Vukotic
unsigned
and
Other geometries can also be built & saved.

used to build ATLAS, and some of the test beam configurations
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 in 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.

Tree to a SQLite file. For the whole ATLAS geometry the total memory model vs database.

High potential for reusability as a software component.

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.

A lightweight version of the VP1 event display depends on a lightweight visualization system. This is coming in the near future.

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:
  - GeoModel (for GENFUNCTION, vectors, transformations)
  - CLHEP (for GENFUNCTION, vectors, transformations)
  - Generically constructed in a suite of Geo
  - geo SHAPES and other geometries.

• It is not really needed but it’s got a super convenient interface to SQLite and a superconvenient platform independent build system (gmake)

• 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 0(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.

In-memory model vs database.

That is not all...

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

Peristify the whole geometry in a database.

SQLite is the chosen DB technology Q55 (Module Q55), Class Q55Database provides a convenient interface to to SQLite.

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

Visualization

VP1 Light is the first client of persitified geometry.

Here you can see it working with a fake geometry.

VP1 Light will be used to debug the geometry persitification.

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

A JSON export of the database is being developed for that.

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.