

PAPER • OPEN ACCESS

## Event visualization in ATLAS

To cite this article: R M Bianchi *et al* 2017 *J. Phys.: Conf. Ser.* **898** 072014

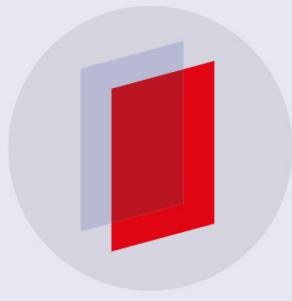
View the [article online](#) for updates and enhancements.

### Related content

- [Rotation, Reflection, and Frame Changes: Computer graphics visualization](#)  
R M Brannon

- [Raw-data display and visual reconstruction validation in ALICE](#)  
M Tadel

- [The future of PanDA in ATLAS distributed computing](#)  
K De, A Klimentov, T Maeno et al.



**IOP** | **ebooks**<sup>TM</sup>

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the collection - download the first chapter of every title for free.

# Event visualization in ATLAS

**R M Bianchi<sup>1,a</sup>, J Boudreau<sup>1</sup>, N Konstantinidis<sup>2</sup>, A C Martyniuk<sup>2</sup>, E Moyse<sup>3</sup>, J Thomas<sup>4</sup>, B M Waugh<sup>2</sup>, D P Yallup<sup>2</sup>  
on behalf of the ATLAS Collaboration**

<sup>a</sup> Corresponding author

<sup>1</sup> University of Pittsburgh, Department of Physics and Astronomy, Pittsburgh PA 15260, US

<sup>2</sup> University College London, Department of Physics and Astronomy, London WC1E 6BT, UK

<sup>3</sup> University of Massachusetts Amherst, Physics Department, Amherst, MA 01003-9337, US

<sup>4</sup> University of Birmingham, School of Physics and Astronomy, Birmingham B15 2TT, UK

E-mail: [riccardo.maría.bianchi@cern.ch](mailto:riccardo.maría.bianchi@cern.ch)

**Abstract.** At the beginning, HEP experiments made use of photographic images both to record and store experimental data and to illustrate their findings. Then the experiments evolved and needed to find ways to visualize their data. With the availability of computer graphics, software packages to display event data and the detector geometry started to be developed.

Here, an overview of the usage of event display tools in HEP is presented. Then the case of the ATLAS experiment is considered in more detail and two widely used event display packages are presented, Atlantis and VP1, focusing on the software technologies they employ, as well as their strengths, differences and their usage in the experiment: from physics analysis to detector development, and from online monitoring to outreach and communication. Towards the end, the other ATLAS visualization tools will be briefly presented as well. Future development plans and improvements in the ATLAS event display packages will also be discussed.

## 1. Event displays — History and usage

HEP experiments have always needed event displays and visualization tools. At the beginning, in the analogue era, photographic emulsions were used with a dual goal: to detect particles and to store the results. The physics was inferred from the measurements taken from the pictures, as well. Figure 1a shows a picture of the photographic emulsion which recorded the discovery of the positron, by C. Anderson in 1933 [1].

Then, the digital era came, the technology evolved, experiments started taking a growing amount of data at a higher rate, and photographic pictures were not appropriate anymore. So physicists started to develop software tools to visualize event data (also called *event displays*), making use of the advances in computer graphics. Figure 1b shows the event display of one of the three-jets events which led to the discovery of the gluon [2].

Nowadays event displays are still used to see what happens inside the experiment. But not only. There are many types of data in a HEP experiment, coming from different sources. There are experimental data, collected by the experiments. The ATLAS Experiment [3] running at the Large Hadron Collider (LHC) collects hundreds of millions of proton-proton collisions per second, which are then selected, filtered, stored and analyzed. There are also computer generated data, which simulate different physics processes and the detector response; they are compared with the experimental data to better estimate the detector and the analysis processes, or to infer



Content from this work may be used under the terms of the [Creative Commons Attribution 3.0 licence](#). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

the performance and the possible experiment reach in upgrade studies. There is the detector geometry, that is the detailed description of all the particle detectors and their infrastructure used in HEP experiments; detectors like ATLAS are described through the usage of geometry libraries (like GeoModel [4]); and the resulting description is used to simulate the detector response or for detector development studies.

