



VecGeom integration in Geant4 and ROOT

Gabriele Cosmo (CERN), Mihaela Gheata (CERN), Guilherme Lima (FNAL)

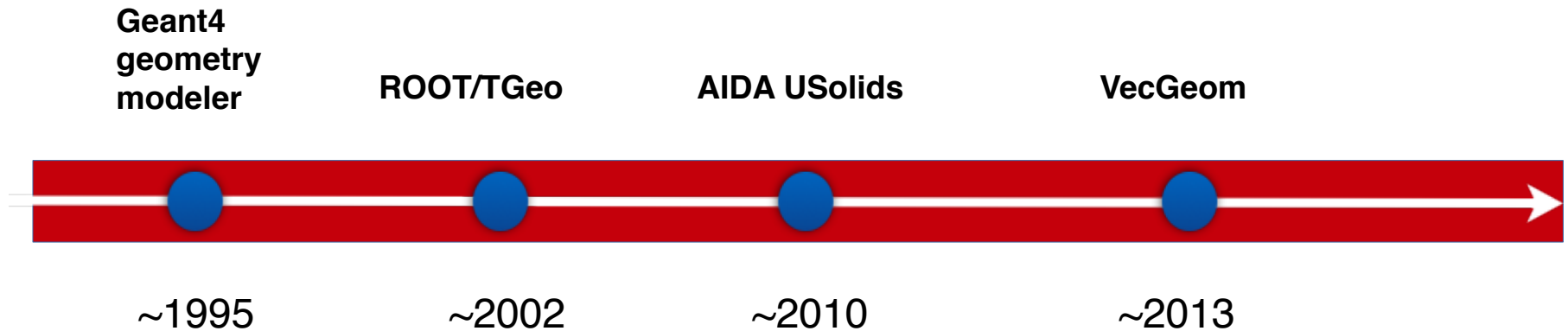
for the VecGeom developers



Introduction

- Goal is to allow current Geant4 and ROOT geometry users to transparently migrate their applications to use VecGeom shapes as soon as possible, with no changes in their application code
 - a short-term milestone on the path to full GeantV deployment
- Advantages for the user: some potentially significant performance improvement
- Bonus: existing Geant4 and ROOT debugging tools can be used to help find and fix bugs, for visualization and for validation and stress testing of VecGeom shapes.

