The Virtual Geometry Model

I. Hrivnacova
IPN, Orsay

Conference for
Computing in High Energy and Nuclear Physics,
Interlaken, 30 September 2004

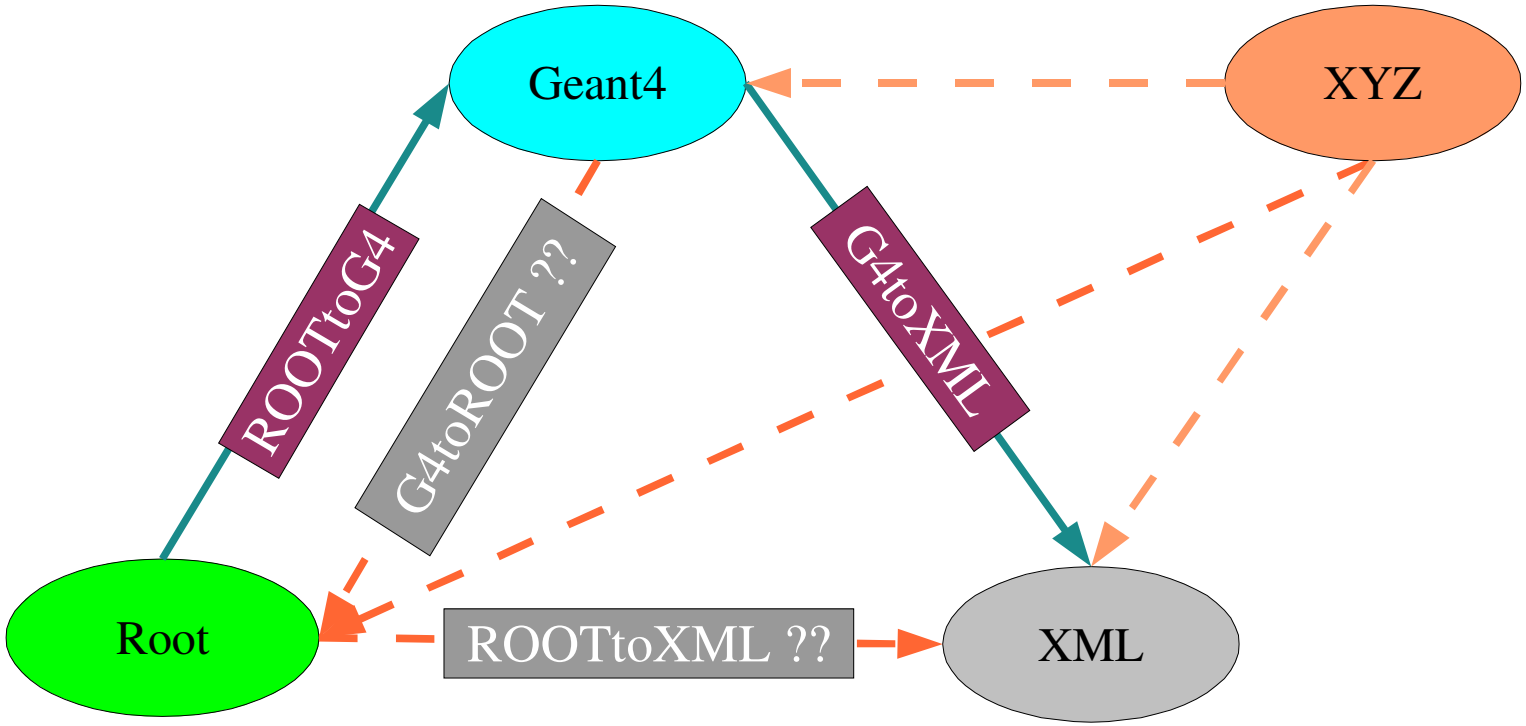
Outline

- Motivation
- Architecture
- Use of VGM
- Testing & Examples
- Present status

Motivation

- Tools for geometry conversion in Geant4 VMC
  - G4toXML – converter from Geant4 to XML, first AGDD then GDML format, gateway from Geant4 to GraXML
    - See poster "GraXML" (J. Hrivnac)
  - RoottoG4 – converter from Root geometry to Geant4, to support a new geometry engine for Virtual Monte Carlo
    - See presentation "The Virtual MonteCarlo, status and applications" (A. Gheata)
- Users' requests for other directions:
  - Geant4 -> Root
  - Root -> XML