All these data need to be visualized, for many different tasks; to check the event selection of a physics analysis; to monitor the data taking in the experiments' control rooms; to verify the shape and the placement of a piece of the detector in a simulation; to prove the correctness of a reconstruction algorithm; to inspect the description of the detector geometry; to test the validity of a simulation; to support the development of a new sub-detector or of a new piece of the experimental apparatus; to document results, findings and discoveries in scientific papers, press releases and technical reports for the funding agencies; as an educational tool, to effectively illustrate complex subjects in textbooks and masterclasses; or, as an outreach tool, to simply let the public see what happens inside the detector of an HEP experiment with a compelling and engaging picture which let them learn about particle physics while looking at images from the frontier science.

Different visualization tools have been developed for the needs of the ATLAS experiment, all of them targeted to different usages and users: two general-purpose tools, which are the focus of this paper, and three specific tools, which will be briefly presented in the last section.

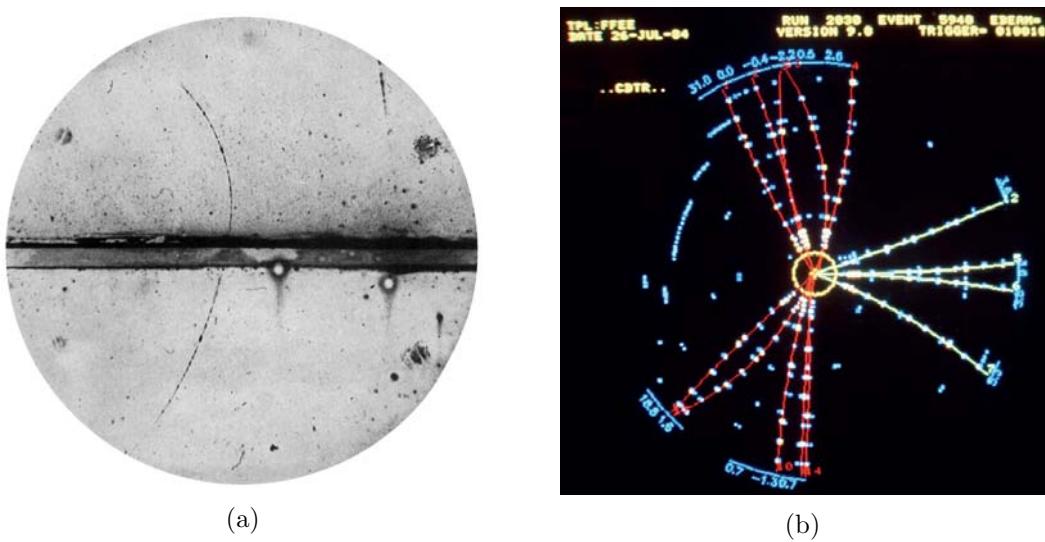


Figure 1: Historical event displays: (a) the photographic picture which induced the discovery of the positron (1933) [1]; (b) the event display of one of the three-jets events which led to the discovery of the gluon (1979) [2].

## 2. Event visualization in ATLAS

The ATLAS Collaboration developed different event display tools, targeting different needs and different end users. One of them is part of the experiment's core software and can natively access all ATLAS data. The others are stand-alone applications which make use of intermediate formats generated from the official ATLAS data. They are summarized in table 1. This paper will focus on the two general-purpose tools, Atlantis [5, 6] and VP1 [7, 8]. In the last section other three tools targeting more specific needs will be briefly presented as well.

	<i>Name</i>	<i>Technologies</i>	<i>Target</i>	<i>Notes</i>	
<i>Desktop</i>	<b>Atlantis</b>	Java	general-purpose, focused on physics analysis	cross-platform	stand-alone application
	<b>VP1</b>	C++, Coin3D (Open Inventor), SoQt, Qt	general-purpose	integrated in the experiment's framework	
	<b>Persint</b>	Fortan, OpenGL	muon system	maintained but not actively developed anymore	
<i>Web-based</i>	<b>TADA visualizer</b>	Javascript, WebGL (three.js)	on-line analysis	prompt	in development
	<b>Tracer</b>	WebGL (three.js)	outreach & education		in development

Table 1: Summary of the different tools for the visualization of event data (*event displays*) used in the ATLAS Experiment.

### 3. Atlantis — A cross-platform tool for physics analysis

Atlantis [5, 6] is a Java stand-alone application aimed at visualizing physics objects (like jets, particles and energy deposits) for physics analysis. As input data, it uses a custom intermediate XML format, generated from the ATLAS data. It shows a simplified geometry of the ATLAS detector, in two dimensions.

