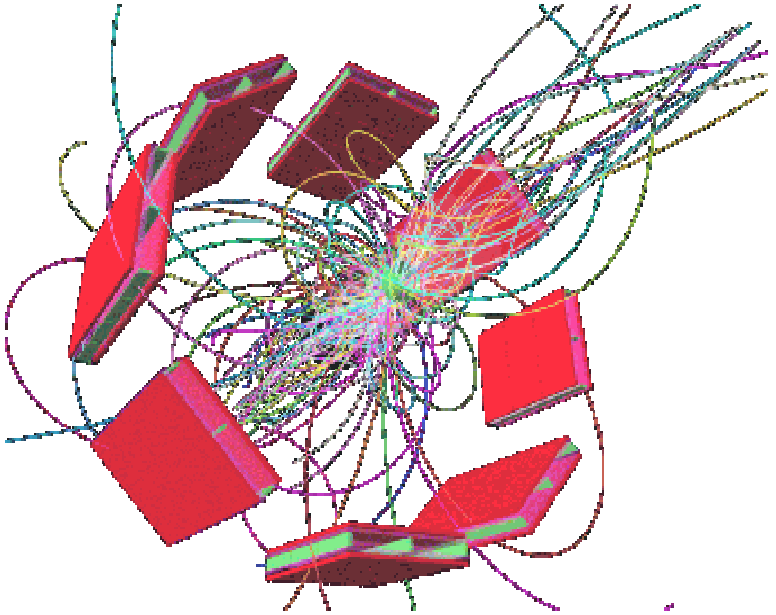
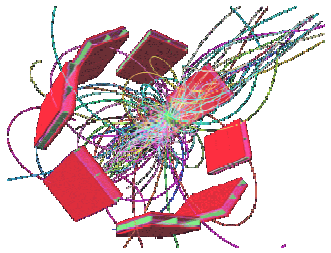
GraXML

Modular Geometric Modeler

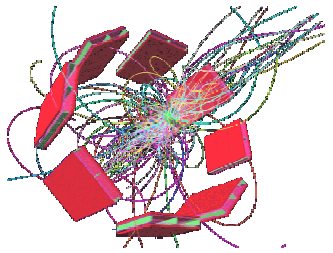
- History
- Architecture
  - Generic Model
  - Geometric Model
  - Core
- Data Sources
- Applications





History (1)

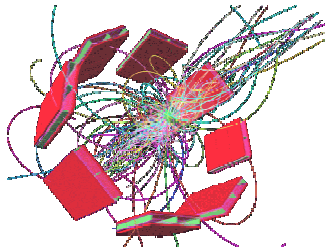
- Originally just 3D Detector visualisation for Atlas Generic Detector Description (AGDD).
- Several other geometry descriptions added later:
  - AliDD (extended AGDD)
  - AGDD v6++ (with math formulas, access to RDBMS,...)
  - AtlasEvent
  - AtlantisEvent (as prototype)
  - GDML (as prototype)
  - Direct access via API (Java directly, C++ via JACE proxies)



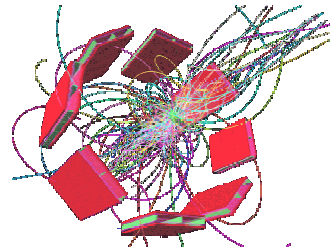
History (2)

- Requirements on Event/Detector Display functionality often correspond to requirements on Geometric Modeler:
  - Logical navigation in complex geometric structures,
  - Geometrical navigation (where am I ?, intersections, picking, transparency, calibration and interactivity,...),
  - Optimised geometry (millions of 3D objects).
- The only additional Display functionality is the manipulation of visual properties (colors,...) - quite memory hungry.
  - 3D Geometry Engine can be used as a Geometric Modeler foundation if visual overhead can be removed.

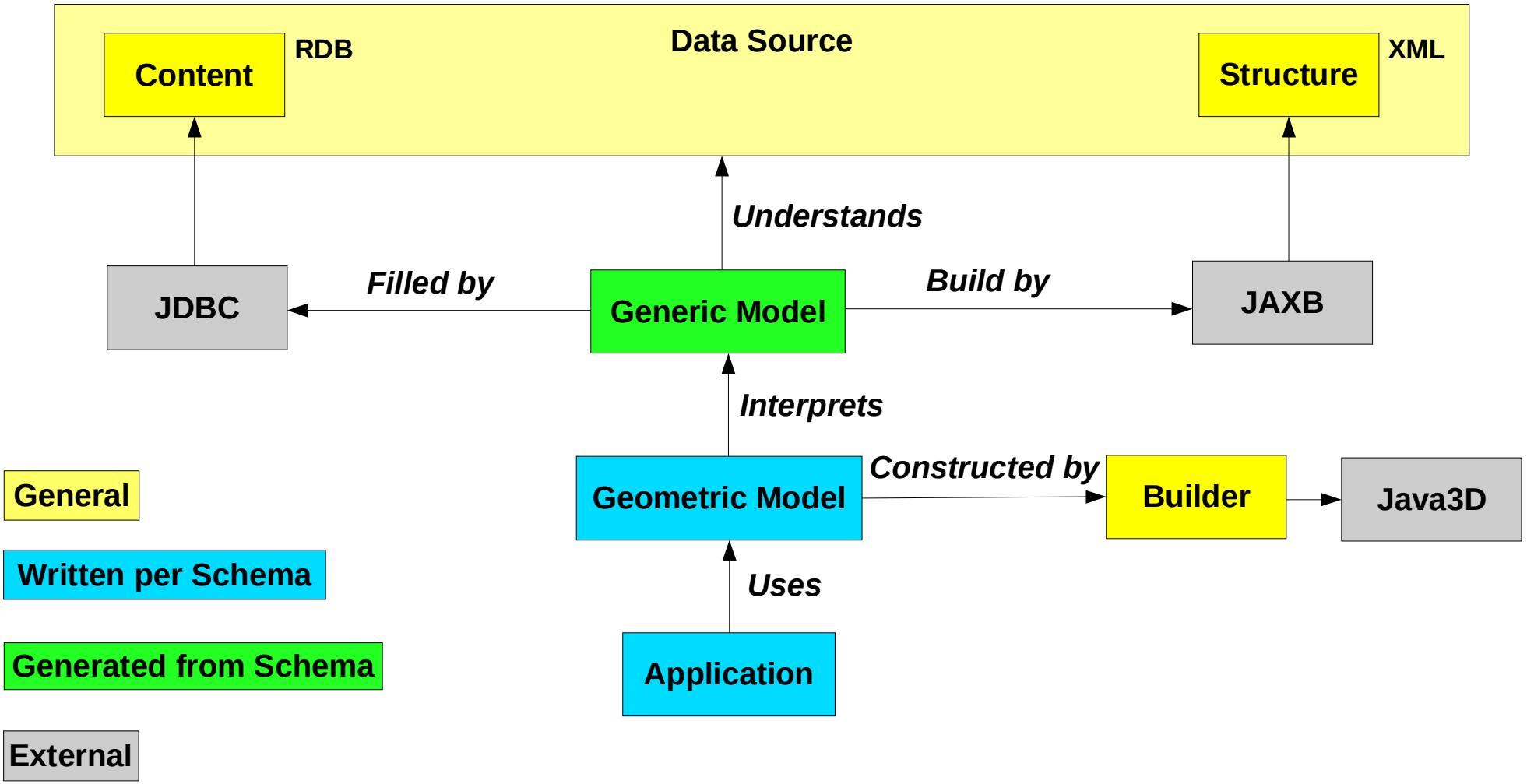
History (3)



