Event Display in ATLAS *

J. Hřivnáč *Institute of Physics, Academy of Sciences, Prague, Czech Republic*

Abstract

The current status of the ATLAS event Display development is described. Emphasis is put on the development of the new software based on an Object Oriented paradigm and written mainly in the C++ language.

Keywords: ATLAS, graphics, virtual reality

1 Status

This report describes the status of the new event display development in the ATLAS collaboration. It does not cover the implementation of the graphics in the current software (Slug[1], Dice[2], Atrecon[3]) which is based on ATLAS-extended Fortran 77 and Geant 3 graphics. While the current implementation is still used it is considered not suitable for the future.
The new ATLAS software is based on the Object Oriented paradigm and uses C++ as the main implementation language. Most work is going on in the ATLAS-MOOSE (RD41) project[4]. The graphical part of this software is based on the OpenGL and VRML packages and is supposed to use Open Inventor in the near future. The development of the graphics side of the new ATLAS software is coordinated by the ATLAS Graphics group[5], which has been established in the summer of 1996. This group has produced the first (working) version of the User Requirements Document[6] and is working on the top-level design of the event display package.

2 User Requirements

The working version of the User Requirements Document[6] has been produced by the ATLAS Graphics group. It emphases the modularity and extensionability of the graphics software, which will have to satisfy a broad range of users for a long time. It is worth bearing in mind that the ATLAS reconstruction software as well as other used packages will change substantially during the period of the graphics software existence. Naturally, all Requirements defined in the connected higher-level URDs should be fulfiled.
The draft of the URD document is divided into three main sections:

- **Functional Requirements:** Each object from the reconstruction software domain should be represented by its graphical partner which should be able to be displayed in various ways through graphical views. It should be possible to perform various operations on the graphical objects (in views). These operations can be purely graphical, but can also access properties of the real objects and change them. The event display should be interfaced to the reconstruction program and to the data database. It should be possible to access, search and change the database from within the event display. The event display should be widely customisable.
- **Operational Requirements:** The event display should run within the whole reconstruction and simulation package. It should be easy to add new graphical objects and views. Multi-user access with priorities should be available.
- **System Requirements:** The event display should be capable of running over the network on all used hardware platforms and Operating Systems. No special graphics hardware should be required, but if it is available, event display should profit from it. It should be possible to use event display in the on-line environment.

* http://www-hep.fzu.cz/ hrivnac/VRML/talks/HEPVis.96/

3 Main Principles

The main principles for the development of the ATLAS graphics system inherit from the top-level ATLAS software development principles. Among them, the most important are:

- Using de-jure or de-facto standards (industry-wide or HEP-wide) if they exist.
- Using existing packages, if they exist.

Other principles clearly follow from these two:

- Using Object Oriented paradigm with C++ as the main implementation language.
- Using OpenGL and OpenInventor as the main visualisation libraries.

4 Existing Prototypes

There are already several running prototypes which may serve as a “proof of the concept” of the afore-mentioned methods. They have been mainly developed in the ATLAS-MOOSE collaboration.

4.1 ATLAS Muon System

The ATLAS Muon System[7] is very simple application, which creates static VRML representation of the muon chambers. This 3D representation is created from the muon chamber database. The whole system enables the study (along with the muon system itself) of the usefulness of the VRML representation of the graphical objects.
See Figure 1.

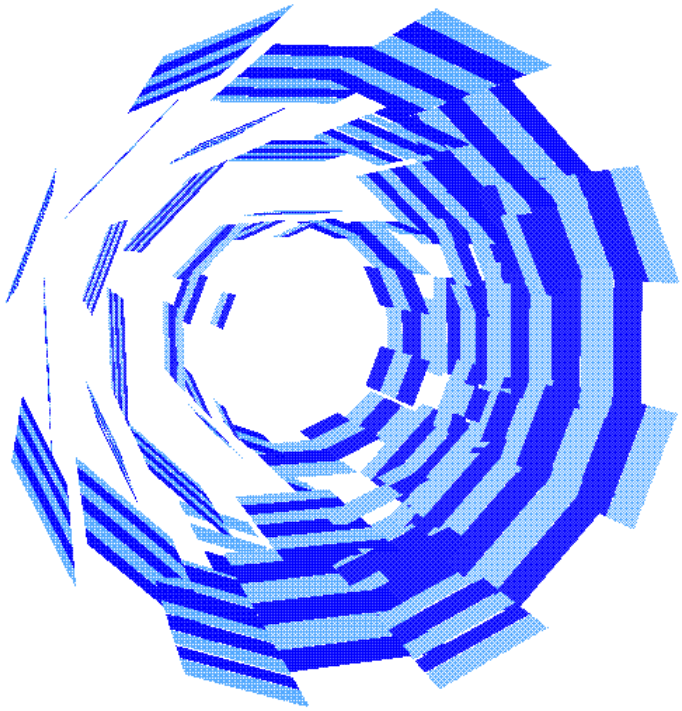

Figure 1: ATLAS Muon Chambers

4.2 Virtual ATLAS

Virtual ATLAS[8] is a standalone application (class library) which enables users to easily create VRML representation of any 3D object. Its first implementation has been based on the Eiffel Moose

prototype (MPATREC [9]). The current implementation is completely in C++ and is uses the main design principles of the ATLAS event display. See Figure 2.