Figure 2a shows a screenshot of the Atlantis user interface, with the tools to generate the display and to interact with it, selecting physics objects and applying kinematic cuts. Figure 2b shows an example of an Atlantis event display, with different views and projections of the same event combined and the simplified geometry overlay.

#### 3.1. Architecture

Atlantis uses Java both as graphics engine and for the graphic user interface. Being a Java-based external application, Atlantis has the advantage of being cross-platform. Physicists working on data analysis are hence able to install it directly on their computers, without constraints on the operating systems they use.

On the other hand, Atlantis is not able to access ATLAS data natively. A dedicated package, called **JiveXML**, produces the custom XML data files which are used as input in Atlantis, from the experiment's data file. As shown in figure 4, where the whole ATLAS Event Data Model (EDM) is presented, JiveXML can produce the XML file both from the RAW and the ESD (*Event Summary Data*) data files.

#### 3.2. Current development

Atlantis is currently being updated for the high-pileup events. The challenge will be the visualization of a large number of vertices ( $\approx 120$ ) in very busy events, allowing full association between detector hits and physics objects. For that, a number of optimizations are being implemented; for instance, on the color coding of the vertex quality and on the display of trigger conditions and objects; also, the XML data format and the data access interface are being optimized and the development of a slim XML file is started. Moreover, tests about different formats and access methods to the experiment's data are foreseen.

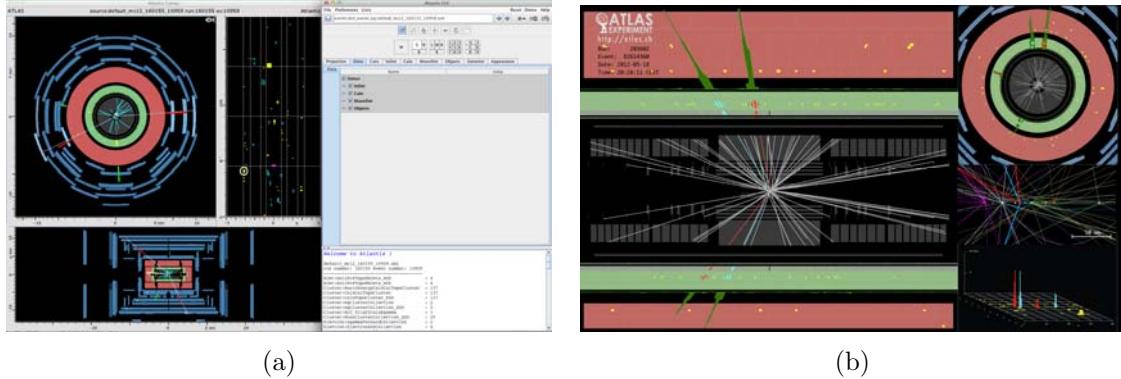


Figure 2: The Atlantis event display. (a) The user interface let users select physics objects applying kinematic cuts, and set display options; (b) Atlantis shows a simplified ATLAS geometry, and it is focused on visualizing physics objects for physics analysis, like tracks, jets, vertices, hits and calorimeter energy deposits.

#### 4. VP1 — Visualizing all ATLAS data

VP1<sup>1</sup> [7, 8] is a general purpose tool, integrated in the experiment’s framework, **Athena** [9]. Being part of the core software of the experiment, VP1 is able to access all sort of ATLAS data, from the files containing the raw experimental data to all intermediate formats storing filtered and reconstructed data, from the on-line services providing calibration constants and conditions data about the data taking to the database storing the geometry of the detector description.

The VP1 event display shows all data in a 3D environment, letting users interact with the display, select data collections, apply cuts on quantities, set cut-out views of detector parts and visualize all sort of data about the event and the physics objects, as shown in figure 3a.

VP1 is used by all experts, to check and validate the physics analysis, the reconstruction, the simulation and the detector geometry at all stages. VP1, in fact, can natively access the experiment’s data at all steps of the ATLAS data chain, without the need of external converters, as shown in figure 4. Also, VP1 shows the actual detector geometry, directly rendered from the description stored in the ATLAS online geometry database, as shown in figure 3b.