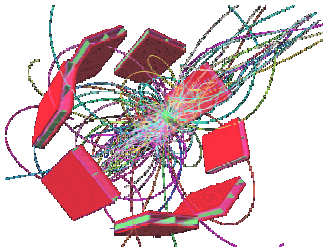
- Java3D, used as Geometry Engine in GraXML, can be used without visual overhead.
- GraXML has been re-engineered to provide flexible Geometric Modeling Toolkit with optional Display capabilities.
- GraXML components:
  - Core
  - Generic Modelers (logical graphs)
  - Geometric Modelers (3D graphs)
  - Applications (visual or not)
  - Utilities

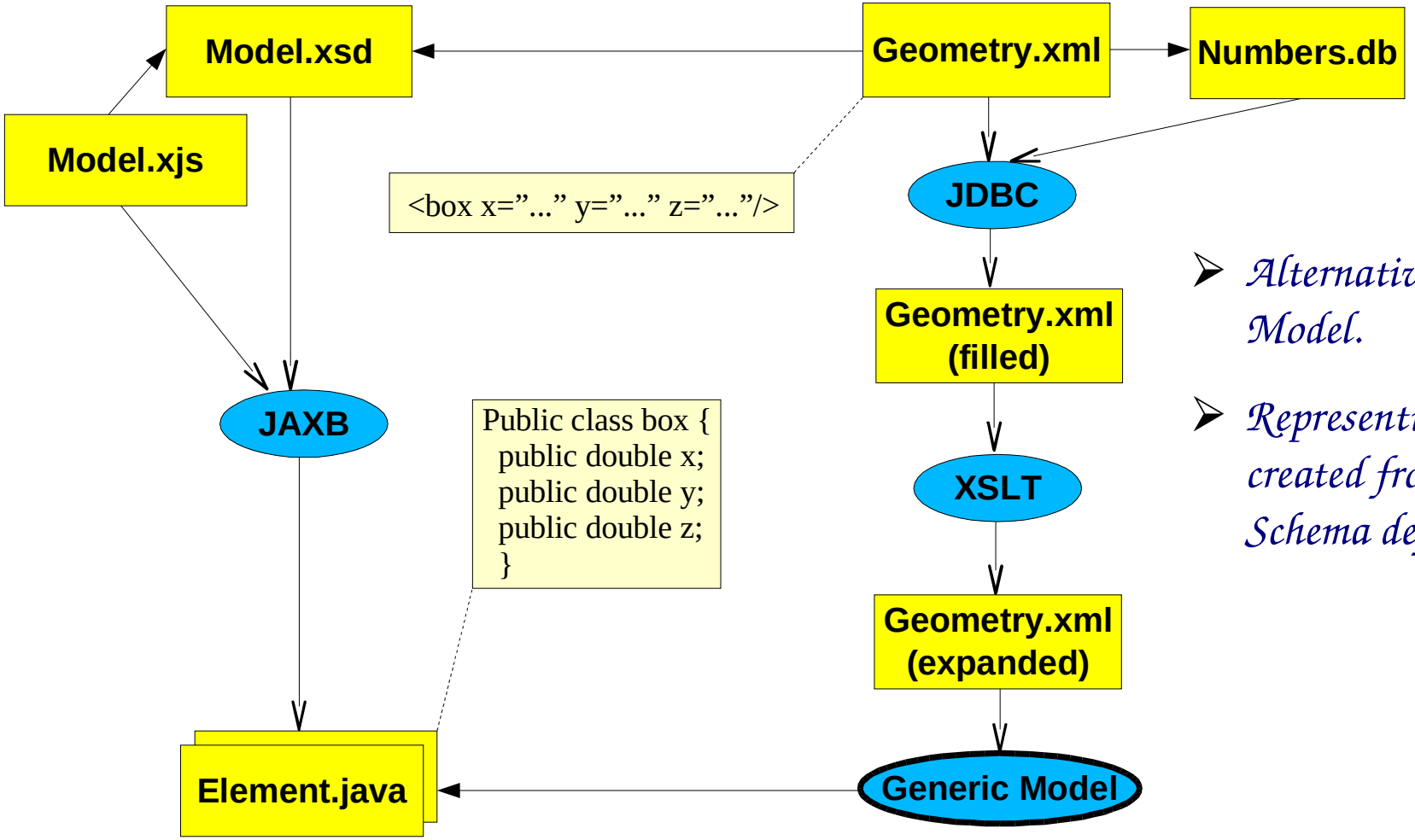


Architecture

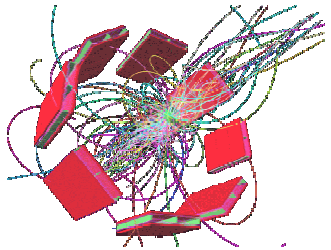


Generic Model





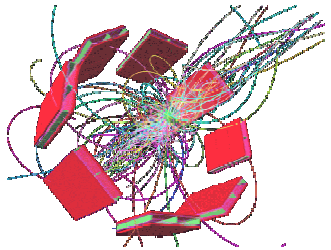
- *Alternative XML Object Model.*
- *Representing Classes created from XML Schema definition.*



JAXB Generic Model (2)

- Representing Classes created from XML Schema definition:
  - Proper set/getters created.
  - Variables have proper types and default values based on Schema definition and optional customisation file.
  - Generation can be customised (via XML customisation description) to capture more complex structures and relations.
  - Generated classes can be modified by extension or by helper conversion classes.
- More natural and faster than DOM/SAX.

Initial values



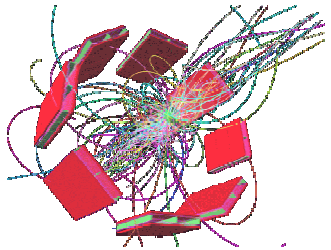
- XML file can have initial values outsourced into RDBS (for example NOVA DetDescrDB).
- XSQL Schema is used to define connections between XML file and RDBS.
- Simple JDBC Connection is used to fill those values into parametrised XML file.

```
<XSQLConfig>
  <connectiondefs>
    <connection name=”demo”>
      ...
      <dburl>jdbc:mysql://atlassw1.phy.bnl.gov/NOVA_dev</dburl>
      </connection>
    </connectiondefs>
  </XSQLConfig>

...

<var connection=”demo” name=”SCT.length”/>
...
```

