WIRED - World Wide Web Interactive Remote Event Display[1]

A. Ballaminut[1], *C. Colonello*[1], *M. Dönszelmann*[2], *E. van Herwijnen*[2], *D. Köper*[3], *J. Korhonen*[4], *M. Litmaath*[5], *J. Perl*[6], *A. Theodorou*[2], *D. Whiteson*[2], *E. Wolff*[2]

[1] Dipartimento di Fisica, Università degli Studi di Udine, Via delle Scienze 208, 33100 Udine, Italy
[2] CERN, CH-1211 Geneva 23, Switzerland
[3] NIKHEF, P.O.Box 41882, 1009 DB Amsterdam, The Netherlands
[4] University of Oulu, Linnanmaa BOX 4500, FIN-90401 Oulu, Finland
[5] Fermilab, P.O.Box 500, Batavia IL 60510-0500, U.S.A.
[6] SLAC, P.O.Box 4349, Stanford CA 94309, California, U.S.A.

Abstract

WIRED[2] is a framework, written in Java, to build High Energy Physics event displays that can be used across the network. To guarantee portability across all platforms, WIRED is implemented in the Java language and uses the Swing user interface component set. It can be used as a stand-alone application or as an applet inside a WWW browser.

The graphical user interface allows for multiple views and for multiple controls acting on those views. A detector tree control is available to toggle the visibility of parts of the events and detector geometry. XML (Extensible Markup Language), RMI (Remote Method Invocation) and CORBA loaders can be used to load event data as well as geometry data, and to connect to FORTRAN, C, C++ and Java reconstruction programs. Non-linear and non-Cartesian projections (e.g. fish-eye, rho-phi, rho-Z, phi-Z) provide special views to get a better understanding of events.

WIRED has grown to be a framework in use and under development in several HEP experiments (ATLAS, CHORUS, DELPHI, LHCb, BaBar, D0 and ZEUS). WIRED event displays have also proven to be useful to explain High Energy Physics to the general public. Both CERN, in its travelling exhibition and MicroCosm, and RAL, during its open days, have displays set up.

Keywords: HEP Event Displays; Java; RMI; XML; CORBA

1 Introduction

The initial version of WIRED[1] was written to study the, at that time, still immature Java technology on the DELPHI experiment by creating a fairly visible application, an event display. WIRED was an applet that could be downloaded in a WWW browser, and physicists all over the world could see and study events. Over time Java matured, became quite a lot faster and more portable. The early version of WIRED was rewritten completely to accomodate multiple experiments and to put emphasis on the visualization of High Energy Physics events.

Today WIRED[2] is a framework[3] to build High Energy Physics event displays. It can run in stand-alone mode or in a web browser and in either mode it can have remote access to data via a CORBA or RMI server. Experiments normally extend the framework via plug-in modules to give WIRED some experiment-specific behaviour. WIRED comes with a powerful graphics engine, which is specially made to handle the display of events in different and sometimes special projections.

---

[1] A longer version of this paper is also available.
[2] URL: **http://wired.cern.ch/**

2 Architecture

The architecture of WIRED is based around a client-server model. WIRED can be run in three different basic configurations, as shown in Figure 1. In the first configuration the client is run in stand-alone mode, pre-installed on a machine, with local access to geometry and event data on that machine in the form of files. The second configuration is different from the first in that the client now receives its geometry and/or event data from a remote server. Local access to data is still possible. In the last configuration WIRED is run as an applet inside a browser. The actual WIRED code is downloaded from a WWW server and for geometry and event data the same server as in the second configuration is used. Due to security restrictions no local data access is possible. Other configurations, which involve gateways to stream the data via specified servers, are also possible.