On the other hand, being part of the experiment core software, VP1 must be run inside the experiment’s framework, in order to access all services and sub-systems; this dependency to the ATLAS framework prevents VP1 being a cross-platform tool. In section 4.2 a new tool developed to address this limitation will be presented.

##### 4.1. Architecture

VP1 is a C++-based framework which uses **Coin** [10] (an Open Inventor clone) as 3D engine and the **Qt** libraries [11] for the graphical user interface. It also makes use of the **SoQt** glue-package to use **Coin** within **Qt** widgets.

VP1 is a framework. The VP1 base packages implement the basic functionalities, like the main access to data, the user interface and the interface to the experiment’s framework; the **VP1Systems** packages define the access and the visualization of specific data collections; specialized views are then defined in the **VP1Plugins** packages. Thanks to the framework structure, users and groups can add specialized algorithms and tools to expand VP1 functionalities; to add, for instance, the visualization of a new piece of the detector or to effectively visualize the outcome of a specific data selection algorithm.

<sup>1</sup> The name **VP1** stands for *Virtual Point 1*, *Point 1* being the name of the CERN site hosting the ATLAS experiment.

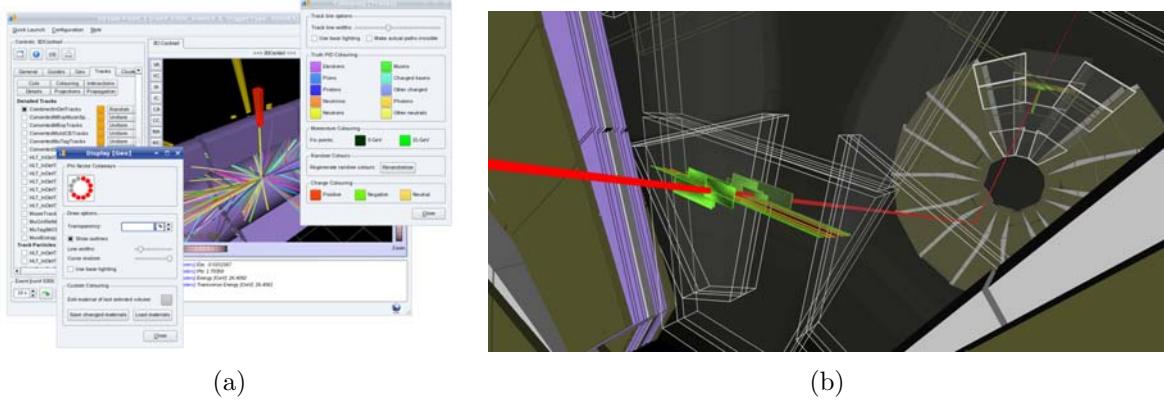


Figure 3: The graphic user interface of VP1, the 3D event display. (a) users interact with data setting different display options and applying requirements on kinematic quantities; (b) VP1 shows the real geometry of the ATLAS detector, as served by the **GeoModel** database running in the experiment’s framework; beside showing physics objects like tracks, jets, vertices, hits and calorimeter energy deposits, it is able to show all kind of ATLAS data at any step of the data chain (for instance, calibration and conditions data, trigger- and reconstruction-related quantities, truth information, simulation, and so forth).

#### 4.2. Current development

As said, being VP1 part of the experiment’s core software, it can run only on the platforms for which the experiment’s framework is built<sup>2</sup>. Therefore, until now, users have to run VP1 only on machines where the whole ATLAS framework can be installed.

But end users request to use VP1 on their own machines, specially those who run physics data analysis jobs on their laptops, analyzing custom-filtered data files for which they do not need access to all the experiment’s services and databases.

For that, the development of a light version of VP1 has started, in order to have a cross-platform stand-alone application which can be installed anywhere, and which let end users explore and visualize filtered experimental data stored in the so-called **xAOD** format [12] (which is based on the **ROOT** [13] data format). To achieve that purpose, different milestones have been set.

The first milestone involved the development of a new mechanism to persistify the detector description, which until now was only built on request and on-the-fly by the geometry database within the framework. For that, the package which is used to describe the detector geometry, the **GeoModel** package [4], has been taken out from the experiment’s framework and decoupled from it: **GeoModel** is now an experiment-agnostic geometry package, which can be used to describe any detector. Then, a new mechanism to store the detector description in a file and to restore it back has been designed and built. A detailed description of those steps can be found in [14]. The new mechanism is now used by the new VP1-Light to visualize the actual detector geometry offline.