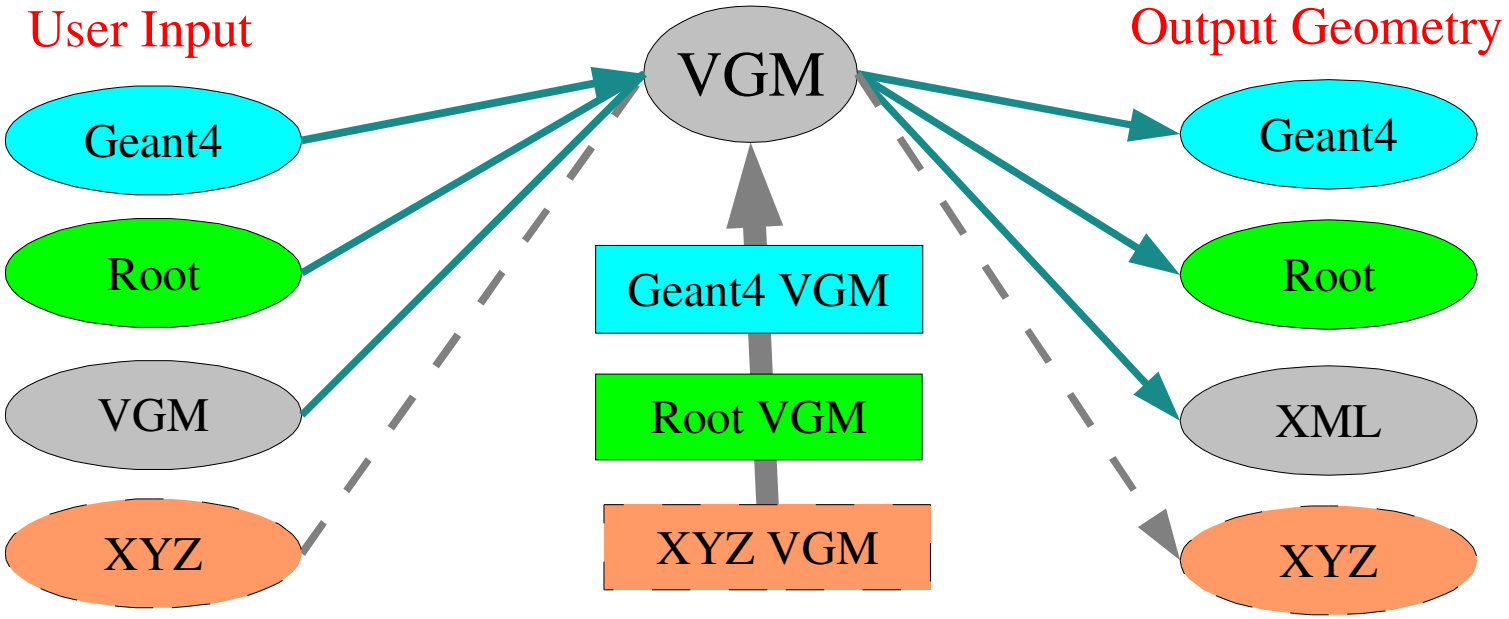
Previous Concept



- Can we do something simpler ?

New Concept

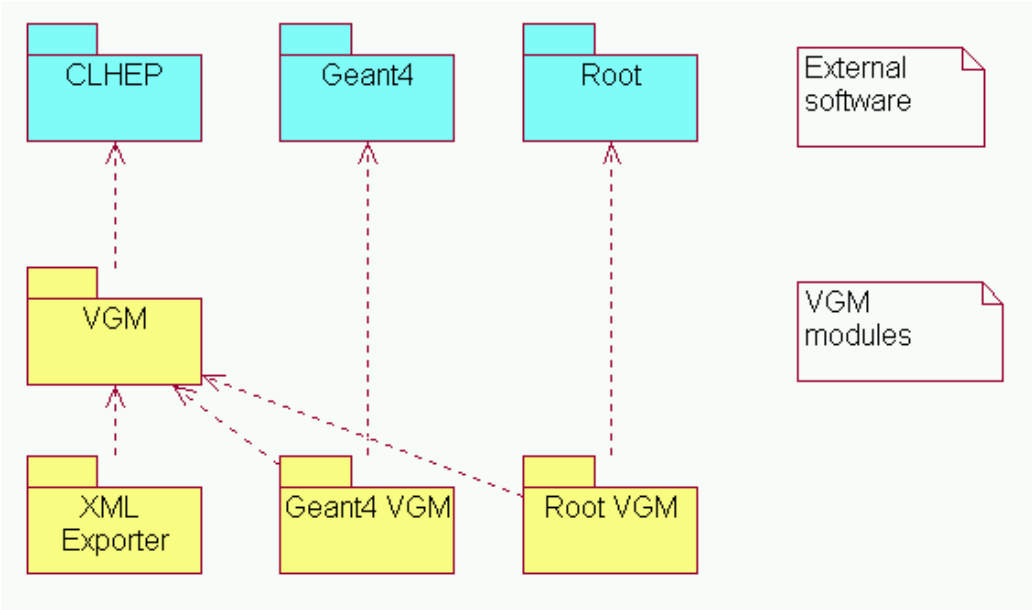
- Instead of adding other converters and multiplying the implementations, the abstract layer to geometry can be defined and the geometry models can be "mapped" to this generalized scheme



