CHEP'07
VICTORIA, BC

International Conference on Computing
in High Energy and Nuclear Physics
2-7 Sept 2007 Victoria BC Canada

The Virtual Geometry Model

I. Hřivnáčová[1], B. Viren[2]

[1]IPN, Orsay; [2]BNL

International Conference On Computing in High Energy and Nuclear Physics,
Victoria BC, 2 - 7 September 2007

Outline

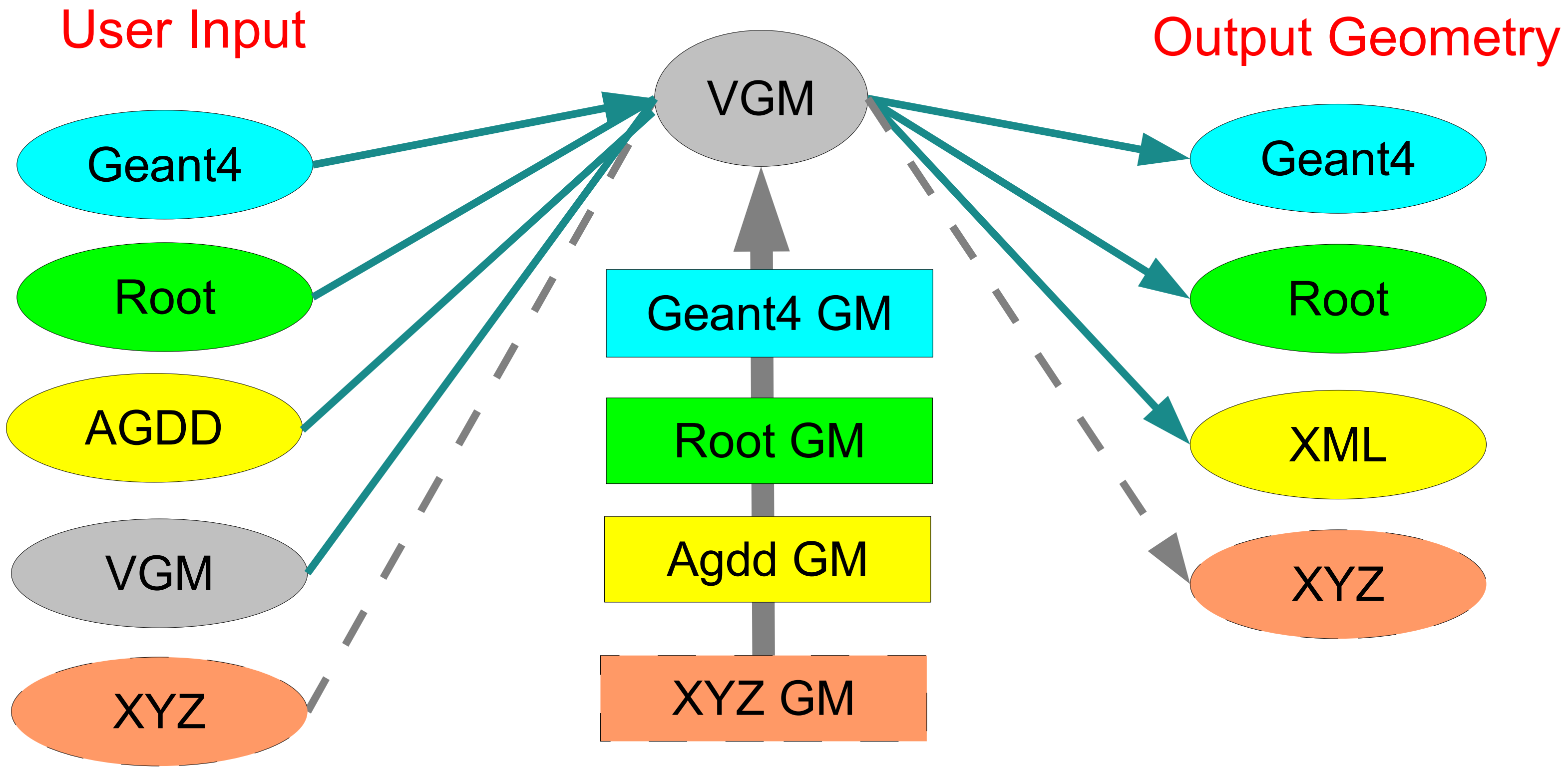
- Motivation
- Architecture
  - Packages, Interfaces
- Use of VGM
- Testing & Examples
- Present Status