The second milestone implied the decoupling from the framework of the base VP1 packages which provide the basic functionalities. All the modules which do not need to access specific ATLAS data or other experiment’s services are now decoupled and made experiment-independent.

With these milestone being achieved, a new cross-platform light version of VP1, VP1-Light, is now available to end users. Currently, it only visualizes the detector geometry, but an interface

<sup>2</sup> At the time of writing, “SLC6” and “CERN CentOS 7” only.

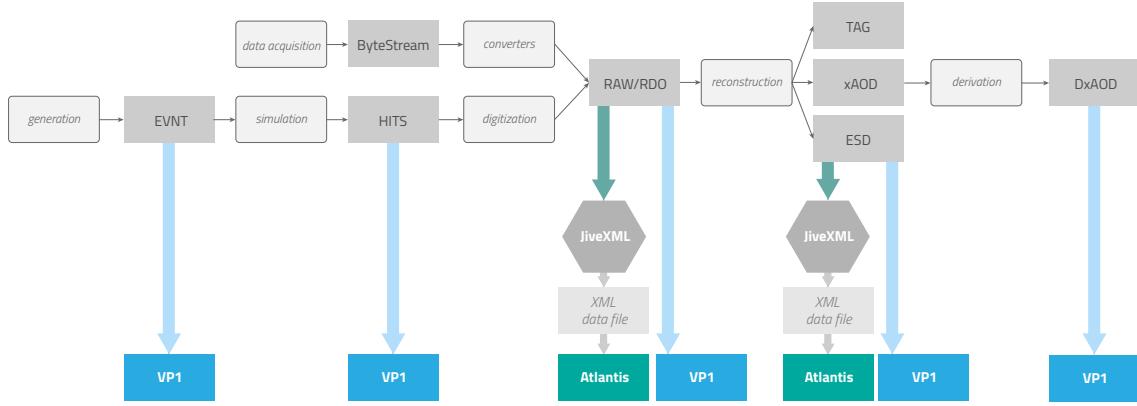


Figure 4: Atlantis and VP1 are able to access and visualize ATLAS data at different steps of the ATLAS Event Data Model (EDM). VP1, being part of the experiment’s framework, can access experiment’s data at all steps, natively; Atlantis, on the other end, uses a dedicated converter (JiveXML) to extract information from RAW and ESD data files.

to the xAOD file format and the visualization of the physics objects stored in it are under development and they will be released soon.

## 5. The ATLAS online event display

Atlantis and VP1 are also used in the online event display [15] running in the control room to check the data taking. The machinery runs a dedicated online reconstruction, to produce data files for both Atlantis (XML) and VP1 (ESD), on-the-fly, directly from the ATLAS data acquisition system (DAQ).

In the current setup, Atlantis 2D images are then produced in an automatic way as well, and served through a web interface and projected in the experiment’s control room; a sub-set of those images are also published for public access. The automatic production of 3D images with VP1 is currently under development. Moreover, optimization for high pileup events is foreseen in the near future.

## 6. Outreach & education

Atlantis and VP1 are used for outreach and education as well, for different projects and purposes.

Atlantis images are used in scientific papers (see figure 5a), and customized versions of Atlantis (MINERVA [16] and Hypatia [17]) are used as educational tools in masterclasses targeted at high-school students. Moreover, simplified Atlantis XML data files are used for an outreach Android-based mobile application, “LHSee” [18], and for the data-sonification project “LHCsound” [19].

VP1 is used to produce detailed 3D images of collision events (see figure 5b) used in scientific papers and to illustrate the experiment’s achievements in particle physics and in detector technology for press releases, multimedia, news, textbooks and technical reports for the funding agencies. VP1 images appear on online newspapers, as well. VP1 is also used to export the ATLAS geometry for other outreach projects, like the virtual reality ATLASriff educational game [20].

All ATLAS public event displays can be found at [21].

