Common Geometry Primitives library

A proposal

Gabriele Cosmo, PH/SFT

Outline

- The context
- Main motivations
- Proposed strategy
- Current plan & resources

Geometry primitives

- Geometrical representations of the detector elements for solid 3D modeling
  - Used in detector simulation: "solids"
  - Include high precision and efficient algorithms and functions for dealing with tracking and navigation techniques
  - Provide tools for computing area and cubic volume of a shape
  - Include hooks for graphical representation and object persistency
- Different implementations existing today
  - Geant4: up to 23 primitives defined (limited to CSG and specific solids)
  - Root: similar set of primitives, sharing most of the functionalities and concepts
- GDML schema defining a super-set of all primitives implemented in Geant4 and Root

Motivations for a common library

- Optimize and guarantee better long-term maintenance
  - A rough estimation indicates that about 70-80% of code investment for the geometry modeler concerns solids
  - Most of the maintenance effort and debugging is on solids, to guarantee the required precision and efficiency in a huge variety of combinations
- Target to create a unique library of high quality implementation
  - Starting from what exists today in Geant4 and Root
  - Adopt a single type for each shape
- Converge on a unique description and interface
  - Allowing to reach also a common persistency GDML schema for solids
- Optimize, extend and rationalize the testing suite

A first analysis

- Examined key methods of Solid interface
  - No showstoppers were found in comparing function capabilities
  - Some key differences identified:
    - Adoption of geometrical tolerance for surface thickness in Geant4, which is no present in Root
    - Mechanism for computing a solid extent is slightly different
    - First plan is to support extra Geant4 functionality (some are certainly mandatory, some potentially optional)
  - Keep auxiliary methods (volume, area, sampling surface points, …)
- Persistency mechanism will be kept external
  - Provide constructors (default or "special") which will be used by persistency solutions
- Visualisation-enabling method (graphics representation) appear similar

Proposed strategy

- Stage ONE: Startup

  a) Birth of type and interface
  - Define common interface
  - Define and implement bridge classes
    - enabling use of new interface, new solids in Root/Tgeo and Geant4
  - Implement a first (test) common solid: Box

  b) Deploy testing suite and existing testing tools for common use
  - Eventually extend and improve what existing, ….