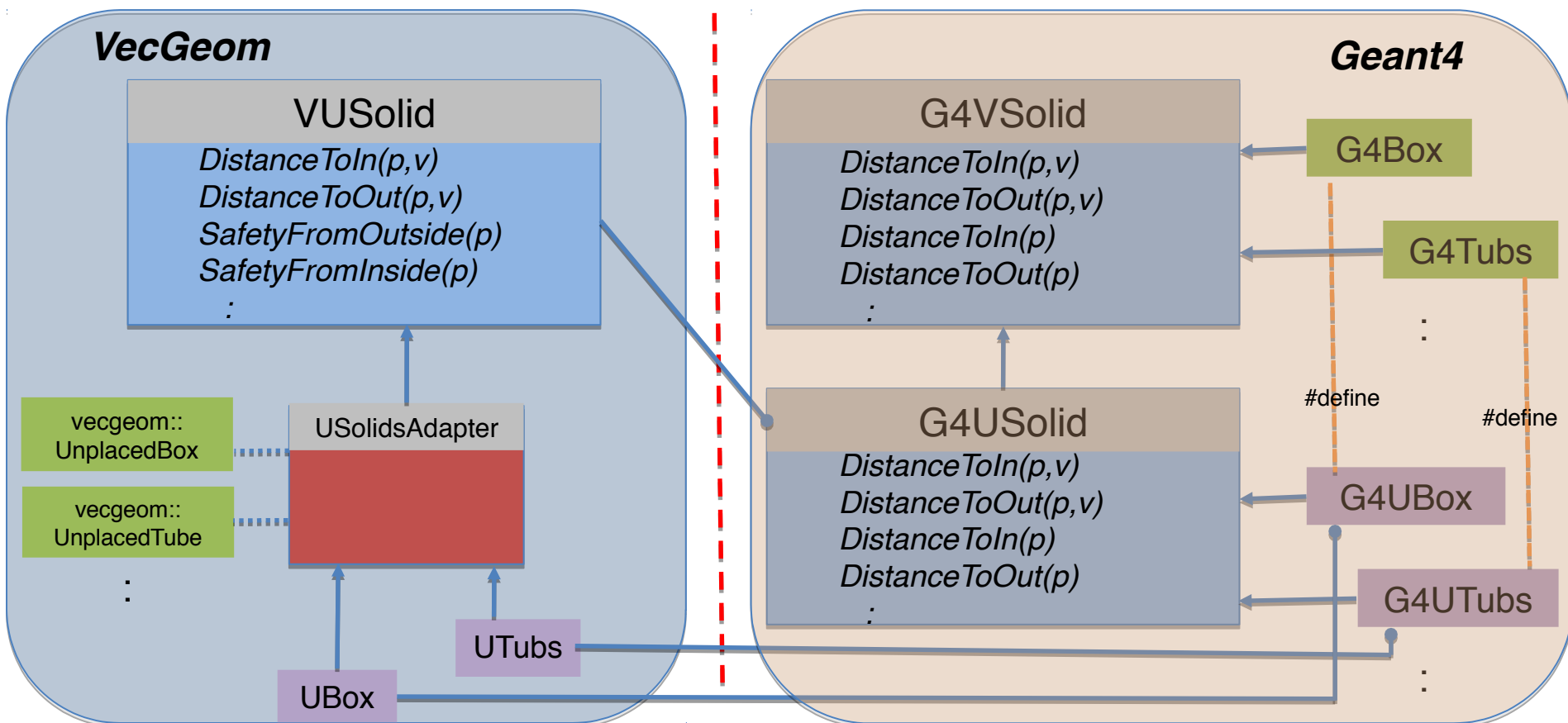
USolids to VecGeom



- USolids (Unified Solids) library started as AIDA project
 - Goal: a new library of geometrical primitives to unify algorithms existing in Geant4 and ROOT
- VecGeom: started as feasibility study of vectorization for geometry
- Geometry primitives code development as long-term evolution of USolids
 - Developed back-to-back with USolids as independent library
 - VecGeom now incorporating USolids implementation and sharing same interfaces
 - Will replace USolids in the medium/long term
 - Part of AIDA-2020/WP3

Bridging shapes to Geant4

- Bridge pattern design to USolids/VecGeom
- Choice of activation of individual wrappers at installation of Geant4



Building Geant4 to use VecGeom

a. Install VecGeom package

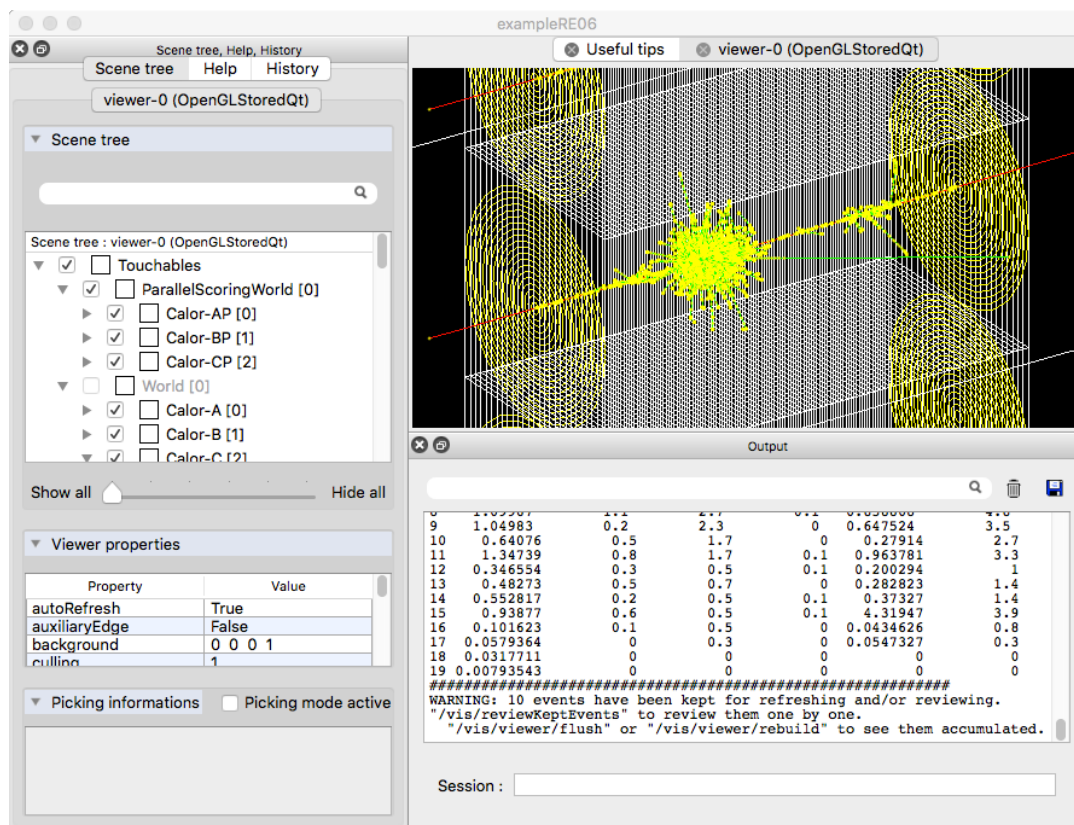
- Specifying "Scalar" backend:
`-DBACKEND=Scalar`
- Configuring to use VecGeom implementation:
`-DUSOLIDS_VECGEOM=ON`