Formulas

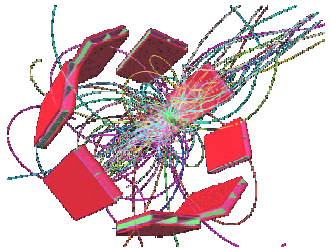


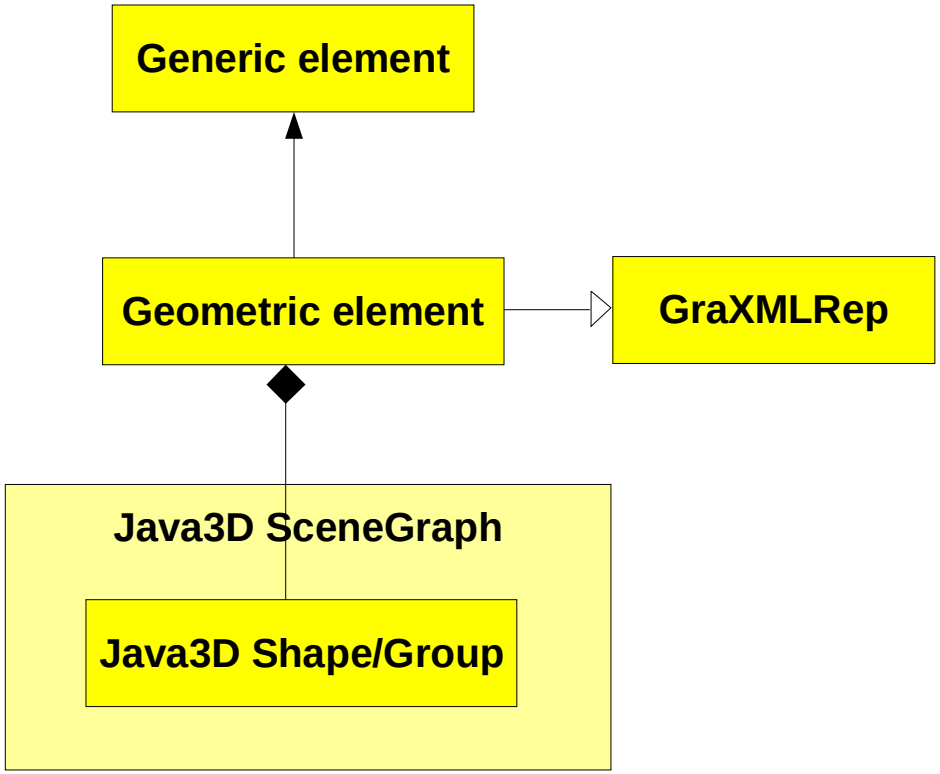
- *Symmetries and dependencies between attributes in XML file can be expressed using standard mathematical formulas.*
- *XSLT stylesheets with simple Java BeanShell evaluator is then used to expand structures with formulas to concrete elements.*

Design by Ch.Arnault

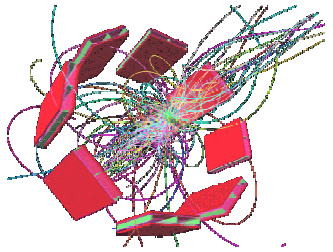
```
<array name="a" values="1;2;3;4;5;6;7;8;9;10"/>
<table name="t">
    <row values="1;2;3;4;5"/>
    <row values="6;7;8;9;10"/>
  </table>
<var name="a0" value="1"/>
<var name="b" value="a0*2"/>
<var name="c" value="a[2]*a[3]"/>
...
<box X_Y_Z=”5.5; a[5]; t[2,3]” name=”abox”/>
...
```

Geometric Model (1)



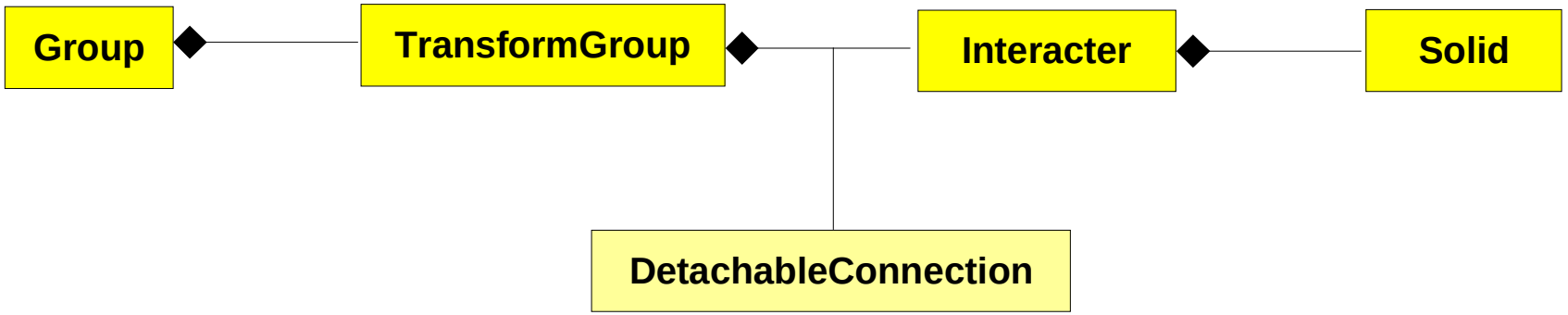


- Java3D SceneGraph build from GenericModel according to BuildOptions:
  - Graphical or not: Whether to insert visual attributes.
  - Level of Optimisation: How far to share representations (SharedGroup) for identical structures.
  - Level of Quality (restricted by Optimisation): How closely to make approximations and visual extensions.
  - Level of Interactivity (defined by Optimisation and Quality): How interactive Display should be, how far can be objects calibrated.
  - Representations (ant their properties) to be used to represent Generic Objects.

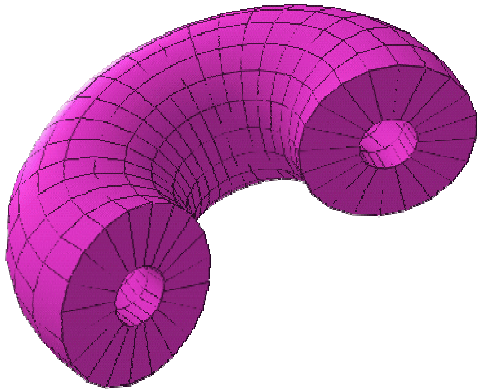


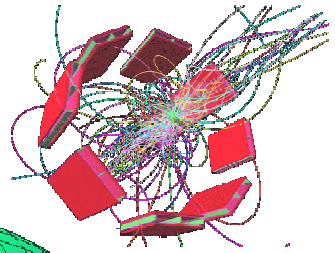
Geometric Model (2)

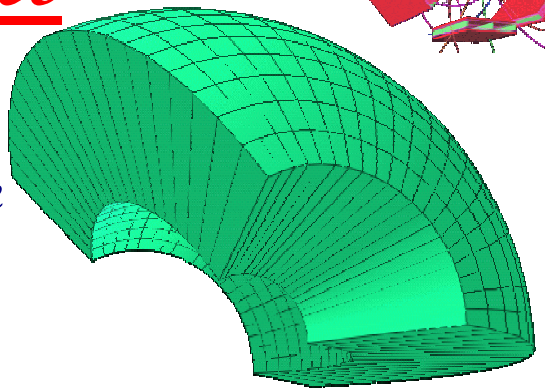
- Functionalities are generally added as special nodes inserted into a SceneGraph.
- As a SceneGraph (or its subgraphs) are then compiled for speed, BuilderOptions can't be changed later (when SceneGraph is active and used).