Architecture

VGM Components

- VGM
  - Interfaces to geometry objects
  - Factories for geometry construction, import and export
  - Common implementation
  - Use of CLHEP for 3D transformations
- VGM layer for specific geometry modules
  - Geant4 VGM, Root VGM
- XML exporter



Architecture

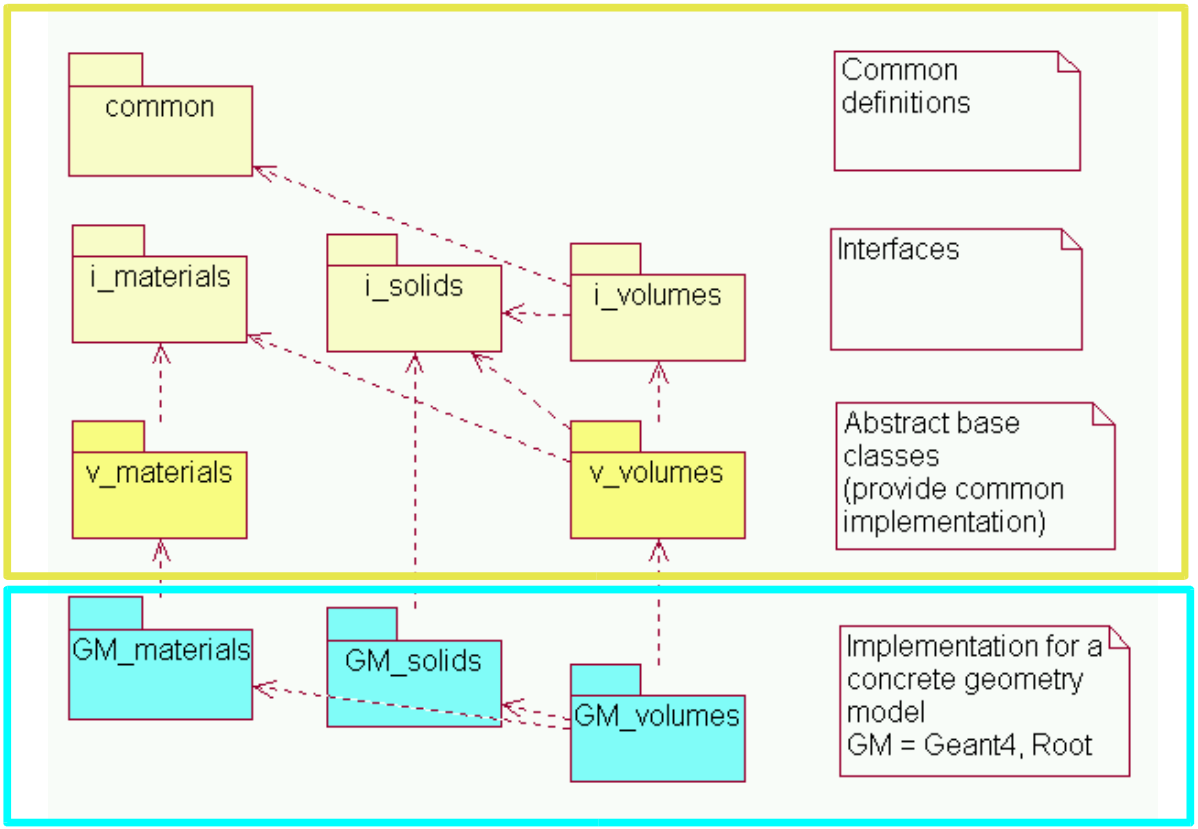
VGM Components (cont.)