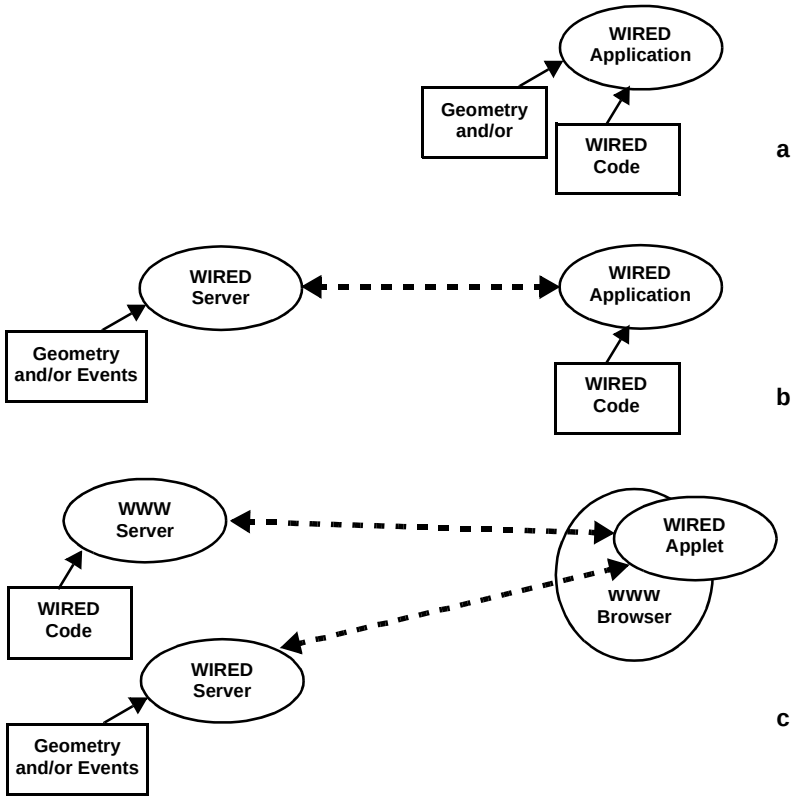

Figure 1: WIRED running in three different configurations: **a** - stand-alone, with local code, geometry and event data, **b** - in client-server mode, with local code, but remote geometry and event data and **c** - in browser mode, where code, geometry and event data are all downloaded into an applet inside a web browser.

The authors chose to implement WIRED in the Java language[4] using the Swing user interface kit[5], both of which allow for easy portability and running as an applet. The server may be written in another language, such as C or C++, to allow for easy coupling with an event reconstruction framework.

The WIRED client Graphical User Interface contains a View Top and a Control Top, as shown in Figure 2. The first allows the user to create several different views on his event. Each view can be directly manipulated with the mouse and context sensitive popup menus allow the user to do most of the interaction. The Control Top shows the different controls, which may act on one or more selected views. These controls can show the status of different views and allow for more complicated interaction. An optional console will be added to allow the user to interact via a

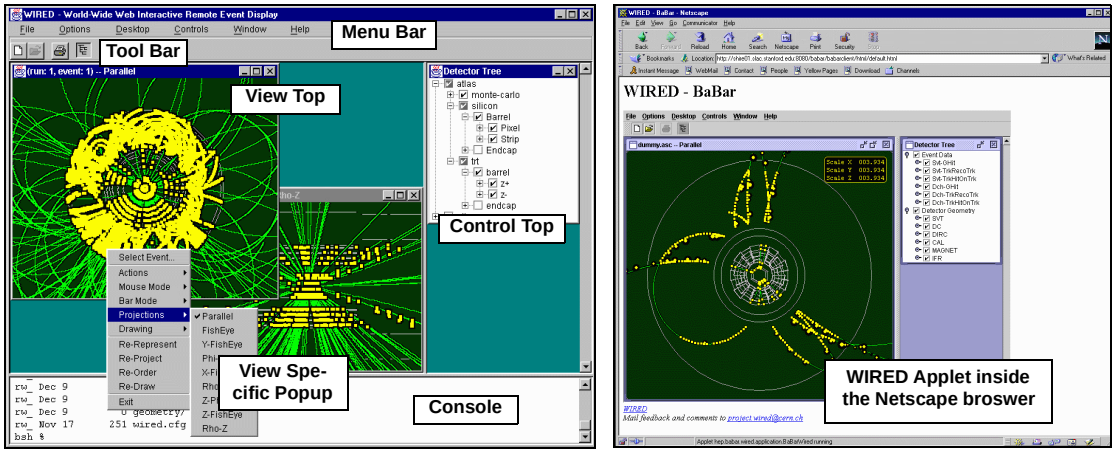

Figure 2: The WIRED Graphical User Interface: on the left the stand-alone version of WIRED showing the ATLAS detector in two views on the view top. The control top shows the component structure of the event. Toolbar and menubar allow for access to general functionality, while popup menus allow for view specific functionality. On the right the browser version of WIRED, showing the BaBar detector, with identical functionality.

scripting language. The client as a whole, the Menubar, Toolbar, View Top, Control Top and Console, is an applet which can be used in browser mode.

3 Data Access

Event and geometry data is read into WIRED using a loader. This loader may load data from a local or remote file system, by using for instance XML, or it may use a communication protocol, such as CORBA or RMI, to access a remote server and load objects directly into memory.

XML is a standard to define human and machine readable formats[6]. WIRED uses XML to define generic and specific formats for writing out geometry and event files from reconstruction or analysis programs. Though no true interaction between WIRED and the supplying program is possible, XML provides a fairly easy mechanism to have access to data. An off-the-shelf XML parser and other XML tools are used in WIRED.

