in Atlantis or online over network. Atlantis - Athena interactive communication allows the Atlantis user to drive the Athena framework while performing interactive physics analysis.

The Atlantis program is available to download from `www.cern.ch/atlantis`, along with plenty of documentation and a picture database. The application itself has comprehensive online help system. User support is available from the Atlantis team in the `atlantis.support@cern.ch` mailinglist.

REFERENCES

[1] H.Drevermann D.Kuhn B.S.Nilsson, "Is there a Future for Event Display?", Proc. of 1992 CERN School of Computing, L'Aquila, Italy, CERN Yellow Report 93-03.

[2] N. Konstantinidis H. Drevermann, "Fast Tracking in Hadron Collider Experiments", ACAT, October 200

[3] `http://www.cern.ch/atlantis` - Atlantis event display website