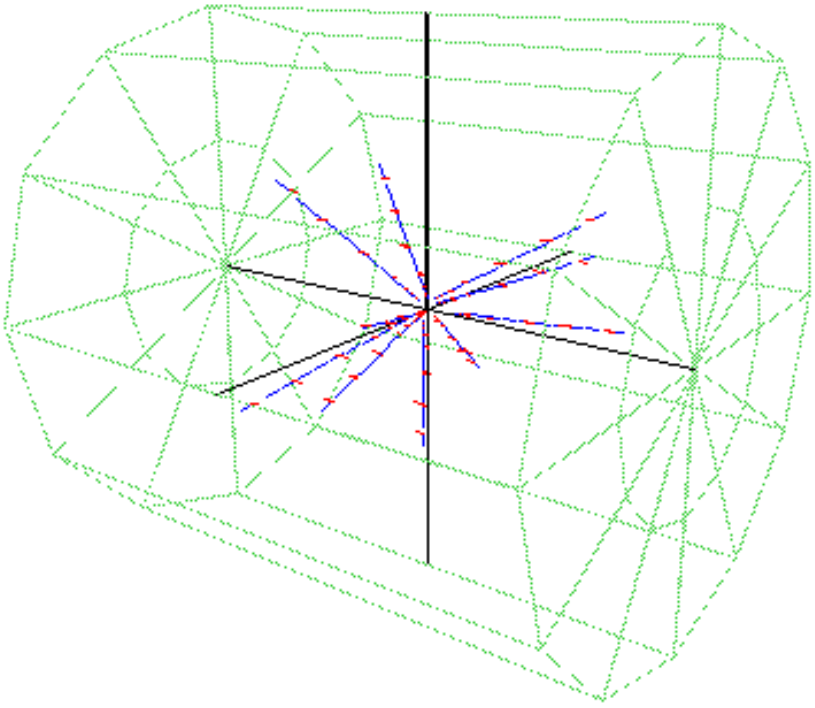

Figure 2: Virtual ATLAS

4.3 ARVE

ARVE[10] (*ATLAS Reconstruction and Visualisation Environment*) is currently the central project for the future ATLAS reconstruction software. It's written completely in C++ and as Geant 4[11] package is not yet available ARVE is based on the Gismo framework[12].
Current implementation of the ARVE runs both on the PC (Windows'95) and HP (Unix) platform. It contains simple event display and GUI, which is now implemented for ATLAS muon chambers. ARVE event display is capable of serving 2D and simple 3D view of the detector and simulated particles and their interactions with the medium. It is also able to create VRML representation of the graphical scene. See Figure 3.

5 Current Design

The event display design is currently divided into several subdomains:

- **Control Structure** deals with the system, how graphical objects and views work together and how they interact with the rest of the program.
- **Scenes** are various implementations of the graphical views (basic 2D, VRML 1.0 and 2.0, special projections,...).
- **Graphical Objects** will be mostly designed by the authors of the corresponding real objects.
- **Collaboration with other domains** (top-level control structure, database, GUI,...).
- **Interface to the current software** should be available because current software will be still used for several years.

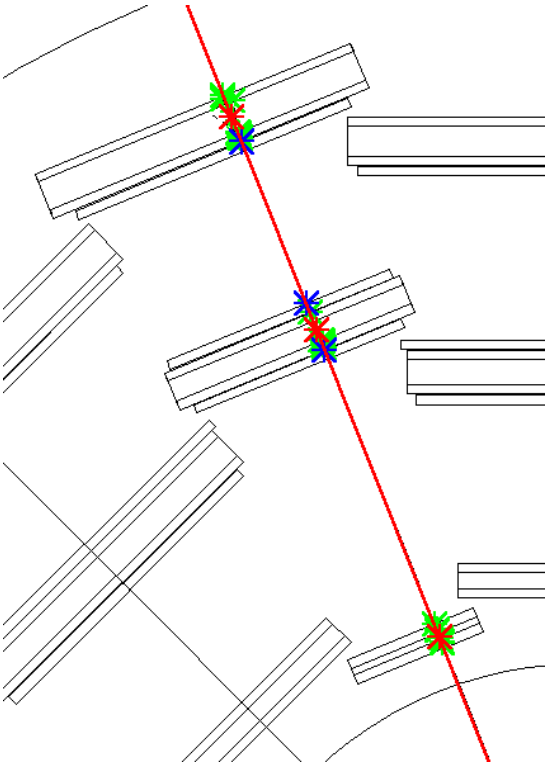

Figure 3: ATLAS Reconstruction and Visualisation Environment

References

1 http://atlasinfo.cern.ch/Atlas/GROUPS/SOFTWARE/DOCUMENTS/simulation.html#slug

2 http://atlasinfo.cern.ch/Atlas/GROUPS/SOFTWARE/DOCUMENTS/simulation.html#dice

3 http://atlasinfo.cern.ch/Atlas/GROUPS/SOFTWARE/DOCUMENTS/reconstruction.html#atrecon

4 http://atlasinfo.cern.ch/Atlas/GROUPS/ATLASOO/Home_atlasOO.html

5 http://atlasinfo.cern.ch/Atlas/GROUPS/GRAPHICS/

6 http://atlasinfo.cern.ch/Atlas/GROUPS/GRAPHICS/Texts/EventDisplayURD.html

7 http://www.cern.ch/Atlas/ENGINEER/detector/amdb/VRML/

8 http://www-hep.fzu.cz/ hrivnac/VRML/AtlasInVRML.html

9 http://moose.cern.ch/moose/prototype-2/prototype-2.html

10 http://www.phys.washington.edu/%7Eburnett/atlas/

11 http://wwwcn1.cern.ch/asd/geant/geant4.html

12 http://www.phys.washington.edu/ burnett/gismo/