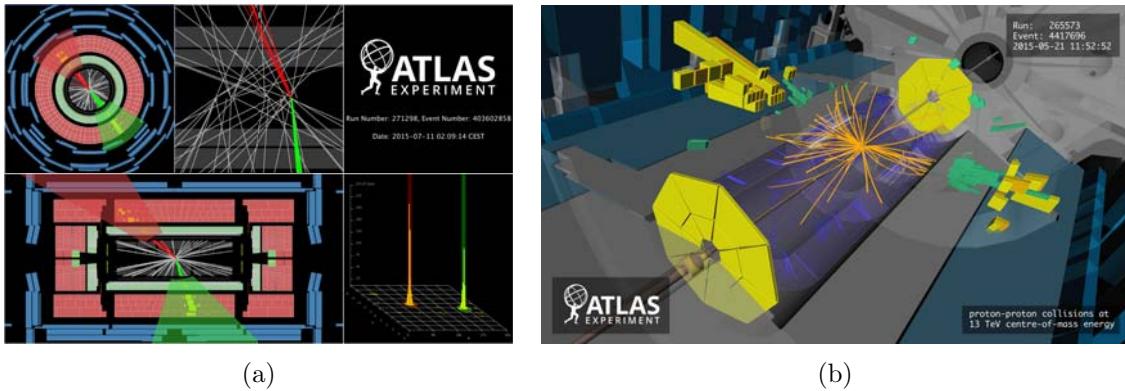


Figure 5: Atlantis and VP1 feature tools to prepare publication-ready event displays, to be used in scientific papers, conferences, press releases and reports. a) an event display made with Atlantis; b) an event display of a di-jet event at 13 TeV made with VP1, showing the actual experiment’s geometry, the ATLAS logo and the event details.

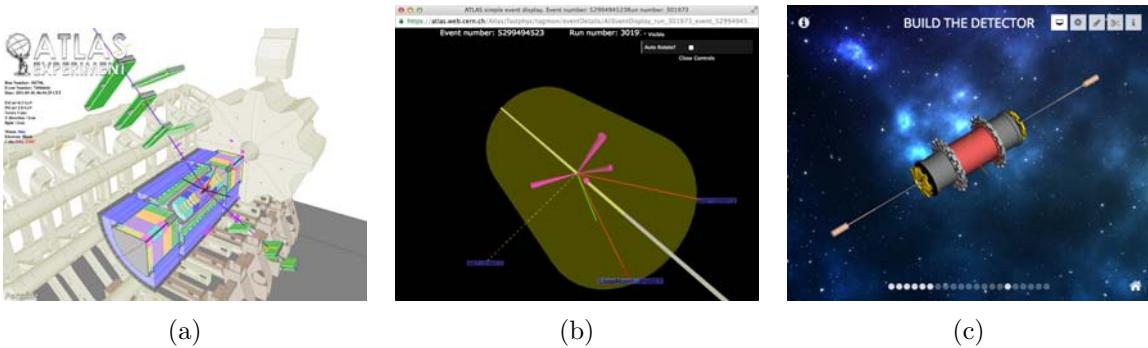


Figure 6: Other visualization tools in ATLAS. (a) Persint is a Fortran-based package aimed at visualizing the muon spectrometer and the reconstructed muon tracks; (b) TADA is a web-based viewer, aimed at visualizing the physics events selected by the ATLAS online prompt analysis; (c) Tracer is another web-based viewer, developed to be used in outreach and education projects.

## 7. Other visualization tools in ATLAS

Other, more specialized visualization tools have been developed within the ATLAS Experiment, with different technologies and with different purposes in mind. Here below a brief overview of them follows.

### 7.1. Persint

Persint [22] (see figure 6a) is a desktop stand-alone application which uses custom intermediate ASCII event files as input, which are mainly produced from raw-data. It has been developed with the primary focus on the visualization of the muon spectrometer geometry, the magnetic field and the reconstructed muon tracks. It uses a custom graphics library written in Fortran, with a C++ wrapper which links the core layer to the experiment’s framework and to the Qt-based graphical user interface. Persint is not actively developed anymore because of the lack of manpower, but it is maintained and it is still used in ATLAS as debugging tool.

### 7.2. TADA visualizer

The new TADA [23] display (see figure 6b) is a web-based viewer, which is under development, and which has been designed for the ATLAS online prompt analysis, the ATLAS Fast Physics Monitoring System. TADA input is a JSON data stream containing very simple information about an event, extracted from the so-called TAG file, an intermediate file containing only basic event information. It is a light-weight display built on top of the `three.js` [24] library (which, in turn, is an interface to WebGL), which offers 3D interactive view of physics objects (like jets and particles) and of a simplified detector geometry; it uses the `dat.gui` [25] library for the graphical user interface.