b. Install Geant4

- Specifying to use USolids/VecGeom installation:
`-DGEANT4_USE_USOLIDS="all"`
- or specifying individual shapes:
`-DGEANT4_USE_USOLIDS="box;tubs"`

Using VecGeom from Geant4

- VecGeom solids can be used as normal G4 shapes/types since Geant4 10.2, as an external library
- Part of nightly Geant4 build testing
 - VecGeom shapes from Git master, updates are automatically propagated into the nightly tests
- All Geant4 features supported
 - Included multi-threading (MT)
 - Ability to select shapes to replace at installation, since 10.2.p02
- Being tested on realistic geometry setups with full physics
 - CMS, LHCb full geometries
 - Currently facing correctness and robustness issues from Polycone shape



(*) Full interactive Geant4 application with parallel geometries, parameterisations and scoring

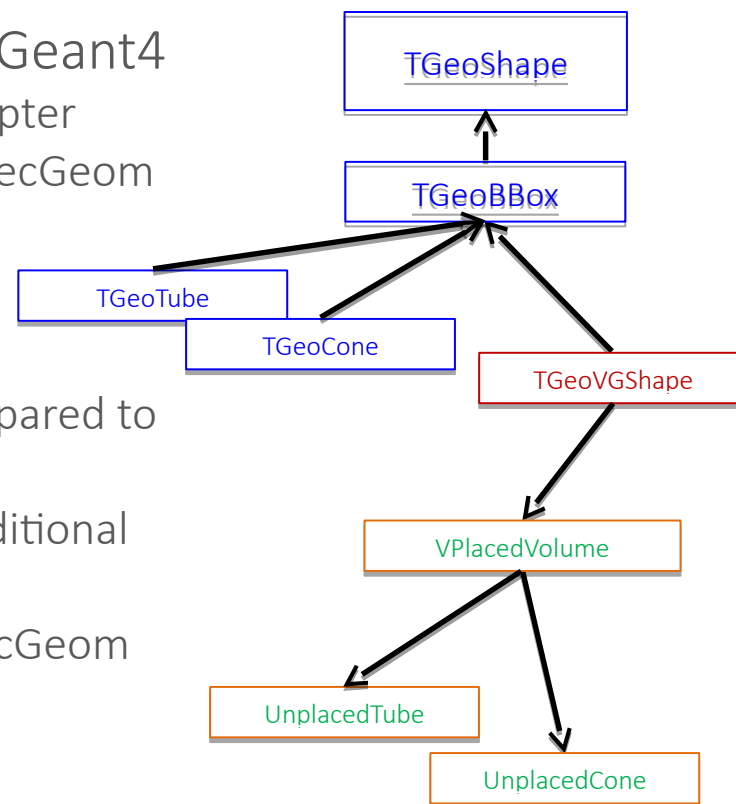
Future plan: provide specialized navigation interface class to VecGeom navigation