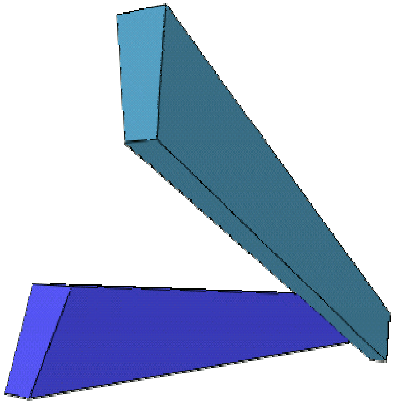
HEP Java3D Shapes

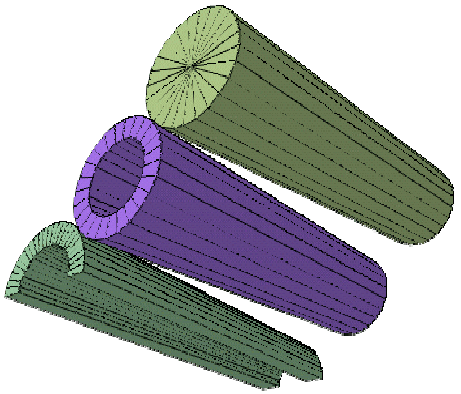




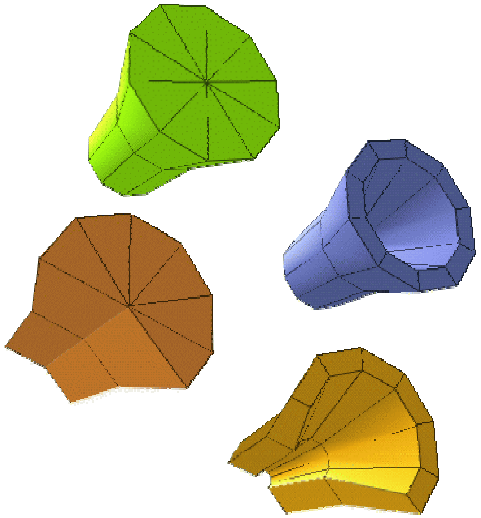


- All Geant4 CSG Solids (and some others) implemented as standard Java3D Shapes (with equivalent constructors). Usually specialisations of more generic Shapes.

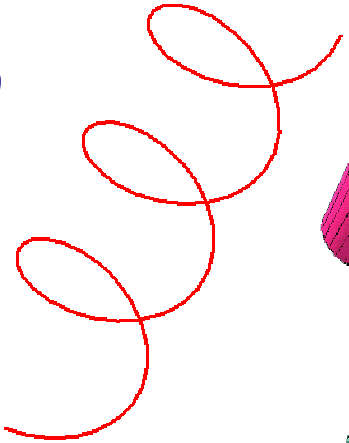


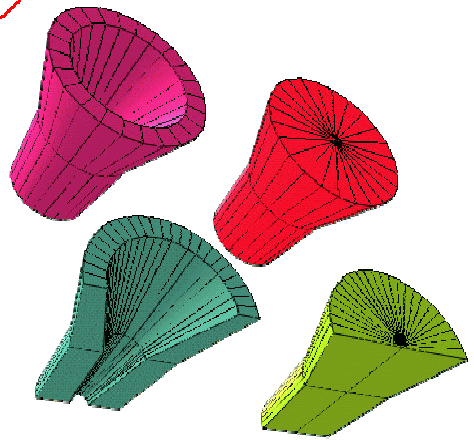


- New Shapes (Helix,...) are added as needed.

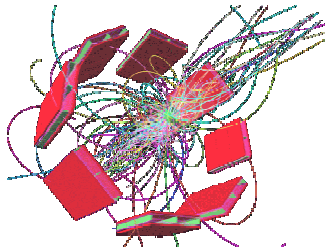


- Special Shapes (Outline,...) included too.



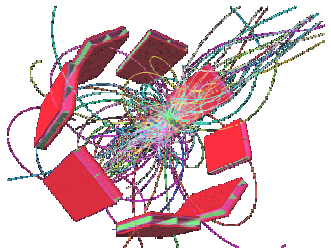


- Contribution to FreeHEP.



SceneGraph Optimisation

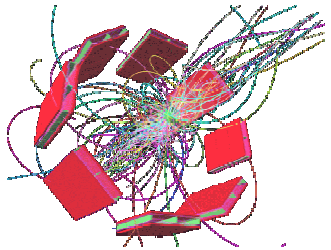
- Repeated structures are discovered during SceneGraph building and reused as SharedGroups.
- Level of optimisation depends on chosen BuildOptions.
- More optimised SceneGraph is smaller and faster, but does allow only limited interactivity (calibration,...) as SceneGraph Groups are often shared and can't be individually changed.



Interactivity / Calibration

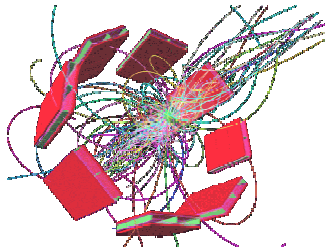
- Depending on level of Optimisation, SceneGraph elements (Shapes, Groups) can be modified at run-time:
  - Calibration
  - Graphical Operations:
    - Modification of Real Object (shape, place, orientation,...)
    - Modification of Visual Characteristics (visibility, color, transparency,...)

Data Sources



- XML:
  - Detector Description:
    - AGDD v4: original explicit Atlas Generic Detector Description
    - AliDD: AGDD with additional elements used in Alice
    - AGDD v6: AGDD with arithmetic formulas and connection to RDBS
    - GDML (prototype): Geant4 proposal
  - Events:
    - AtlasEvent XML files
    - AtlantisEvent XML files
- API:
  - Both Generic and Geometric Models can be created directly from Java and C++ (using JACE-created proxies).

Applications

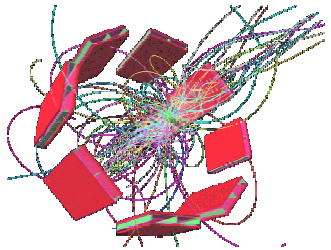


built using GraXML Toolkit (1)

- Detector and Event 3D Display:
  - Very small application build on top of GraXML Toolkit.
  - Complete 3D picture (solid, wireframe, transparency,...)
  - Extensive interactivity (both real and visual properties of objects can be interactively changed)
  - Java scripting via BeanShell (access to full Java environment)
  - Customisable 3D representation
  - Other standard 3D Display features (snapshot, picking, ...)