Motivation

- Geometry modelers of interest
  - Geant4 – naturally utilized by many Geant4 users
  - Root TGeo – coming later, disconnected from any simulation tools
    - Used in the context of Virtual Monte Carlo
  - GDML – an application-independent geometry description based on XML
    - Used mainly for providing persistence for Geant4 geometry model
  - AGDD – another geometry description based on XML
    - AGDD abandoned in Atlas, adopted and further developed in STAR; recently adopted in Daya Bay; both for primary geometry implementation
    - Both AGDD and GDML inputs supported in GraXML tool
- Each geometry model provides a set of tools for geometry verification and visualization
  - The VGM development aims to provide the user the possibility to use all the available tools regardless their primary geometry format

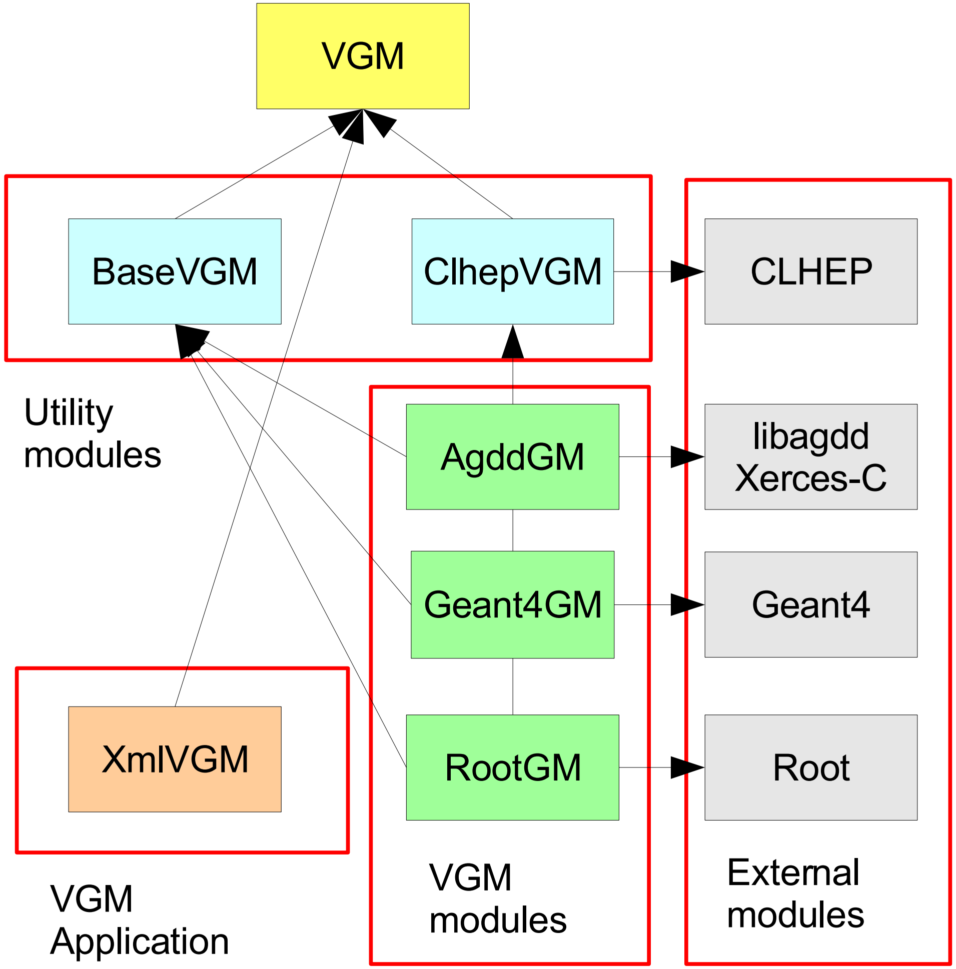
The VGM Concept

- The VGM defines an abstract layer to geometry and maps the geometry models to this generalized scheme
  - Geometry can be then accessed in a common way and converted from one geometry model to another



VGM Packages

- Since first release, the VGM packages have been cleaned up from unnecessary dependencies
  - The VGM package now includes only abstract interfaces and does not depend on any external package
  - All common implementation has been moved from the VGM package in BaseVGM
  - The CLHEP dependent code has been moved in the utility package ClhepVGM
- AgddGM, Geant4GM, RootGM
  - VGM implementation for the given geometry model
- XmlVGM - XML exporter