Expected features in Geant4 10.3

- Release 10.3 of Geant4 planned for December 2016
 - Beta preview (10.3-beta) released end of June
- Ability to exercise a limited set of shapes from VecGeom
 - Box, Tube, Cone, Orb, Sphere, Trap, GenericTrap, Trd, Torus, Polycone, Polyhedron
 - Aiming to approach “production quality” in terms of correctness and robustness for those shapes
 - Expecting few % CPU speedup from most complex VecGeom shapes in realistic detector geometries
 - Coming with improved geometry optimisation (*smart-voxels* structure)
- Geant4 CMake build system extended to enable use of VecGeom shapes individually (already available in 10.2.p02 and 10.3-beta)
- Ability to scale shapes along Cartesian axes
 - Extended GDML schema (new version 3.1.4)

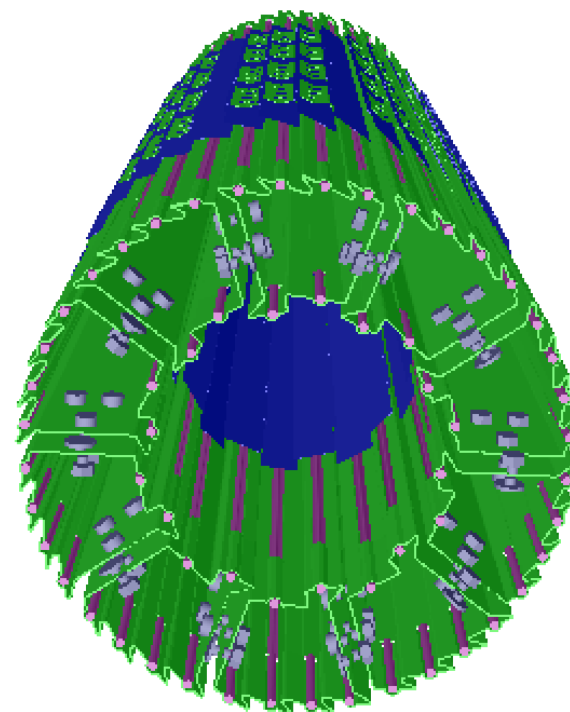
VecGeom solids bridge in ROOT geometry

- Same idea as using Usolids/VecGeom in Geant4
 - Call VecGeom solids behind a TGeoShape adapter
 - Replace TGeo shapes runtime with adapted VecGeom solids
- Objectives
 - Better scalar performance in many cases compared to native shapes
 - Allow to cross-check VecGeom solids with additional TGeo tools (ray-tracing, visualization checks)
 - First step in a long term transition to using VecGeom full navigation behind ROOT TGeo