- Lower level packaging

VGM

VGM Geant4

VGM Root



Architecture

VGM Interfaces

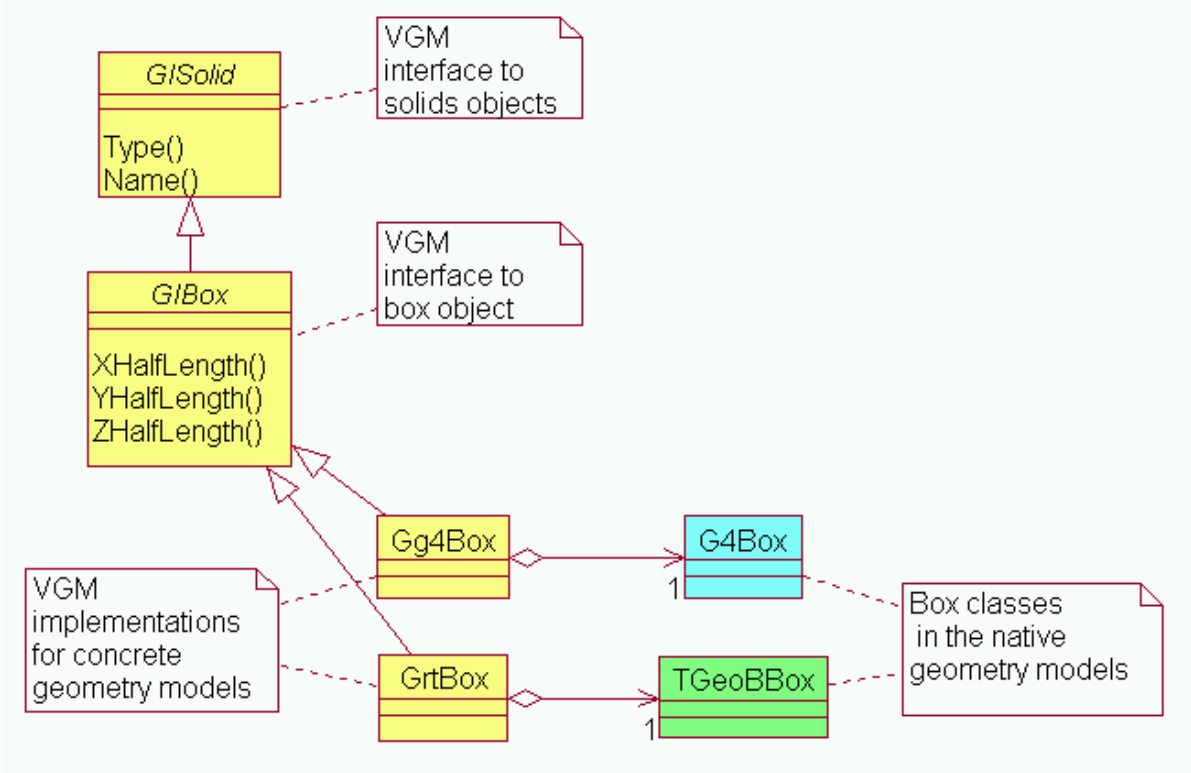
- Geometry objects:
  - Solid, Volume, Placement – hierarchical volume structure
  - Element, Material, Medium – material properties
- Some objects have more specifications
  - Solid - box, tube, cone, ...
  - Placement – simple placement, multiple placement
- The VGM defines an abstract interface for each geometry object or object specification

Architecture

VGM Interfaces (cont.)

- Solids:
  - Geant4: G4Box, G4Tubs, G4Cons, ... : G4VSolid
  - Root: TGeoBBox, TGeoTube, TGeoCone, ... : TGeoShape
  - VGM: GIBox, GITubs, GICons, ... : GISolid
- The implementation of the VGM interface has to be provided by the VGM layer specific for the concrete geometry model
  - Geant4 VGM: Gg4Box, Gg4Tubs, Gg4Cons, ...
  - Root VGM: GrtBox, GrtTubs, GrtCons, ...

VGM Interfaces
Solid (Box)



Architecture

VGM Interfaces (cont.)