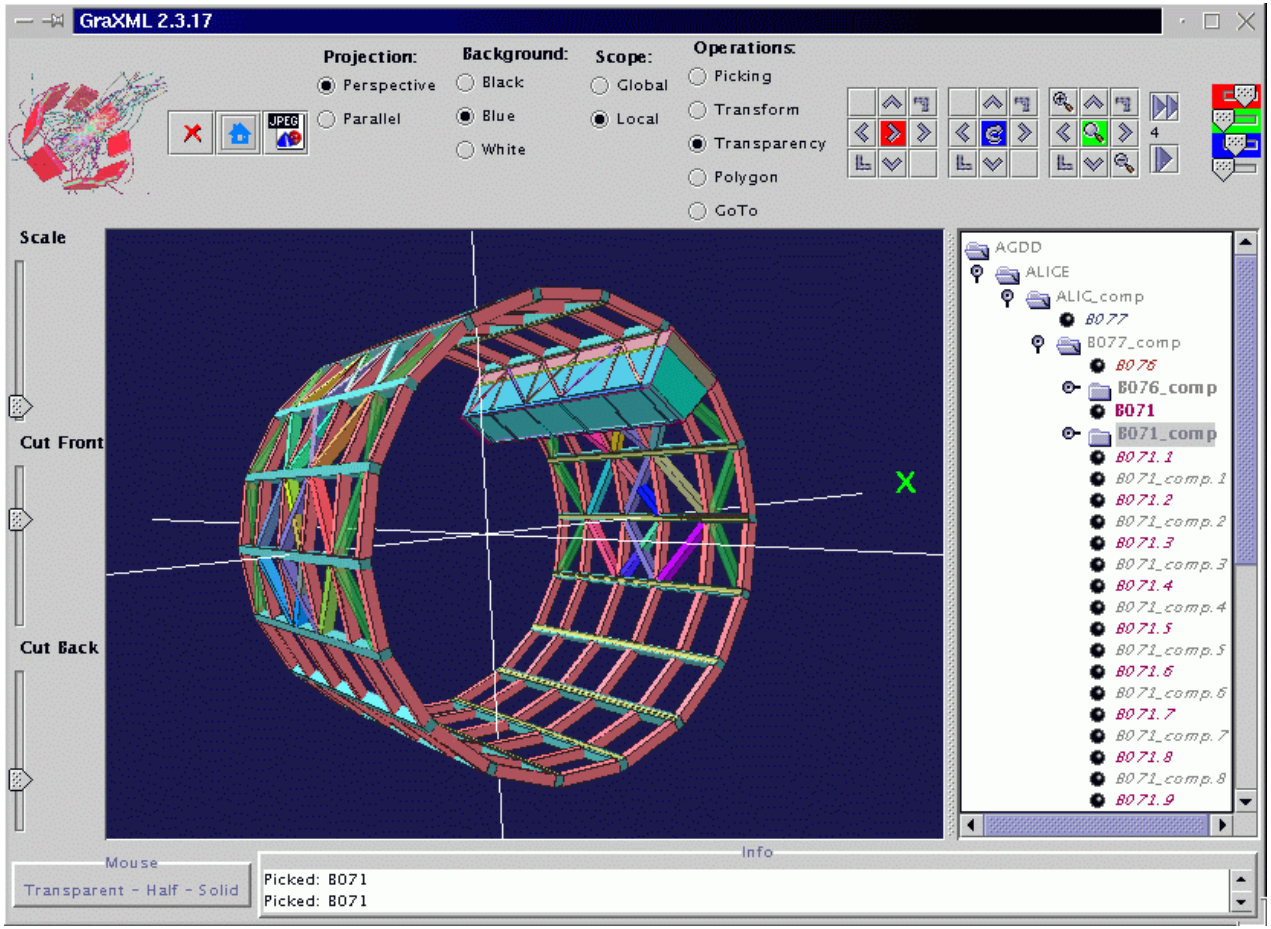
```
Config.setQuality(9);
SelectedColor.setPalette(SelectedColor.ATLANTIS);
TruthTrack.setPtCut(5.0);
Hit.asSphere();
Hit.colorFromKine();
w.show("Test.xml");
j3d.snapshot("Picture.jpg");
```

Script



Applications

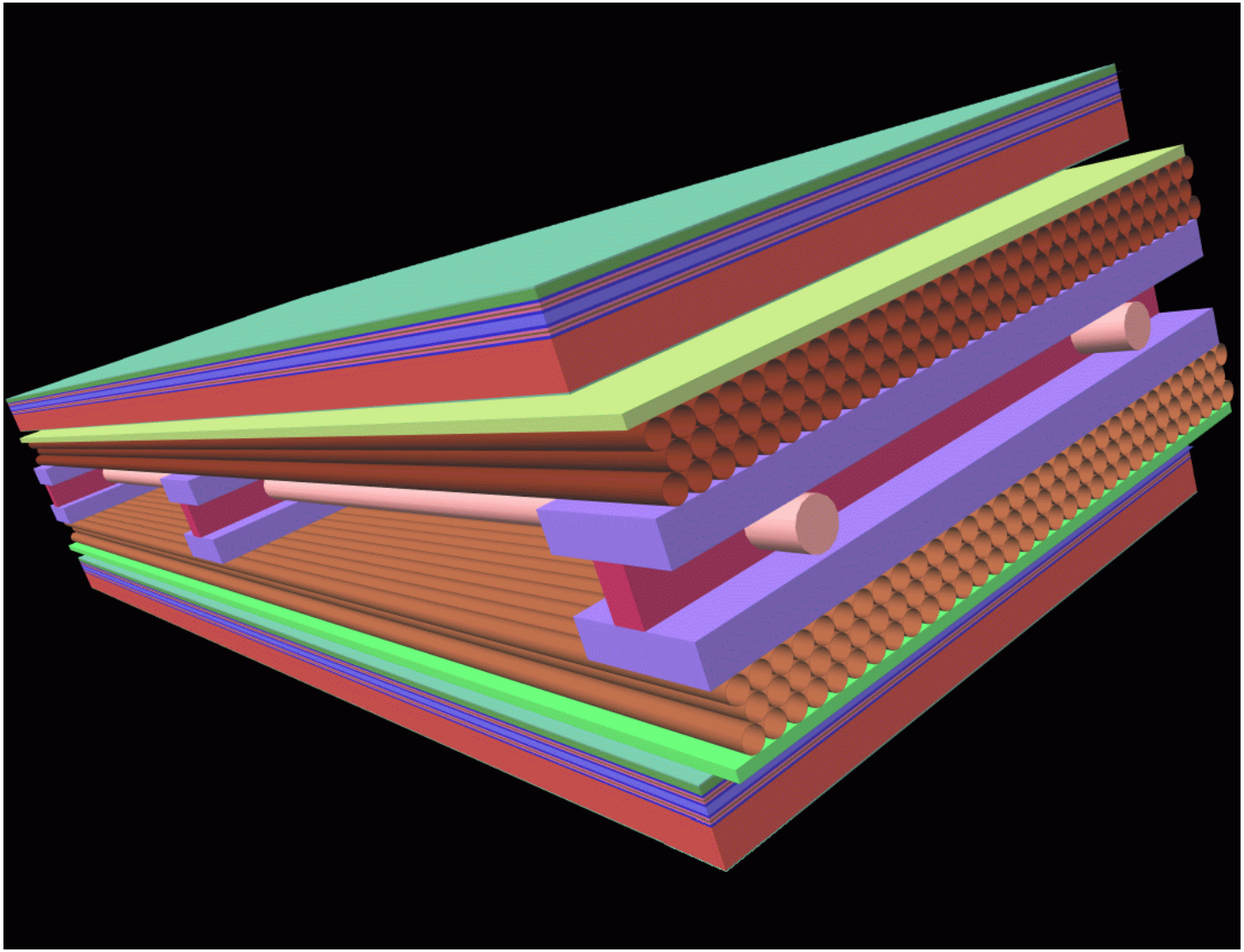
built using GraXML Toolkit (2)



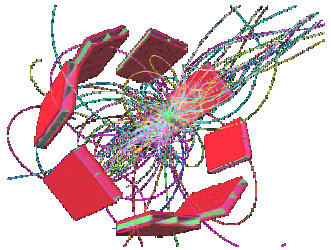
Alice Frame in GraXML (XML file generated from Geant4).