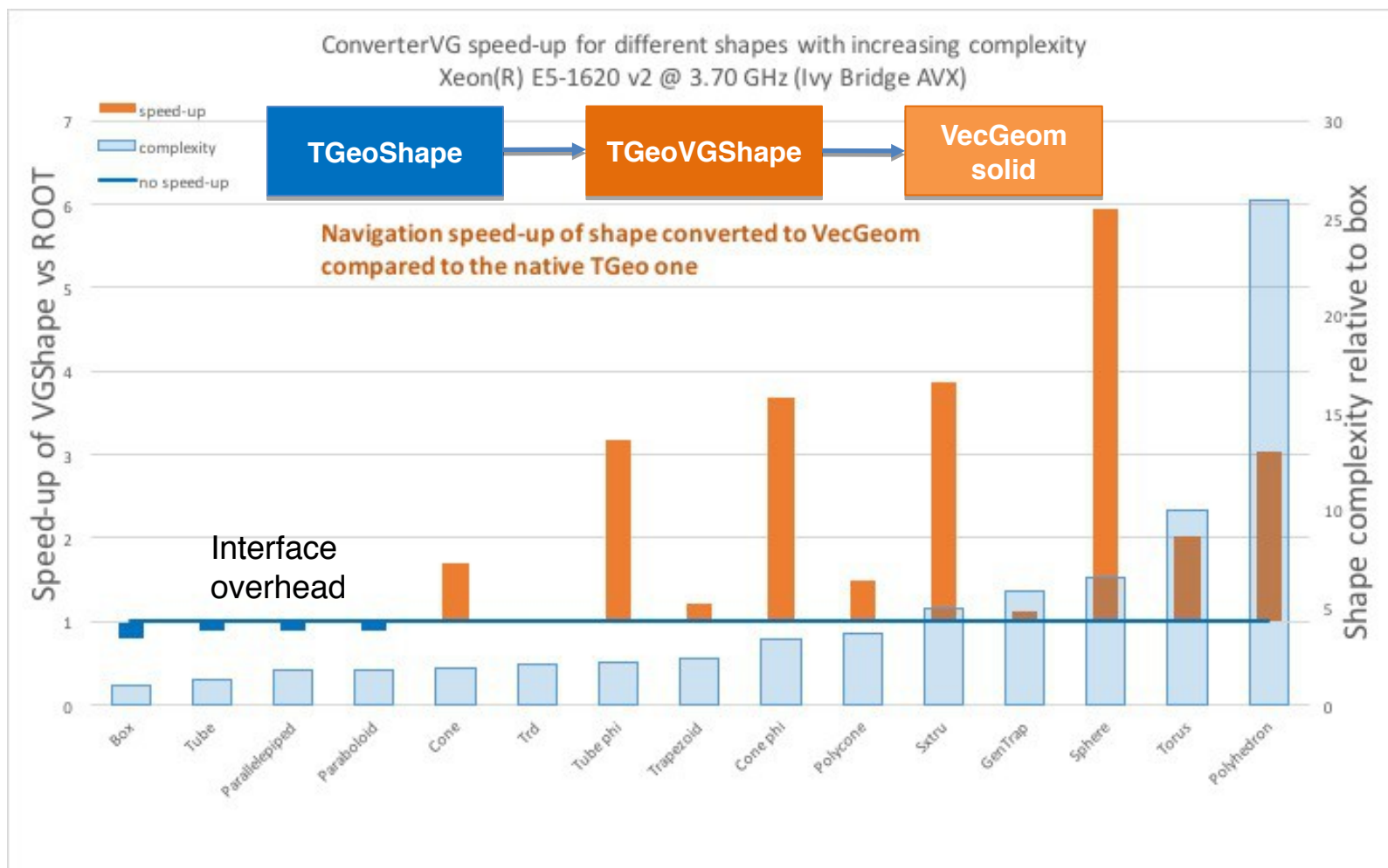
Implementation of adapter/converter

- On-demand compilation and usage
 - Library loaded using ROOT plugin manager
 - Use on demand:
`TVirtualGeoConverter::Instance()::ConvertGeometry()`
- Adapter class deriving from TGeoShape
 - Providing only distance/contains functions:
 - *`Contains, DistanceToIn/Out, SafetyToIn/Out, Normal`*
 - Original ROOT shape pointer kept for other functionality (e.g. visualization, alignment)
- Converter parsing all volumes in TGeo geometry
 - Replacing shapes with the adapted ones from VecGeom



Ray-traced image of ALICE ITS-SPD detector using navigation over converted VecGeom shapes

Conversion performance



Conclusion

- VecGeom shapes ready to be used from either Geant4 and ROOT
- VecGeom expected to become the default geometry library for use in Geant4 and ROOT
- Production quality expected by end of this year for a large set of primitives
- Promotes adiabatic adoption of GeantV from the experiments

Future plans

- Progress in the implementation of the missing shapes from the standard set (tessellated solid, extruded solid, generic polycone, etc...)
 - Also part of the AIDA-2020 program of work
- Extend testing coverage for all shapes and their possible topologies
- Extend testing on realistic geometry setups by including new snapshots from LHC detectors
- Interface VecGeom navigation algorithms in Geant4 and ROOT

Backup slides

Building Geant4/ROOT to use VecGeom

a. Install VecGeom package

- Configure to either use VecGeom implementation:

```
cmake -DBACKEND=Scalar -DGEANT4=OFF -DUSOLIDS=ON -DUSOLIDS_VECGEOM=ON \  
  [...other optional VecGeom switches as needed...] \  
  -DCMAKE_INSTALL_PREFIX=${VecGeomINSTALLDIR} ${VecGeomSOURCE} \  
make install
```

b. Install Geant4

- `export USolids_DIR=${VecGeomINSTALLDIR}/lib/Cmake/USolids`
- Add `-DGEANT4_USE_USOLIDS="ALL"` when configuring Geant4 with Cmake

c. Install ROOT

- `-DCMAKE_PREFIX_PATH=<path_to_vc>;<path_to_vecgeom> -Dvc=ON -Dvecgeom=ON`