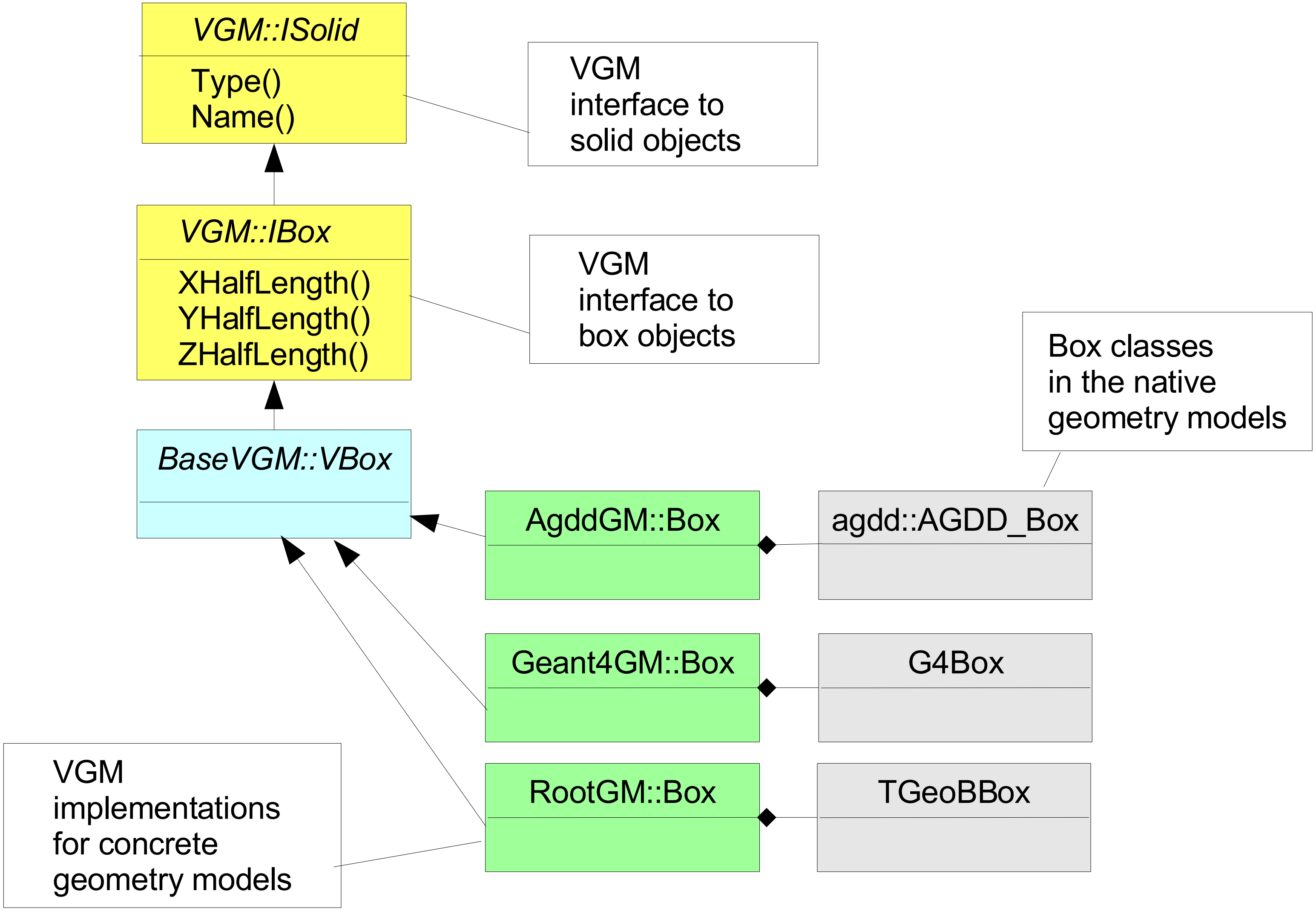
VGM Interfaces

Geometry Objects

- Geometry objects:
  - Solid, Volume, Placement – hierarchical volume structure
  - Isotope, 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
  - Solids: VGM::IBox, VGM::ITubs, VGM::ICons, … : GISolid
  - Volumes: VGM::IVolume
  - Placements: VGM::IPlacement
  - Material objects: VGM::IIsotope, VGM::IElement, VGM::IMaterial, VGM::IMedium

VGM Interfaces

Solid (Box)