Applications

built using GraXML Toolkit (3)

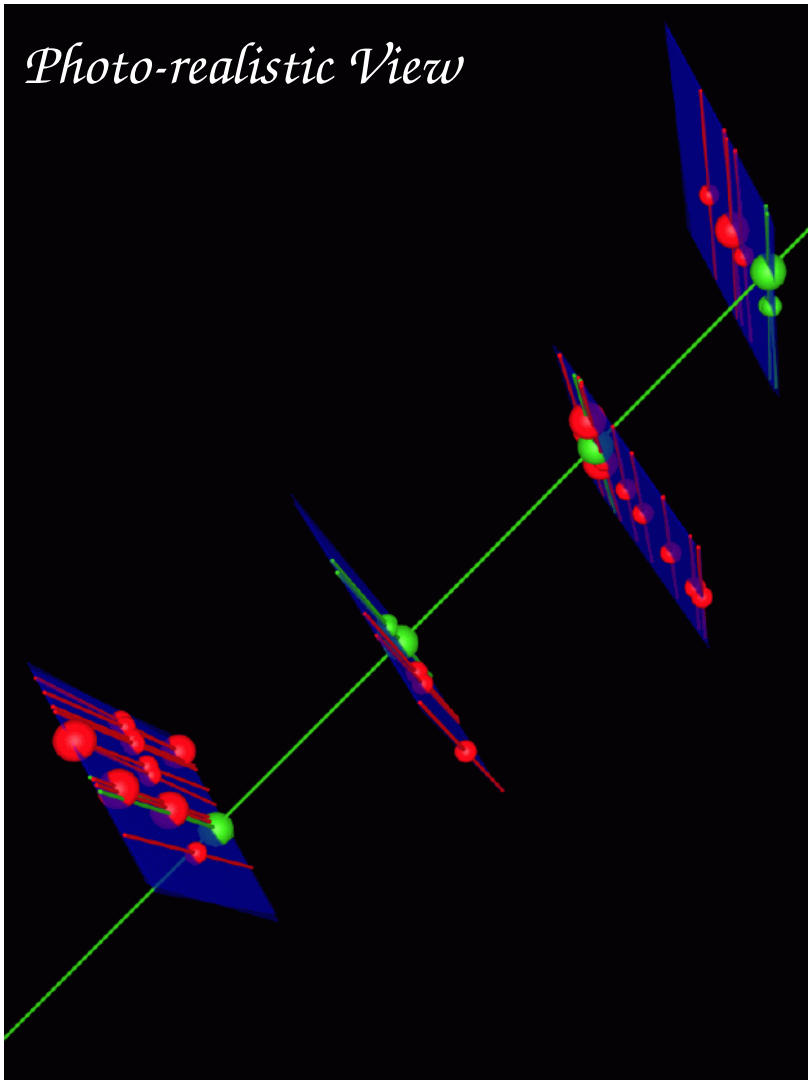


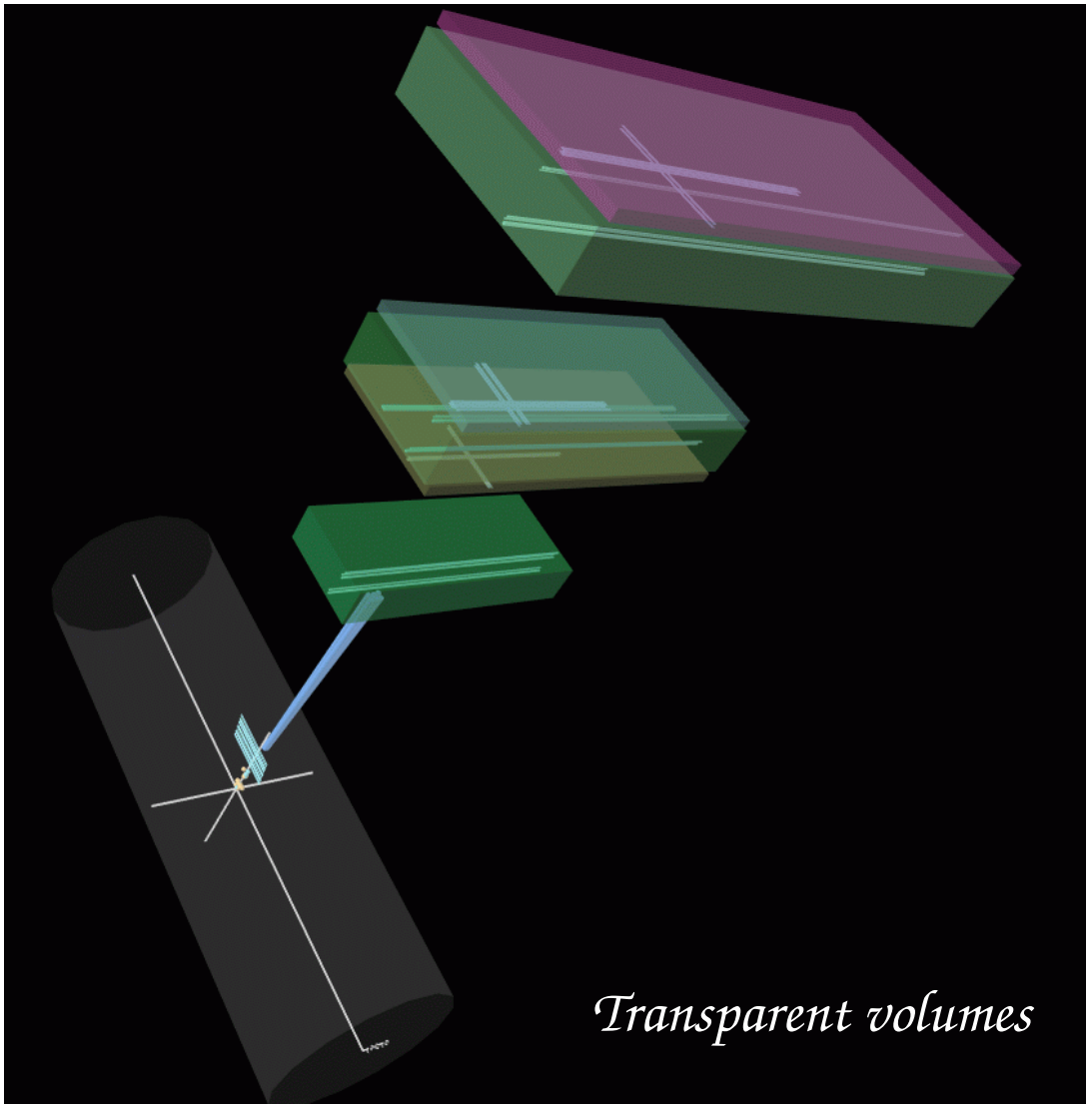
Atlas Muon chamber.