Other geometry objects:

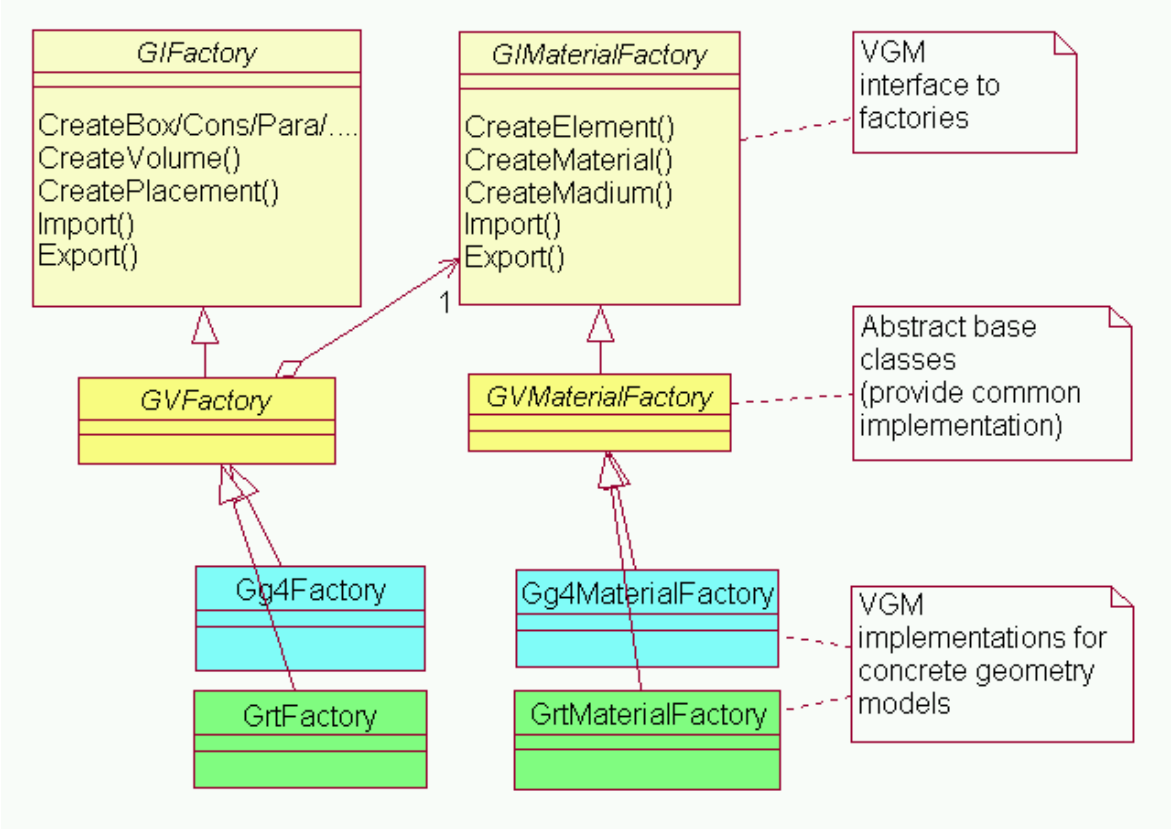
- Volumes
  - Geant4: G4LogicalVolume
  - Root: TGeoVolume
  - VGM: GIVolume
- Placements
  - Geant4: G4PVPlacement, G4PVReplica, ... : G4VPhysicalVolume
  - Root: TGeoNodeMatrix : TGeoNode
  - VGM: GIPlacement
- Materials
  - Geant4: G4Element, G4Material
  - Root: TGeoElement, TGeoMaterial, TGeoMedium
  - VGM: GIElement, GIMaterial, GIMedium

Architecture

VGM Interfaces (cont.)

- VGM abstract factories:
  - GIFactory, GIMaterialFactory
- Define methods for geometry construction, import and export
  - Using the interfaces to geometry objects
- Common implementation:
  - GVFactory, GVMaterialFactory – export function
- Geometry model specific implementations:
  - VGM Geant4: Gg4Factory, Gg4MaterialFactory
  - VGM Root: GrtFactory, GrtMaterialFactory

