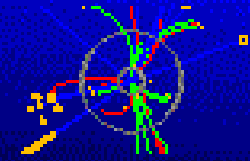


# WIRED



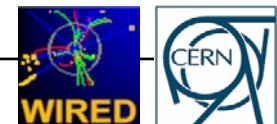
World-Wide Web Interactive Remote Event Display

## WIRED Detector Description in XML

Mark Dönszelmann, Applications for Physics and Infrastructure, IT, CERN

XML Detector Description Workshop

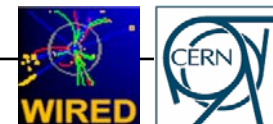
CERN, 14 April, 2000



# Contents

---

- XML Usage in WIRED
- Detector Description for WIRED
- GEANT3 to XML Conversion
- Example: WIRED Detector Description
- SAX and DOM Parsers
- Performance
- Solutions (now)



# XML Usage in WIRED

---

## ■ Now:

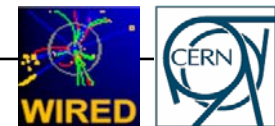
- Detector Description
- Event Description

## ■ Planned:

- Menu and Toolbar Configuration
- User Configuration Persistency
- HEPRep Description (generic event description)

## ■ JAS (Java Analysis Studio)

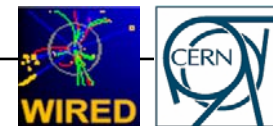
- Menu and Toolbar Configuration
- Particle Data Table (PDG)



