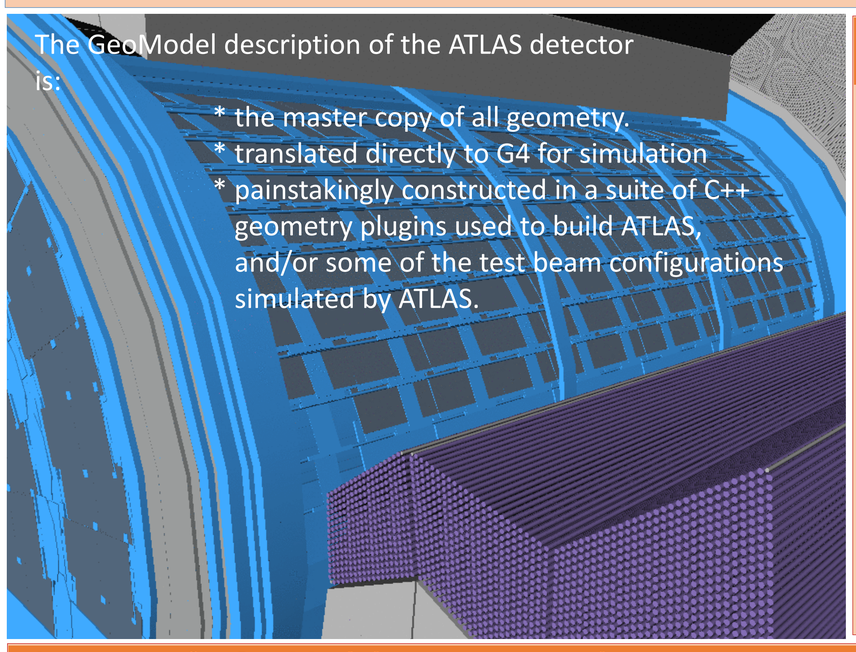
THE PERSISTIFICATION OF THE ATLAS GEOMETRY

Riccardo Maria Bianchi, ***Joe Boudreau***, Ilija Vukotic
University of Pittsburgh, PACC. Email: boudreau@pitt.edu

CHEP 2016
San Francisco (California)

The master copy of all **detector geometry** in the ATLAS software is the **GeoModel**, a graph of **geometrical primitives** resembling a scene graph. This geometry graph consists of both placed volumes and parameterized volumes. Detector factories running within a geometry service read a database of primary numbers and build a highly detailed description of ATLAS, which is then transferred to Geant 4. The procedures to build the GeoModel are complex, depend on a large stack of ATLAS software, and lack portability. ***We dump the whole description into a database and provide a portable mechanism to read it back in.***



In-memory model vs database.

A modern programming language with control structures, loops, a math library, etc. is probably needed to construct ATLAS geometry from raw numbers and arrange for access to information used in hit processing.

These calculations are carried out within the geometry plugins and are in the hands of detector subsystem experts.

The plugins may depend on a large amount of ATLAS core Software and its *Athena* framework and subsystem software.