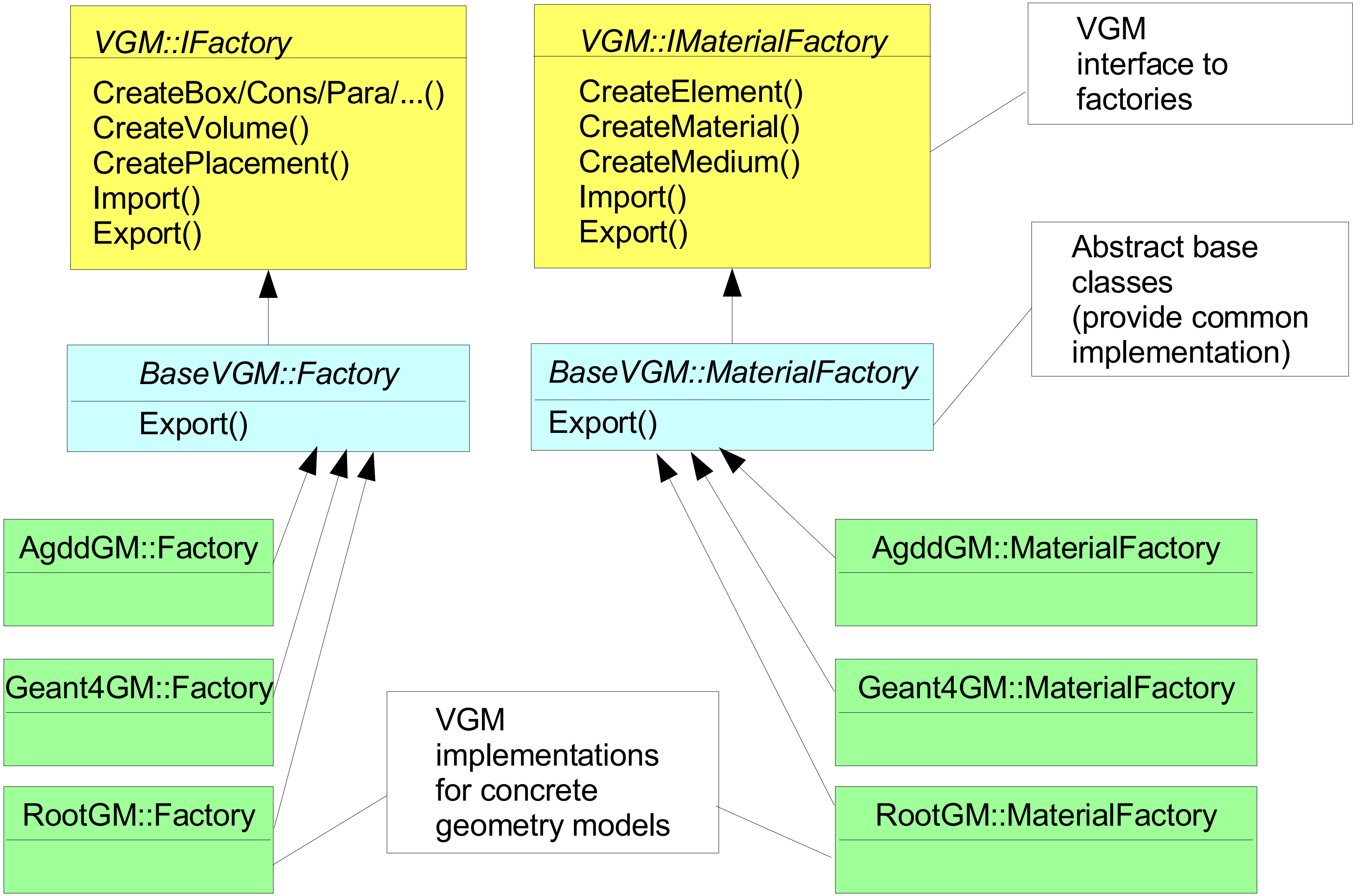

VGM::ISolid
Type()
Name()
VGM interface to solid objects
VGM::IBox
XHalfLength()
YHalfLength()
ZHalfLength()
VGM interface to box objects
Box classes in the native geometry models
BaseVGM::VBox
AgddGM::Box
agdd::AGDD_Box
Geant4GM::Box
G4Box
RootGM::Box
TGeoBBox
VGM implementations for concrete geometry models

VGM Interfaces

Factory Objects

- The Factory objects
  - Define methods for geometry construction, import and export
  - Using the interfaces to geometry objects
- Each VGM package for a specific geometry model has to implement two factories:
  - The factory for definition of the volumes tree
  - The factory for definition of materials

VGM Interfaces for Factories



Use Of VGM

- Conversion between geometry models
  - AGDD -> Geant4, Root, GDML
  - Geant4 -> Root, AGDD, GDML
  - Root -> Geant4, AGDD, GDML
    - GDML import not available
- Use of VGM factory & interfaces
  - Possibility to define geometry via VGM – and so to decouple dependency of the user code on a concrete geometry model
  - Possibility to define a geometry application (eg. for visualization) based on the VGM interfaces