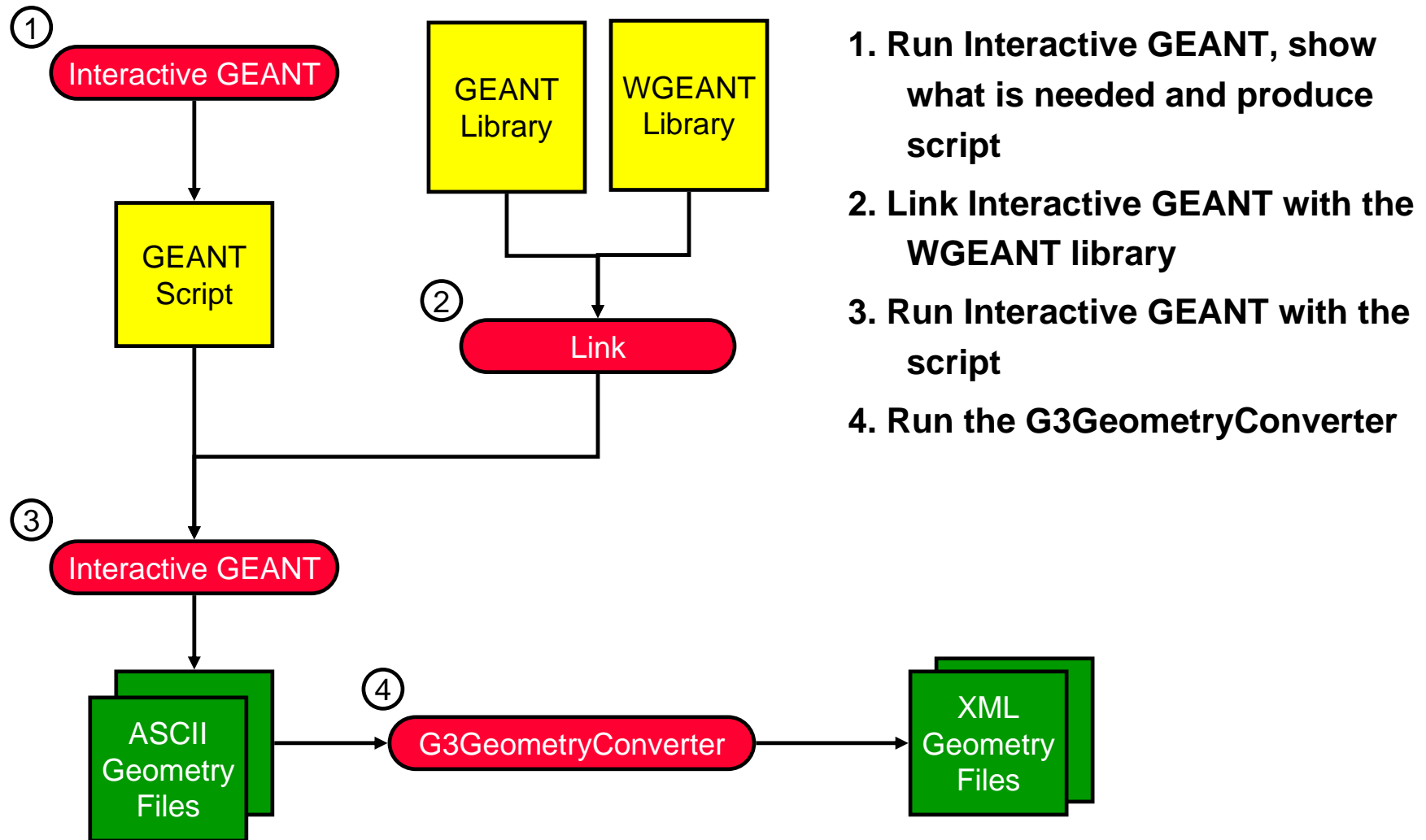
# Detector Description for WIRED

---

- **Defined in November 1998.**
- **3D Wire Frames specific for Event Display**
- **Primitives**
  - Polyline
  - Polygon
  - Arc
  - Circle
  - Text
- **Coordinates expressed as  $x,y,z$  or  $\rho,\phi,z$**
- **Part tag to group several primitives and groups**
- **Part tag can refer to different file (deferred loading)**



# GEANT3 to XML conversion



# Example: WIRED Detector Description

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE geometry SYSTEM "chorusgeometry.dtd" >
<geometry type="detector" name="Detector Geometry"
  description="neutrino-physics experiment">
  <head>
    <experiment>CHORUS</experiment>
    <author>M.Litmaath</author>
    <version>1.0</version>
    <date>26-Apr-98</date>
  </head>
  <part type="detector" name="chorus">
    <part type="subdetector" name="TARG"
      description="Emulsion Target"
      href="TARG.xml.gz" load="true" />
    <part type="subdetector" name="HEXA"
      description="Hexagonal Magnet"
      href="HEXA.xml.gz" load="true" />
    <part type="subdetector" name="CALO"
      description="Calorimeter"
      href="CALO.xml.gz" load="true" />
    <part type="subdetector" name="SPEC"
      description="Spectrometer"
      href="SPEC.xml.gz" load="true" />
  </part>
</geometry>
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE geometry SYSTEM "chorusgeometry.dtd" >
<geometry type="detector" name="Chorus">
  <head>
    <experiment>Chorus</experiment>
    <author>Geant 3 Geometry Converter to XML</author>
    <version>1.0</version>
    <date>Jul 8, 1999</date>
  </head>
  <part type="subdetector" name="TARG" description="">
    <polygon N="4">
      <coord y="100.0000000" z="100.0000000" x="-13.81750011"/>
      <coord y="-100.0000000" z="100.0000000" x="-13.81750011"/>
      <coord y="-100.0000000" z="-100.0000000" x="-13.81750011"/>
      <coord y="100.0000000" z="-100.0000000" x="-13.81750011"/>
    </polygon>
    <polygon N="4">
      <coord y="100.0000000" z="100.0000000" x="-5.841500282"/>
      <coord y="-100.0000000" z="100.0000000" x="-5.841500282"/>
      <coord y="-100.0000000" z="-100.0000000" x="-5.841500282"/>
      <coord y="100.0000000" z="-100.0000000" x="-5.841500282"/>
    </polygon>
    <polyline N="2">
      <coord y="100.0000000" z="100.0000000" x="-13.81750011"/>
      <coord y="100.0000000" z="100.0000000" x="-5.841500282"/>
    </polyline>
    ...
  </part>
</geometry>
```

# SAX and DOM parsers

---

## ■ SAX - Simple API for XML

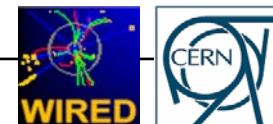
- Developer implements the SAX interface

```
public void startElement(String tag, AttributeList list) throws SAXException;  
  
public void endElement(String tag) throws SAXException;  
  