VGM Interfaces
Factories



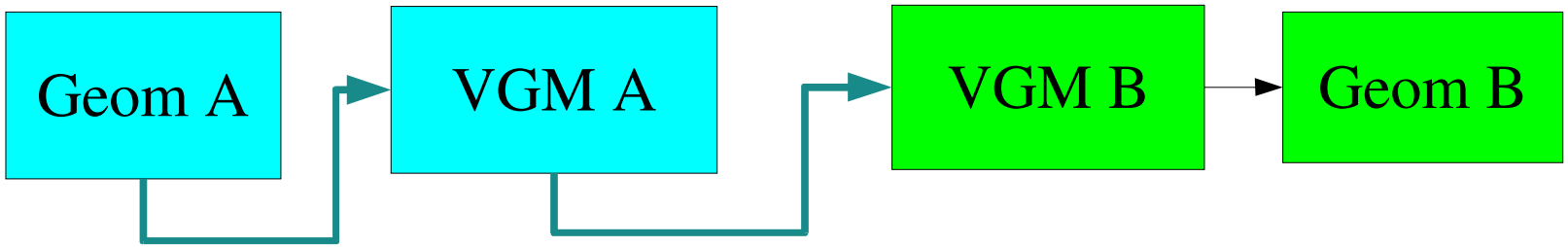
Use Of VGM

- Conversion between geometry models
  - Geant4 <-> Root
- Use of VGM factory
  - Possibility to define geometry via VGM – and so to decouple dependency of user code and a concrete geometry model
  - The same philosophy as Virtual Monte Carlo
- Export to XML
  - AGDD – Atlas Generic Detector Description in XML
  - GDML – Geometry Description Markup Language
    - Both presented at CHEP'01
    - AGDD now frozen in Atlas, GDML in LCG simulation project

Use of VGM

Geometry Conversions

- Converting the native geometry from one geometry model (A) to another (B):
  - I. Import the geometry in VGM using the VGM factory for this geometry model (A)
    - the native geometry objects are mapped to the VGM interfaces
  - II. Export it into the VGM factory for the other geometry model (B)



I. FactoryA->Import(A) II. FactoryA->Export(FactoryB)

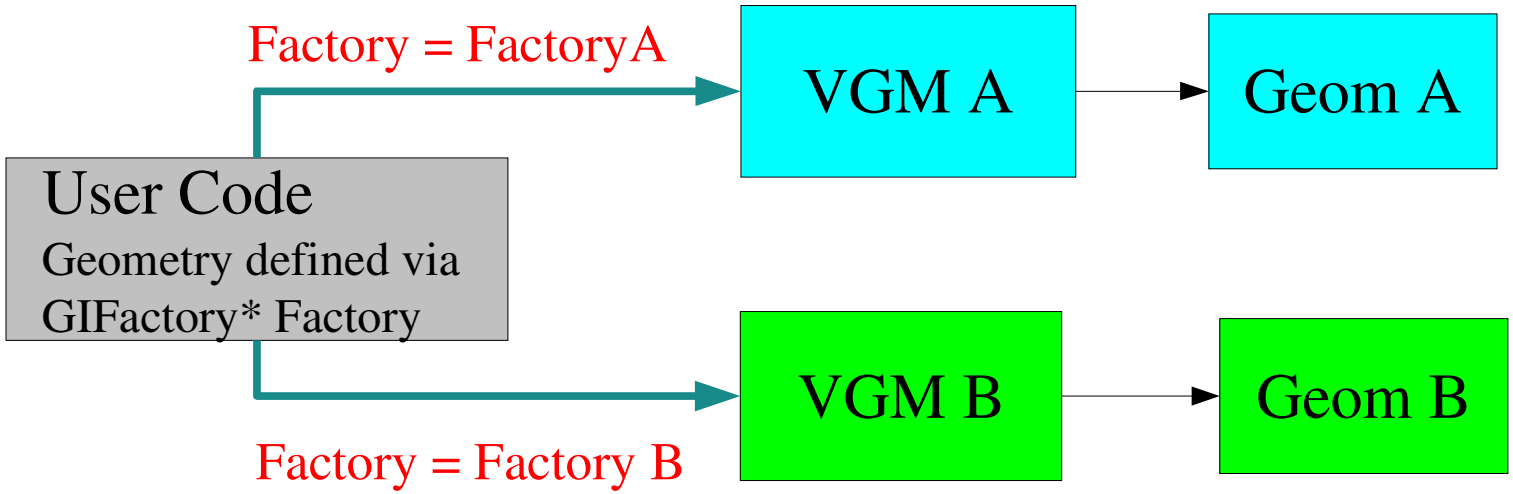
Geometry Conversions

Example: Geant4 -> Root

```
#include "Gg4Factory.h"
#include "GrtFactory.h"
#include "TGeoManager.h"

// Import Geant4 geometry to VGM
Gg4Factory g4Factory;
g4Factory.Import(physiWorld);
                    // where physiWorld is of G4VPhysicalVolume* type
// Export VGM geometry to Root
GrtFactory rtFactory;
g4Factory.Export(&rtFactory);
gGeoManager->CloseGeometry();
return rtFactory.World();
                    // returns Root top volume, of TGeoVolume* type
```

Geometry Construction Via VGM

- Geometry can be defined via VGM interfaces
  - Geometry definition is then independent from a concrete geometry model
- The geometry model will then be chosen with the instantiation of the concrete factory