Applications

built using GraXML Toolkit (4)

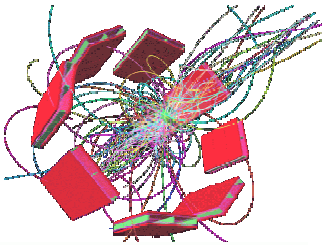


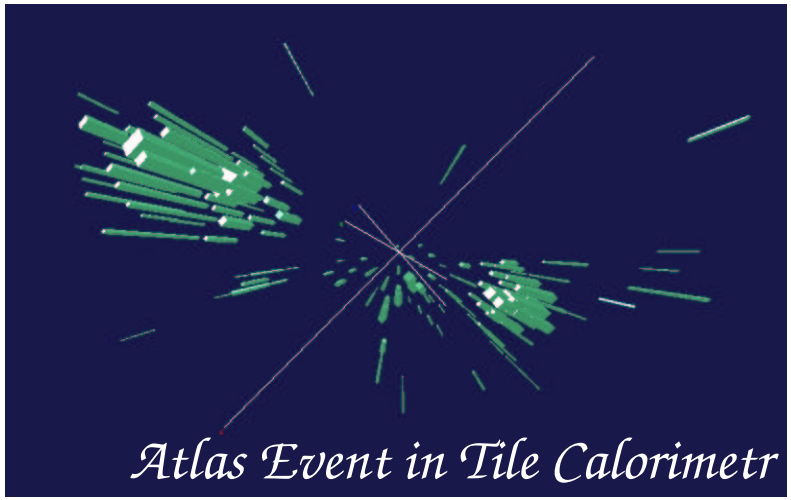


Track in Atlas Inner Detector and Muon Detector.

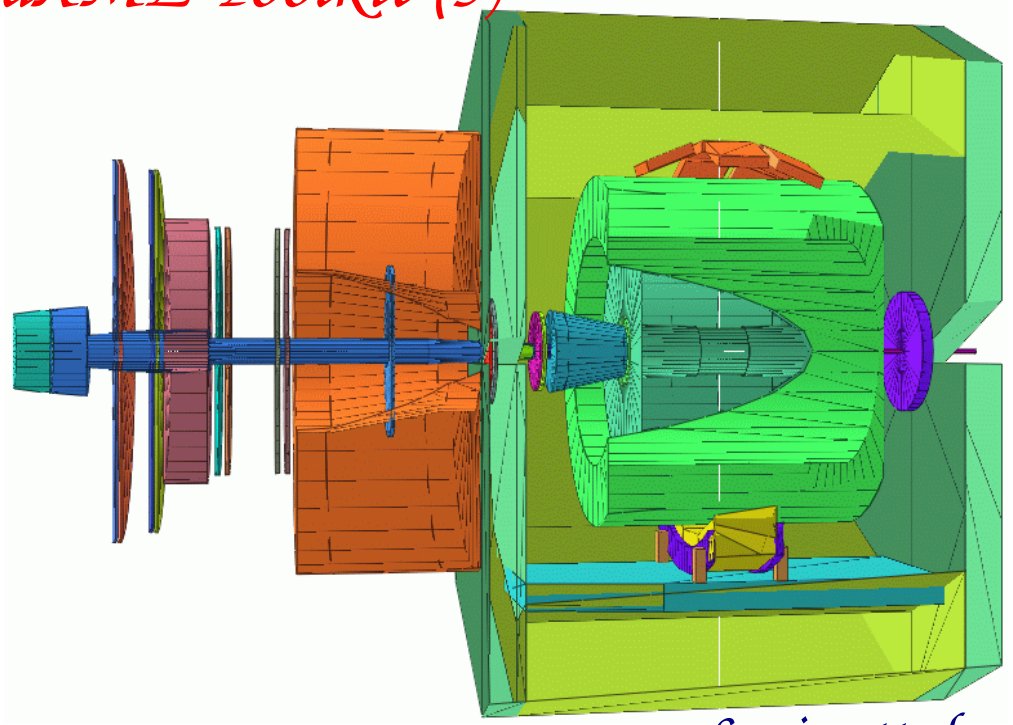
Applications

built using GraXML Toolkit (5)