public void charData( char ch[],  
                    int start,  
                    int length) throws SAXException;
```

- Developer calls parse(String filename) on the XMLParser

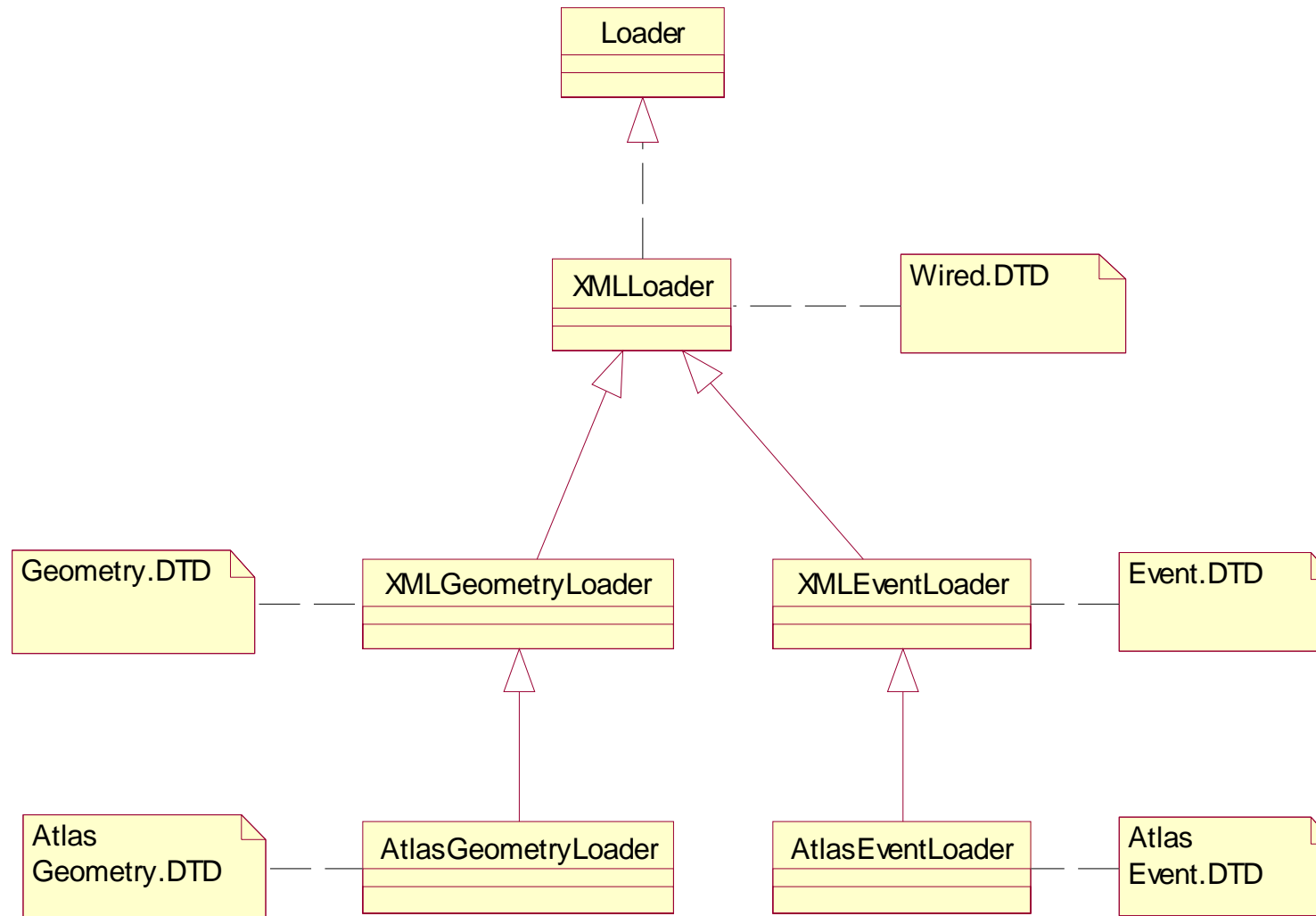
## ■ DOM - Document Object Model

- Parser builds an internal DOM tree in memory from the XML file
- Developer may create or modify a DOM tree
- Developer may write out a DOM tree into an XML file



# WIRED Classes and DTD's

---





# Performance

---

## ■ File Size:

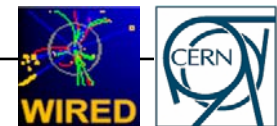
- DELPHI 2 levels of detail: 5 Mbyte, compressed 500 kbyte
- BaBar: 1.5 Mbyte, compressed 150 kbyte
- Compression of average a factor 10

## ■ Memory Usage:

- SAX parsing only creates the WIRED Objects
- DOM parsing creates objects for every attribute.  
From these WIRED Objects are created.  
Too memory intensive.

## ■ Parsing Speed:

- Reading compressed files or normal files does not differ if access is fast
- 10% read, 30% parse, 40% validation, 20% conversion of doubles
- XML versus Fixed format costs factor 2-3 in speed!



# Solutions (now)

---

- **Split detector description into multiple files**
  - use the PART tag to link to different files
  - files can be loaded on demand
- **No easy way to split DTD**
  - trick used with entity to include generic Wired.DTD
  - path to search for this file not very well defined in XML
- **DOM parser creates too many Objects for Data use**
- **Use SAX parser and read without validations**
  - WIRED uses the Aelfred (24 kbyte bytecode) parser
  - Aelfred is a non-validating pre-SAX parser
  
- **For the future we should look at XML Schema**