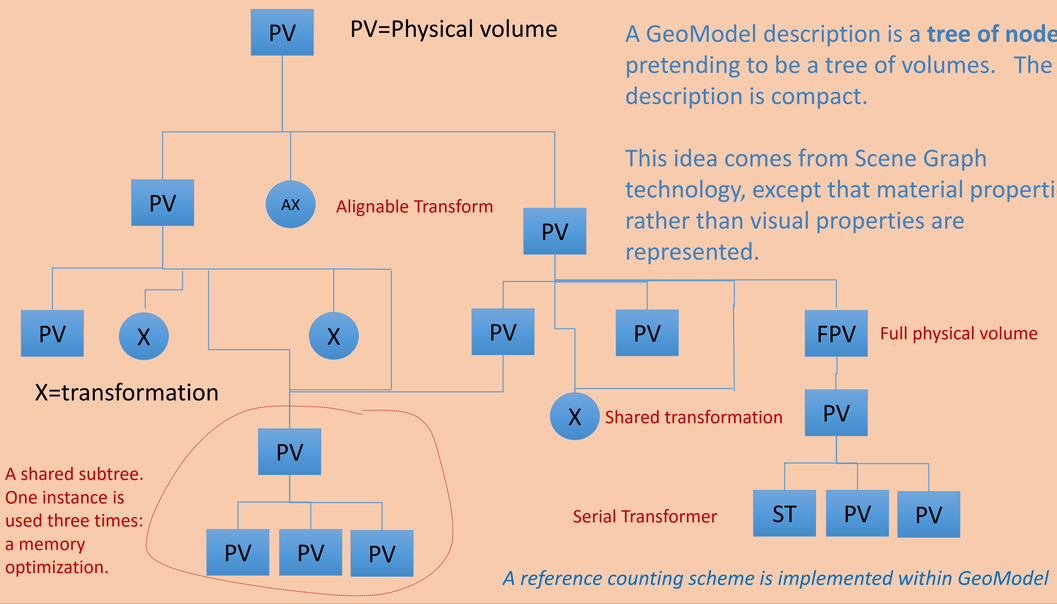
The ATLAS geometry cannot be instantiated outside of an *Athena* job:

- Difficult to port the geometry builders into lightweight applications designed to run without the full ATLAS software stack.
- Difficult to import geometry code in ATLAS-neutral apps.

Note: the CPU and memory requirements of the GeoModel are modest. Geometry Persistification is about **portability**.

System	Memory (Mb)	CPU (s)
Pixel	12.8	0.35
SCT	13.3	0.14
TRT	22.2	0.18
LAr	32.3	1.45
Tile	14.7	0.36
Muon	36.1	0.79
Total	**131.4**	**3.27**

Anatomy of a GeoModel geometry graph:

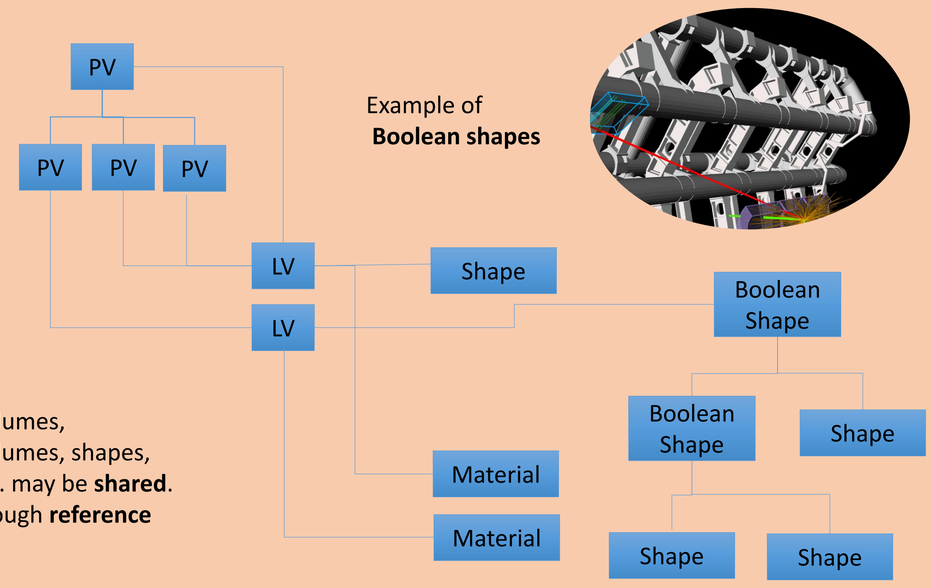


A GeoModel description is a **tree of nodes** pretending to be a tree of volumes. The description is compact.

This idea comes from Scene Graph technology, except that material properties rather than visual properties are represented.