### 7.3. Tracer

ATLAS Tracer [26] (see figure 6c) is another new web-based viewer, under development, and designed to be used in outreach and educational programs, where it is used to let users learn about the detectors used in particle physics experiments and to explore simplified event data. It uses the WebGL-based `three.js` library and a simplified geometry based on the actual one stored in the ATLAS geometry database.

## References

- [1] Anderson C D 1933 The Positive Electron *Physical Review* **43** (6) pp. 491–494, DOI:10.1103/PhysRev.43.491.
- [2] Flegel I and Söding P 2004 Twenty-five years of gluons *CERN Courier* Nov <http://cerncourier.com/cws/article/cern/29201>
- [3] ATLAS Collaboration 2008 The ATLAS Experiment at the CERN Large Hadron Collider *JINST* **3** S08003.
- [4] Boudreau J and Tsulaia V 2004 The GeoModel Toolkit for Detector Description, *CHEP '04, Book of Abstracts*, <https://indico.cern.ch/event/0/contributions/1294152/>
- [5] Konstantinidis N *et al.* 2004 The Atlantis event visualisation program for the ATLAS experiment *Computing in High Energy Physics and Nuclear Physics 2004, Interlaken*, DOI 10.5170/CERN-2005-002.361
- [6] The Atlantis event display web site, <http://atlantis.web.cern.ch/>
- [7] Kittelmann T and Tsulaia V and Boudreau J and Moyse E 2010 The Virtual Point 1 event display for the ATLAS experiment *Journal of Physics: Conf. Ser.* **219** 032012, doi:10.1088/1742-6596/219/3/032012
- [8] ATLAS Collaboration, The VP1 ATLAS 3D event display, <http://atlas-vp1.web.cern.ch/>
- [9] Calafiura P *et al.* 2004 The ATHENA control framework in production, new developments and lessons learned, *Interlaken 2004, Computing in high energy physics and nuclear physics*, pp. 456–458
- [10] Coin, an OpenGL-based, 3D graphics library, <http://coin3d.bitbucket.org/Coin/index.html>
- [11] The Qt Company, Qt 5, <https://www.qt.io/>
- [12] Buckley A *et al.* 2015 Implementation of the ATLAS Run 2 event data model *J. Phys.: Conf. Ser.* **664** 072045
- [13] Brun R, Rademakers F 1997 ROOT — An Object Oriented Data Analysis Framework *Nucl. Inst. & Meth. in Phys. Res. A* **389** pp. 81–86. <http://root.cern.ch/>.
- [14] Bianchi R M and Boudreau J and Vukotic I 2016 A new experiment-agnostic mechanism to persistify and serve the detector geometry of ATLAS *Proceedings of the CHEP 2016 conference J. Phys.: Conf. Ser.*
- [15] ATLAS Collaboration, ATLAS Live event display, <http://atlas-live.cern.ch>
- [16] MINERVA educational event display, <http://atlas-minerva.web.cern.ch>
- [17] Hypatia educational event display, <http://hypatia.phys.uoa.gr>
- [18] LHSee and Collider, <https://www2.physics.ox.ac.uk/about-us/outreach/public/lhsee>
- [19] LHCsound, the sound of science, <http://lhcsound.hep.ucl.ac.uk/>
- [20] Vukotic I and Moyse E and Bianchi R M 2015 ATLASrift — a Virtual Reality application arXiv:1511.00047 [physics.ins-det]
- [21] ATLAS Collaboration, public ATLAS event displays directory, <https://twiki.cern.ch/twiki/bin/view/AtlasPublic/EventDisplayPublicResults>
- [22] Pomarède D and Virchaux M 2003 The Persint visualization program for the ATLAS experiment *Conference for Computing in High-Energy and Nuclear Physics, La Jolla, CA, USA*,
- [23] Eifert T and Elsing M and Gumpert C and Kamioka S and Moyse E and Nairz A and Sabato G 2016 ATLAS Fast Physics Monitoring: TADA *Proceedings of the CHEP 2016 conference J. Phys.: Conf. Ser.*
- [24] Three.js, a WebGL library, <https://threejs.org/>
- [25] dat.GUI, <https://workshop.chromeexperiments.com/examples/gui/#1--Basic-Usage>
- [26] ATLAS Tracer, <https://atlas-tracer.web.cern.ch/>