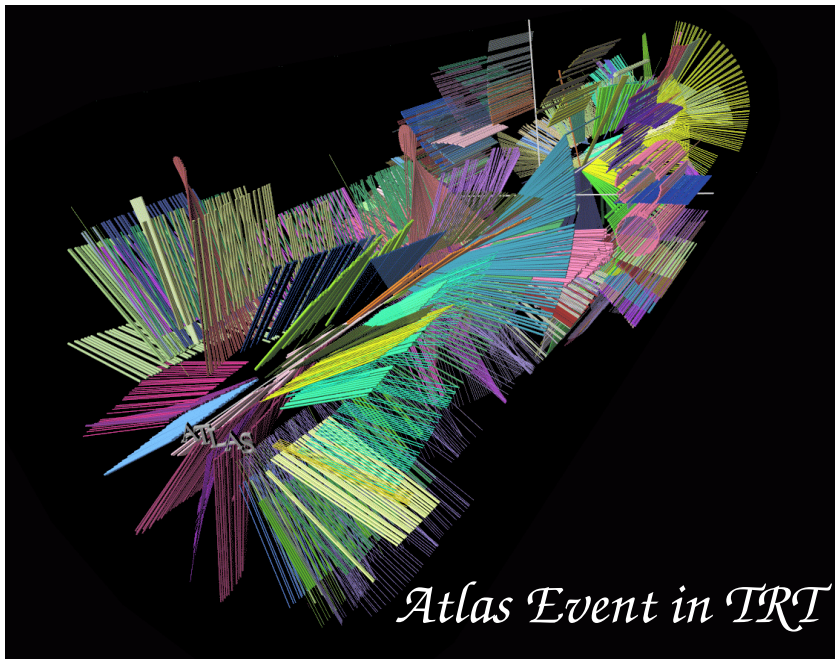




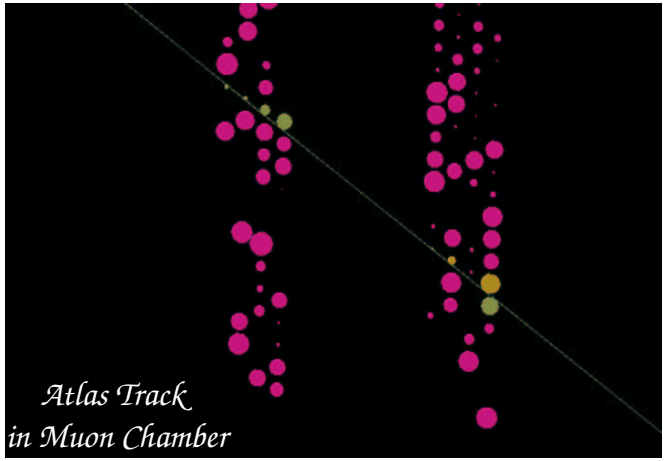
Atlas Event in Tile Calorimetr



Semi-cutted,
semi-transparent
Alice



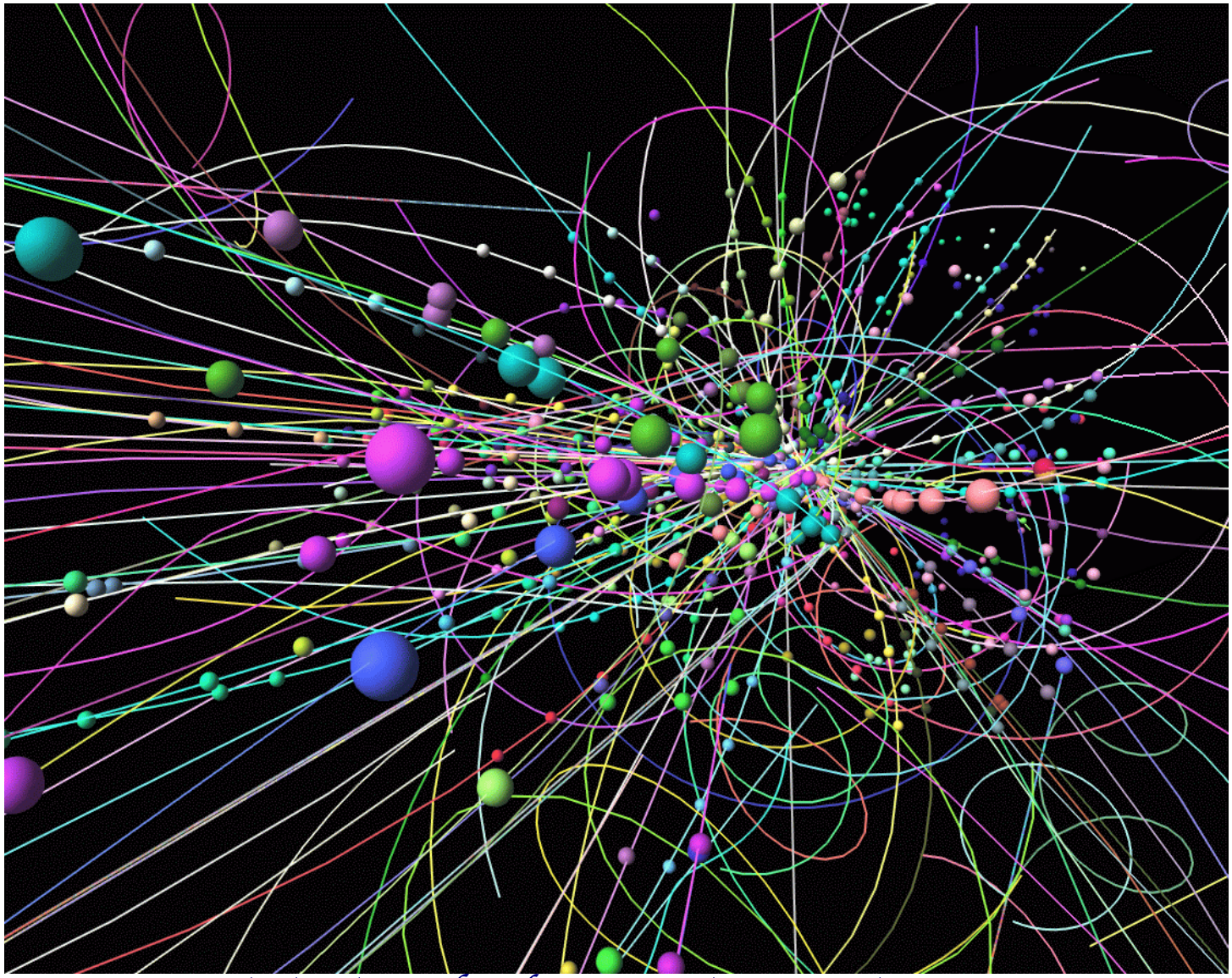
Atlas Event in TRT



Atlas Track
in Muon Chamber

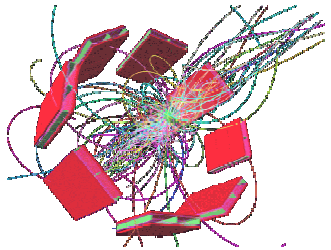
Applications

built using GraXML Toolkit (6)



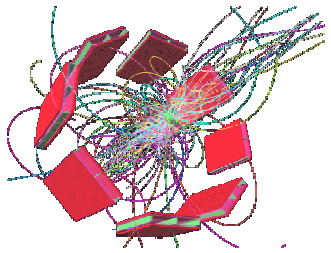
Artistic view of Atlas Event in Inner Detector.

Applications



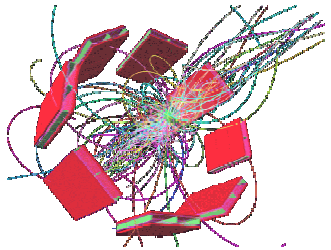
built using GraXML Toolkit (7)

- Exporters:
  - Into VRML/X3D: to be used by any VRML browser or in the 3D Cave.
  - Into TXT: for debugging.
- Converters:
  - Between AGDD v4, v6, v6++ via XSLT stylesheets
  - Between AtlasEvent and AtlantisEvent XML via simple application.
- Importers:
  - Geant4 to AGDD: simple C++ application provided in Virtual MC built in Alice (thanks to I.Hrivnacova).



Summary

- GraXML toolkit provides flexible foundation for modeling of 3D spacial data in HEP.
- Most of the required functionality is provided directly by Java3D, the rest is implemented on top.
- Applications build on GraXML can run in Graphical or non-Graphical mode.



Links

- GraXML: http://hrivnac.home.cern.ch/hrivnac/Activities/Packages/GraXML
  - More details talk: http://hrivnac.home.cern.ch/hrivnac/Activities/2001/September/GraXML
- AGDD: http://atlas.web.cern.ch/Atlas/GROUPS/DATABASE/detector_description
- JAGDD: http://hrivnac.home.cern.ch/hrivnac/Activities/Packages/JAGDD
- GDML: http://gdml.web.cern.ch/gdml
- AtlasEvent: http://hrivnac.home.cern.ch/hrivnac/Activities/Packages/GraphicsAtlasEvent
- NOVA: http://atlassw1.phy.bnl.gov/NOVA/index.php3
- Geant4 Geometry Convertor: http://root.cern.ch/root/vmc/XML.html
- Java3D: http://www.j3d.org
- JAXB: http://java.sun.com/xml/jaxb
- Author: http://hrivnac.home.cern.ch/hrivnac