That is not all…

Logical volumes
Shapes
Materials
Boolean Shapes

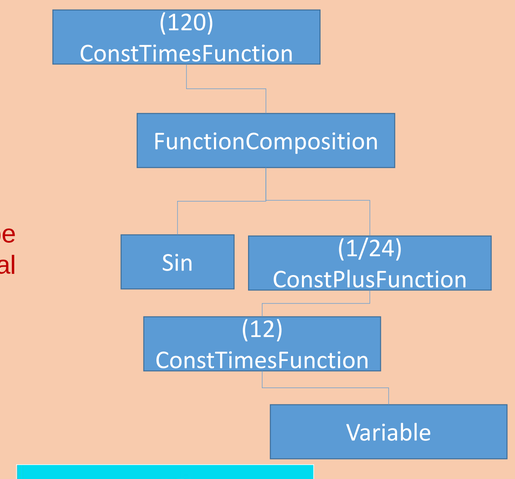


Example of **Boolean shapes**

All instances of physical volumes, transformations, logical volumes, shapes, materials, Boolean shapes.. may be **shared**. Memory management through **reference counting**.

Serial transformers: more acyclic graphs:

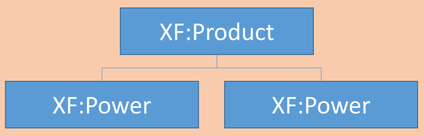
$$120 * \sin(12 * i + 1/24)$$

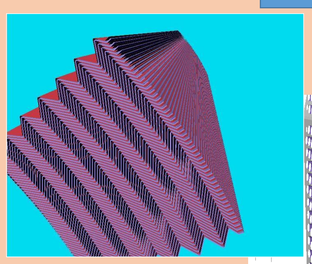


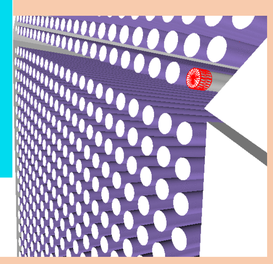
In **CLHEP** mathematical expressions are binary expression tree.
The interface is natural.

In the geometry kernel: A totally generic recipe for parameterizing transformations (of physicial volumes, including subtrees):

$ROT_x^{f(i)} * ROT_y^{g(i)} * TRANS_x^{h(i)}$ ….. & cetera







```
const unsigned int NPLATES=100;
Variable    i;
Sin         sin;
GENFUNCTION f = 360*deg/NPLATES*i;
GENFUNCTION g = -sin(4*f);
TRANSFUNCTION t1 = Pow(HepRotateZ3D(1.0), f)*
                   HepTranslateX3D(1100*cm)*
                    Pow(HepTranslateZ3D(800*cm),g);
```

Peristify the whole geometry in a database.

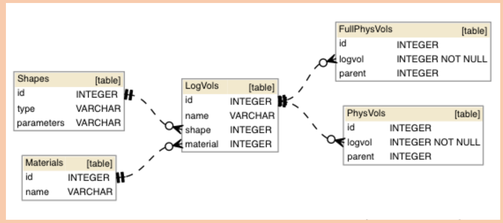
SQLite is the chosen DB technology
Qt5 (Module QtSql, Class QSqlDatabase) provides a convenient interface to to SQLite.

Elements of the geometry tree are stored in the appropriate table. Shared instances appear only once in the table. We don't want to bloat the database.

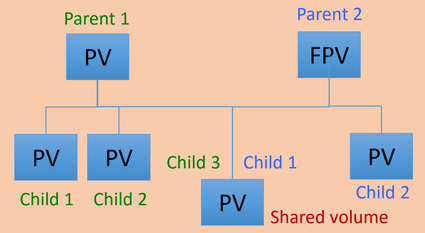
Tables reflect the entities in the GeoModel:

- GEOPHYSVOL
- GEOFULLPHYSVOL
- GEOLOGVOL
- GEOMATERIAL
- GEOSHAPE
- GEOSERIALDENOMINATOR
- GEOSERIALTRANSFORMER
- GEOTRANSFORM
- GEOALIGNABLETRANSFORM
- GEONAMETAG
- FUNCTIONS

Classes TransfunctionPersistifier and TransfunctionReader store the transfunction expression tree as a nearly human readable string.