RMI (Remote Method Invocation) or CORBA[7] can be used to provide a direct connection between WIRED and an event reconstruction program. RMI only allows Java to Java connections, however the JNI (Java Native Interface)[8] can be used to interface to other languages. Or CORBA provides an alternative to connect to remote servers written in Fortran, C or C++. Both RMI and CORBA allow the full event object structure to be transported into WIRED.

Different experiments opt for different solutions. WIRED uses a class plug-in structure, see Figure 3, to specify the option used. Two experiment-specific loaders need to be written to load both geometry and event data. Both loaders, which either use XML, CORBA or RMI, will map geometry and event data onto so called representables. These representables are for instance pieces of the detector, tracks, tracker hits or calorimeter hits, and each contain enough information to be displayed. The graphics engine takes the representables, converts them, and shows them on the screen. The integration of a generic interface for both representables and their representation is ongoing[9]. .

4 Graphics Engine

The WIRED graphics engine[10] uses a traditional graphics pipe[11] to convert data before displaying it (see Figure 3). Representables (tracks, hits, calorimeter hits, ...) are represented by

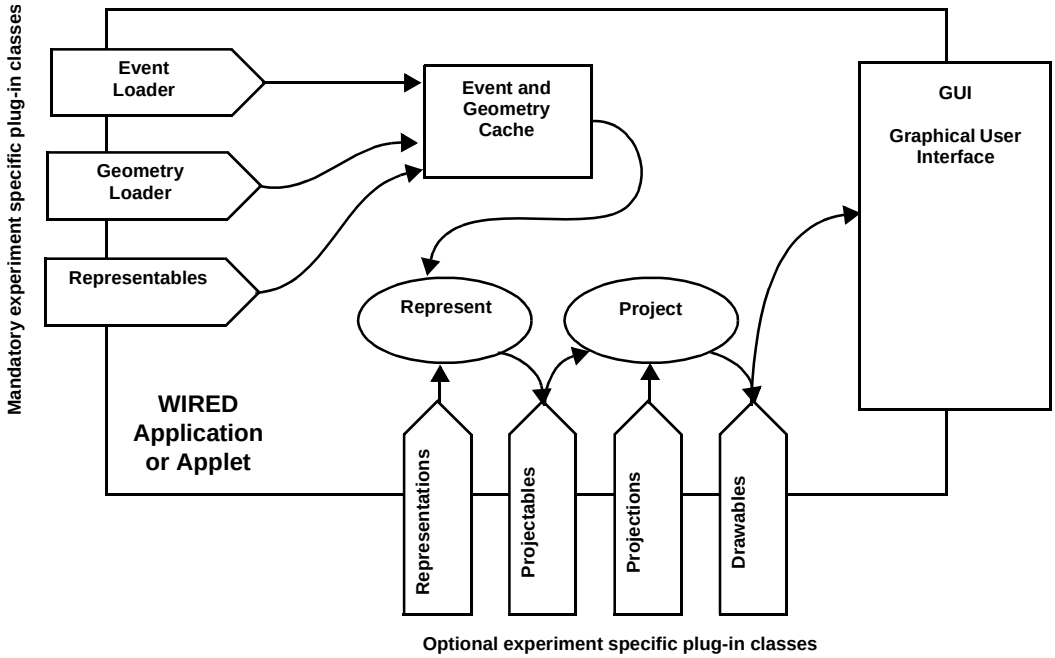

Figure 3: The plug-in architecture of WIRED. Several mandatory experiment specific plug-in classes are needed to tell WIRED what to show on the screen. Optional plug-in classes may add extra functionality and behaviour to WIRED.

projectables (squares, lines, ...) by using representations (hit as square, track as line, ...). These projectables are 2D symbols with 3D coordinates. These coordinates are then projected into drawables by using one or more cascaded projections. Drawables are 2D symbols with 2D coordinates and can therefore be drawn on the screen. The use of 2D symbols for hits becomes clear, see Figure 4, when the hits of an event should always be recognizable even if one zoomed out very far. A traditional 3D graphics engine will scale down the hits accordingly, while the WIRED graphics engine keeps its symbols - the hits - at the same size. WIRED provides a set of standard representables, projectables and drawables and some standard representations and projections, to convert one into the other.

Projections in WIRED are specified in terms of functions and can therefore be linear as well as non-linear. Parallel, flat, scalable, rho-fish-eye, XYZ-fish-eye, rho-phi, rho-Z and phi-Z are a few of the many possible standard projections, and the user can easily extend this set. Projections can be cascaded: for instance, to obtain a scalable rho-fish eye one would apply a scalable projection followed by an rho-fish-eye projection. Special projections can be very useful to enhance a particular aspect the user is studying[12]. The rho-fish eye projection, for instance, blows up the inner parts and compresses the outer parts of the detector, thereby not sacrificing the visibility of the latter, as would be the case for an ordinary scalable projection.