Geometry Construction Via VGM

Example

```
MyDetectorConstruction::Construct ( GIFactory* factory )
{
   double wSize = 10*m;
   GISolid* worldS
       = factory->CreateBox("worldS", wSize, wSize, wSize);
                // create the world solid

   GIVolume* worldV
      = factory->CreateVolume("worldV", worldS, "Air");
                // create the world volume

   factory->CreatePlacement("world", 0, worldV, 0, 0, Hep3Vector());
               // place the world volume
}
```

Geometry Construction Via VGM

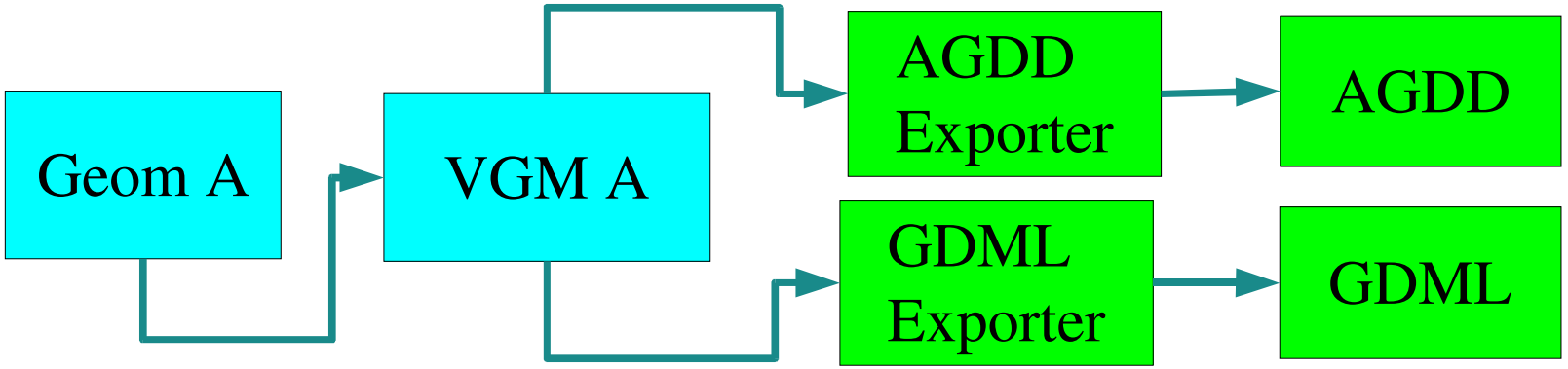
Example (cont.)

```
#include "Gg4Factory.h"
MyDetectorConstruction myDetectorConstruction;
Gg4Factory theFactory;
myDetectorConstruction->Construct(&theFactory);
      // Geant4 geometry is built

#include "GrtFactory.h"
MyDetectorConstruction myDetectorConstruction;
GrtFactory theFactory;
myDetectorConstruction->Construct(&theFactory);
      // Root geometry is built
```

Use of VGM

Export to XML

- XML exporter - generates XML files from the VGM volume tree
- Complying with the XML schema is embedded in the VGM XML exporter code itself, no external XML parser is used and so needed.



I. FactoryA->Import(A)

II. Exporter->GenerateXMLGeometry(FactoryA)

Export to XML
Example

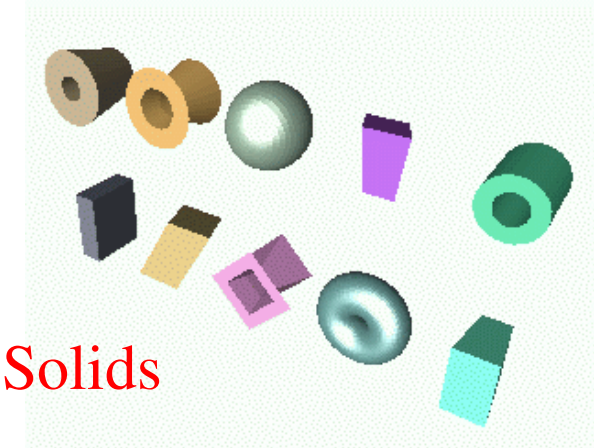
```
#include "GAGDDExporter.h"
GAGDDExporter xmlExporter1(&theFactory);
xmlExporter1.GenerateXMLGeometry();
    // Export geometry to AGDD

#include "GGDMLExporter.h"
GGDMLExporter xmlExporter2(&theFactory);
xmlExporter2.GenerateXMLGeometry();
    // Export geometry to GDML
```