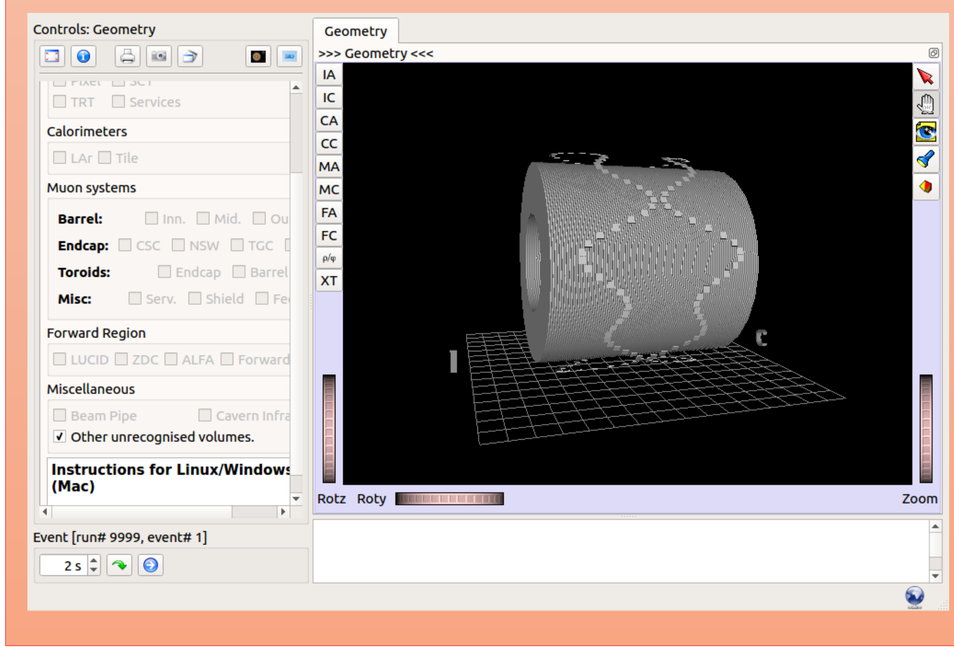


*A **link table** CHILDRENPOSITION establishes the parent-child relationships between nodes in the geometry graph. Order is important.*



Status of persistication: sample geometries and some pieces of ATLAS have been **successfully** persistified.

Visualization



VP1 Light is the first **client** of persistified geometry.

Here you can see it working with a fake geometry.

VP1 Light will be used to debug the geometry persistification.

Other clients include a **3D virtual reality** display based upon Oculus 3D headsets.

A **JSON** export of the database is being developed For that.

A C++ API to read and write Geometry, independent of ATLAS software

- The Geometry "Kernel" (GeoModelKernel) is disentangled & extracted from other ATLAS software; external dependencies are:
  - CLHEP (for GENFUNCTION, vectors, transformations)
  - SQLite is the database
  - Qt is not really needed but it's got a super convenient interface to SQLite and a superconvenient platform independent build system (qmake)
- Its function:
  - Dump a GeoModel tree to a SQLite file. For the whole ATLAS geometry the result is a very compact file whose size is *O(Mb).*
  - Read in a file and instantiate a GeoModel geometry tree.
  - Other geometries can also be built & saved.
- Applications:
  - A lightweight version of the VP1 event display depends on a lightweight geometry system. This is coming in the near future.
  - Other fancy 3D visualization systems, and ultimately other applications, can soon read the ATLAS geometry either through the database directly or through the geometry API.
- The future:
  - High potential for reusability as a software component.
  - Opens the door to lightweight Visualization APIs, too.
  - Geo2G4 Converters could also be disentangled from ATLAS code and included in the API

Summary

The GeoModel geometry system has been in service in ATLAS for more than a decade and is a fully mature system. The possibility of saving and restoring the geometry has been envisioned from the outset, and is finally being realized with a short term goal of permitting a lightweight event display. The design has been carried out the goal of complete faithfulness to the original representation, CPU efficiency and low memory cost. It will be part of a lightweight ATLAS application soon and can be made public if there is interest in that.

Thank you!