The drawables, which appear on the screen, are layered to allow the event to be drawn on top of less important geometry information. Layering also handles the drawing of thin frames around the drawables and this enhances the clarity of the picture substantially.

5 Conclusions

WIRED provides a framework in Java to write High Energy Physics event displays. It is highly portable, runs as a stand-alone application or as an applet inside a WWW browser. It uses a client-server architecture to allow the user to access geometry and or event data locally or remotely. The Graphical User Interface with its View Top and Control Top allow the user to easily

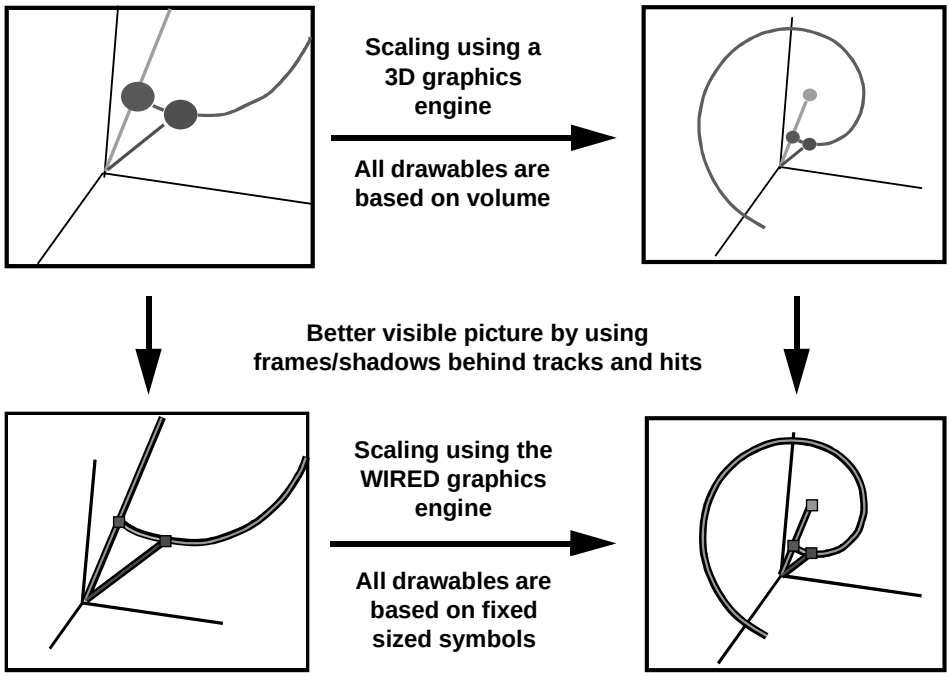

Figure 4: Conventional 3D graphics engines scale all volumes they draw (top graphs), while the WIRED graphics engine scales the 3D coordinate, but not the symbols, such as the hits and the tracks (bottom graphs). Frames/shadows around the symbols enhance the visibility of the picture.

view events in different ways and to interact with them. Three methods of accessing data, via XML, CORBA or RMI, provide every experiment with a way to write their plug-in modules. WIRED's powerful graphics engine is unique in the sense that it handles non-linear and non-cartesian projections. The layering model used by the graphics engine produces nicely enhanced pictures.

References

1 M.C. Coperchio, M.Dönszelmann, P. Gunnarsson, "WIRED - World-Wide Web Interactive Remote Event Display", 1997 CERN-97-01.

2 M.C. Coperchio et al., "WIRED - World-Wide Web Interactive Remote Event Display", Computer Physics Communications 110 (1998) 155-159.

3 Don Roberts and Ralph Johnson, "Evolving Frameworks: A Pattern-Language for Developing Object Oriented Frameworks", Pattern Languages of Program Design 3, Addison-Wesley, 1998.

4 James Gosling, Bill Joy and Guy Steele, "The Java™ Language Specification", Addison-Wesley, 1996.

5 Robert Eckstein, Marc Loy & Dave Wood, "Java Swing", O'Reilly, 1998.

6 Tim Bray et al., "Extensible Markup Language (XML) 1.0", W3C, 1998.

7 Elliotte Rusty Harold, "Java Network Programming", O'Reilly, 1997.

8 Rob Gordon, "Essential JNI: Java Native Interface", Prentice-Hall, 1998.

9 Joseph Perl, "HepRep: a Generic Interface Definition for HEP Event Display Representables", SLAC-PUB-8332, Stanford, California, January 2000.

10 A. Ballaminut, "A Graphics Engine for High Energy Physics Event Displays", Thesis, University of Udine, Italy, 1997/1998.

11 J.D. Fowley et al., "Computer Graphics: principles and practice", 2nd Edition, Addison-Wesley, 1996.

12 H. Drevermann, D. Kuhn, B.S. Nilsson, "Event Display: Can we see what we want to see?", 1995 CERN-ECP/95-25.