Testing

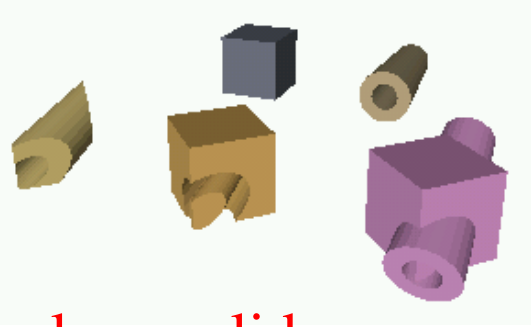
- The same simple geometry setups were defined via Geant4, Root, VGM to test different aspects of VGM:
  - Solids, Placements, Reflections, Boolean solids
- Test program:
  - vgm_test inputType inFactory outFactory outXML selectedTest [...]
    - inputType = Geant4, Root, VGM
    - inFactory, outFactory = Geant4, Root, (None)
    - outXML = AGDD, GDML, noXML
    - selectedTest = Solids, Placements, Reflections, BooleanSolids
- Test suite
  - All combinations of input/output/selectedTest included
  - Output from the test can be compared to the reference output

Testing

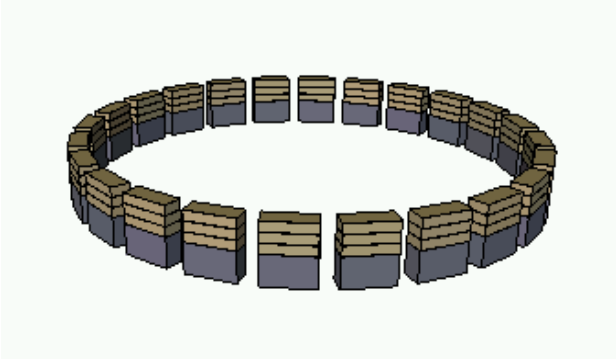
Geometry Setups



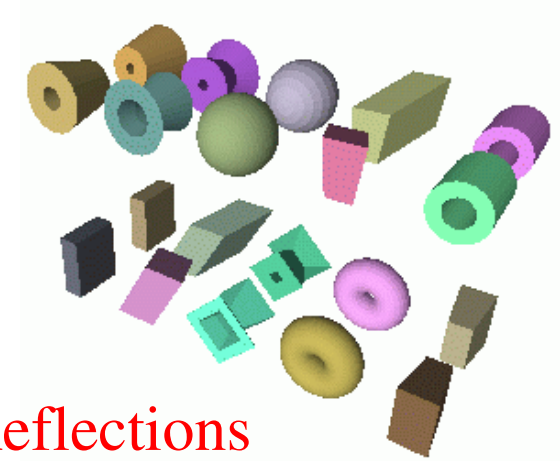
Solids



Boolean solids



Placements



Reflections

Examples

- Demonstrate use of VGM, much simpler than the extensive test program
- Example 1: Geant4 -> Root
  - Geant4 novice example N03 geometry converted to Root
- Example 2: Root -> Geant4
  - Root geometry (loaded from Root file generated in Example 1) converted to Geant4
- Example 3: Geant4 -> XML
  - Geant4 novice example N03 geometry exported to XML
- Example 4: Root -> XML
  - Root geometry defined in rootgeom.C tutorial macro exported to XML
- The examples 1, 2 require installation of both Geant4 and Root, the examples 3, 4 need just their geometry package

Present Status

- Supported features:
  - Most of solids – all CSG solids and polyhedra, polycone specific solids in Geant4 and their counterparts in Root
  - Boolean solids (Geant4), composite shapes (Root)
  - Reflected solids (Geant4), positioning with reflection (Root)
  - Multiple placements – replicas, divisions (Geant4), divisions (Root)
- Unsupported:
  - "Exotic" solids – solids that have no counterpart in the other geometry model
  - Parameterised volumes (Geant4)
  - Positions with "MANY" option (Root)
  - Boolean solids in XML exporter – on to do list

Conclusions

- The VGM introduces a general approach for conversion of geometries between specific geometry models
  - Geant4, Root TGeo, XML (AGDD, GDML)
  - This gives a possibility for a user of one specific package to use the tools supported by other packages:
    - Root TGeo => Virtual MC
    - XML (AGDD, GDML) => GraXML
- It also allows the user to define geometry independently from a specific geometry model
  - However this was not the main goal of the tool
- Available from
  - http://ivana.home.cern.ch/ivana/VGM.html