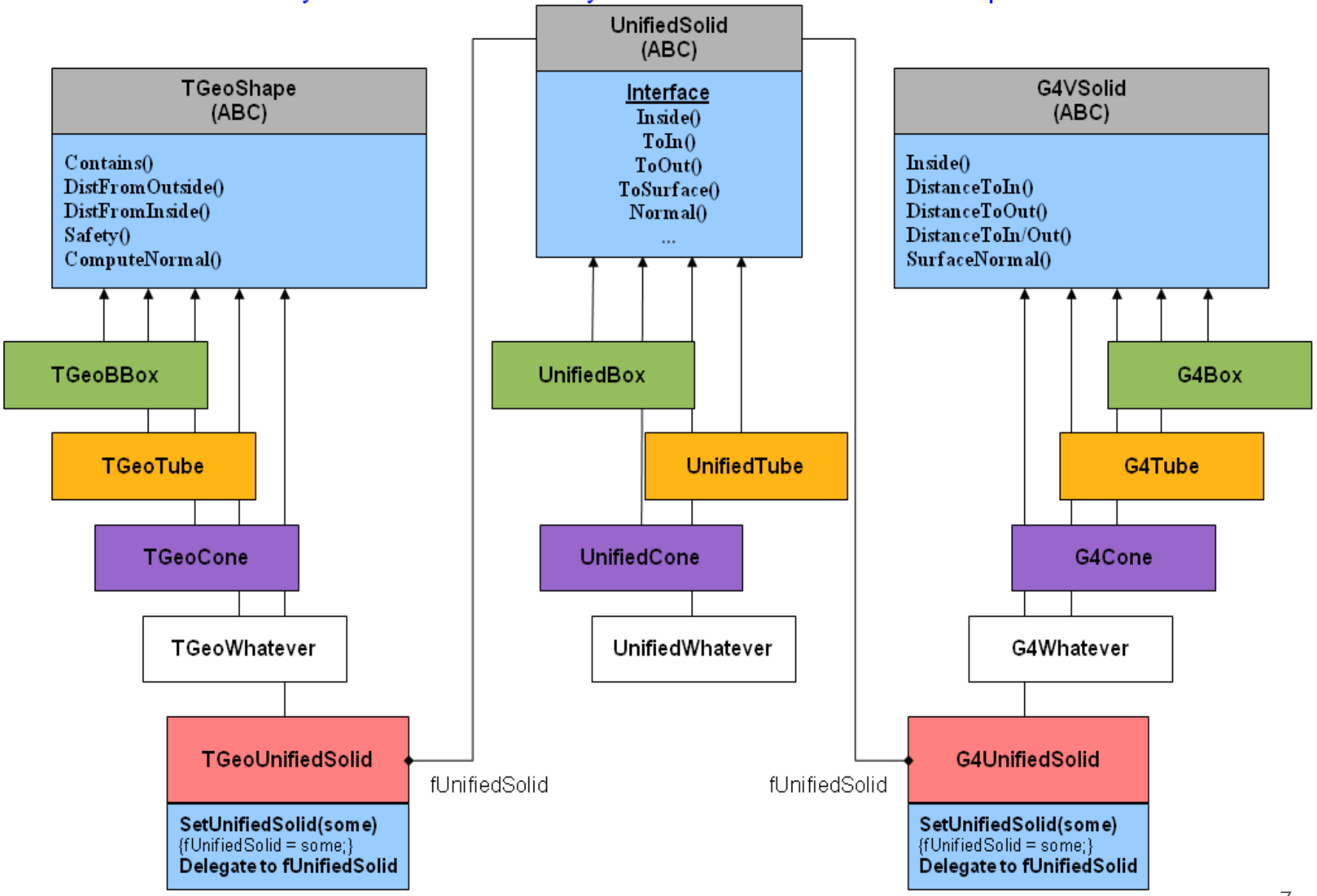
  c) Design new "Union of Many" volume

- Stage TWO: Migration

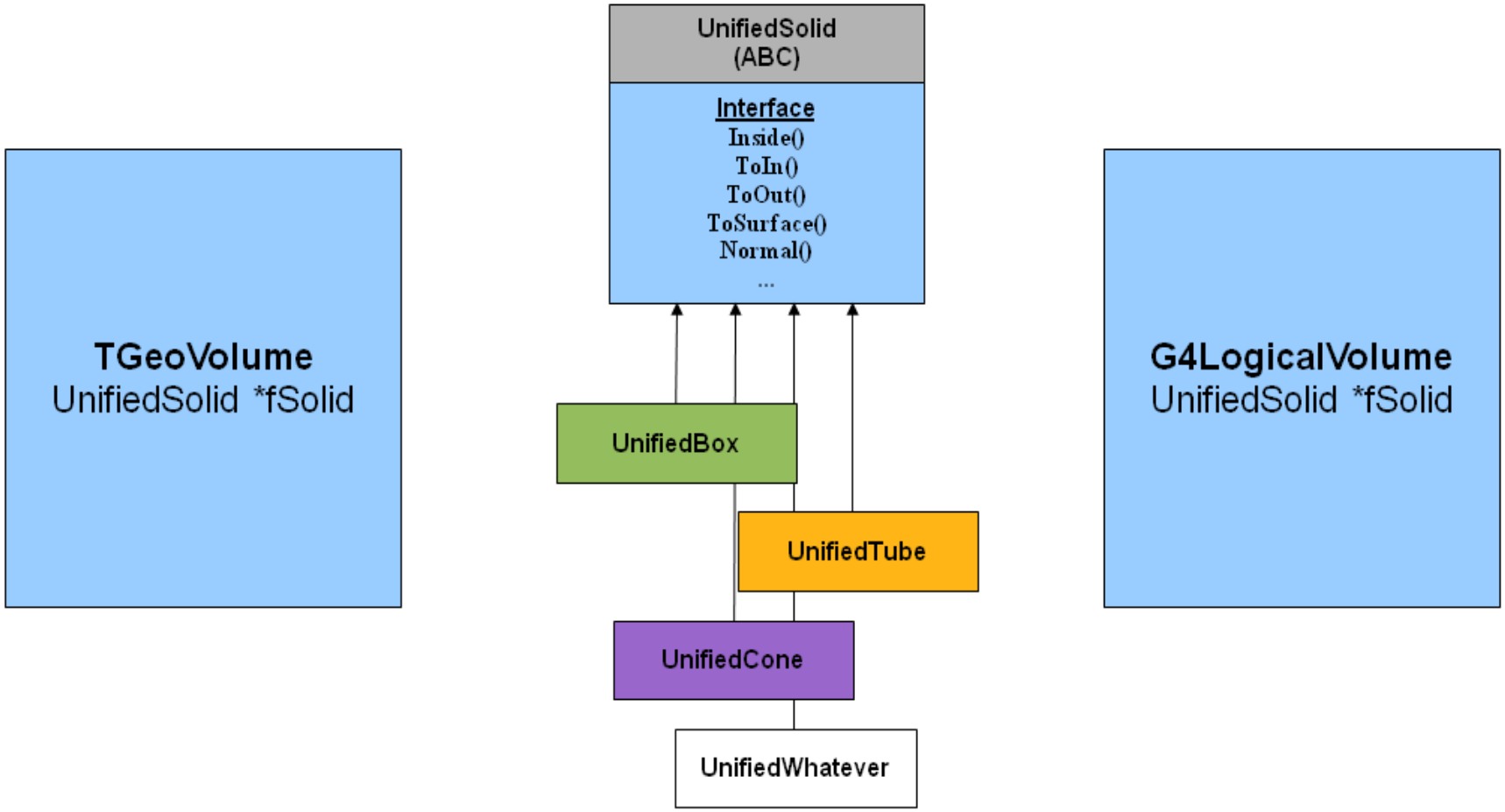
  a) Evaluate weaknesses of solids for priority
  b) Implement migration of each solid according to priority

Bridge classes for a smooth transition

Any solid in the unified library can be seen as a native TGeoShape or G4VSolid



Library used directly by TGeo and GEANT4 after consolidation



Current resources

- Assuming current resources sum up to ~0.4 FTE, adding contributions from
  - Andrei Gheata (ALICE)
  - John Apostolakis (PH/SFT)
  - Tatiana Nikitina (PH/SFT)
  - Astrid Munnich (PH/SFT)
  - Gabriele Cosmo (PH/SFT)
- Estimate to complete stage ONE in 2011 with current resources
- Additional 1 FTE for 1.5-2 years required in order to achieve stage TWO

Thanks!