Use Of VGM

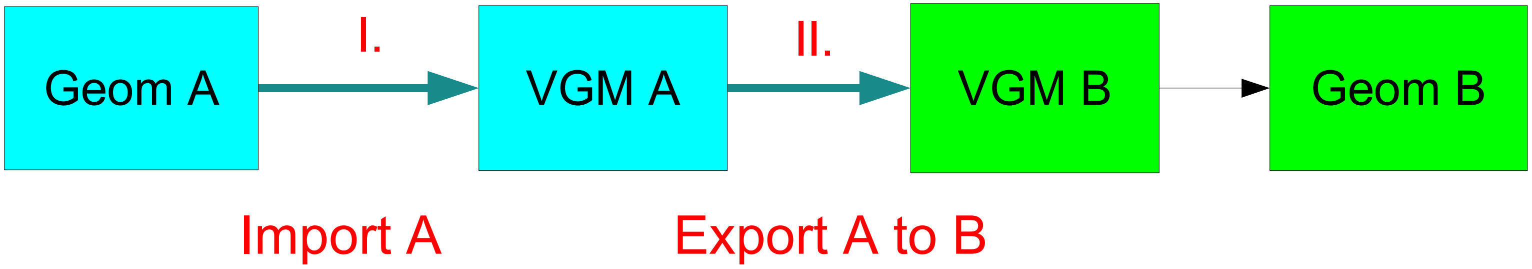
Geometry Conversions

- 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)



Geometry Conversions

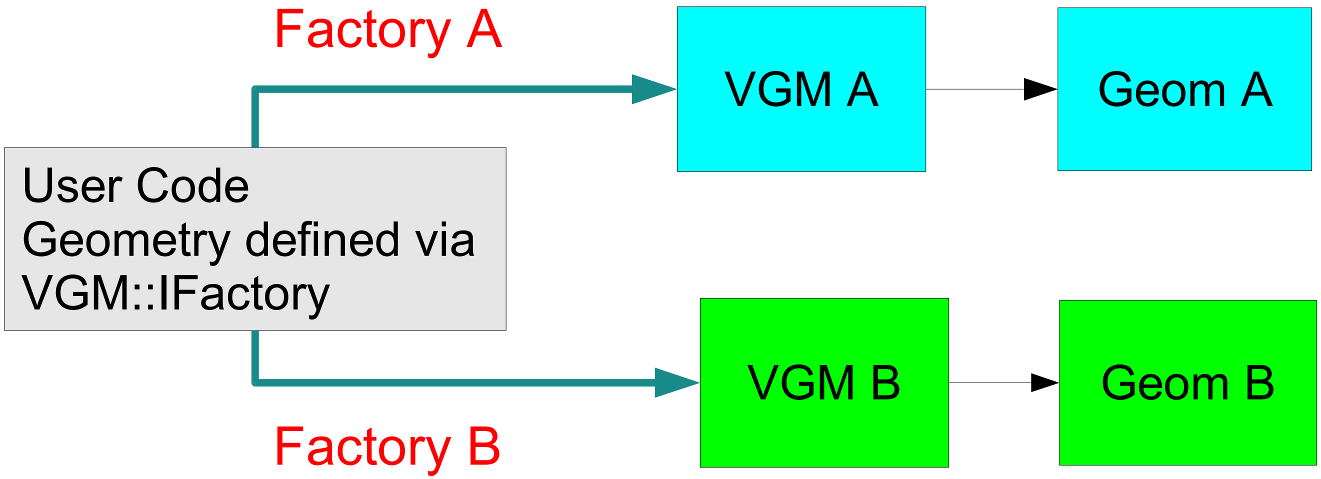
Example: Geant4 -> Root

```
#include "Geant4GM/volumes/Factory.h"
#include "RootGM/volumes/Factory.h"
#include "TGeoManager.h"

 // Import Geant4 geometry to VGM
 Geant4GM::Factory g4Factory;
 g4Factory.Import(physiWorld);
                  // where physiWorld is of G4VPhysicalVolume* type
 // Export VGM geometry to Root
 RootGM::Factory rtFactory;
 g4Factory.Export(&rtFactory);
 gGeoManager->CloseGeometry();
 return rtFactory.World();
                  // returns Root top volume, of TGeoNode* type
```

Use Of VGM

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



Use Of VGM

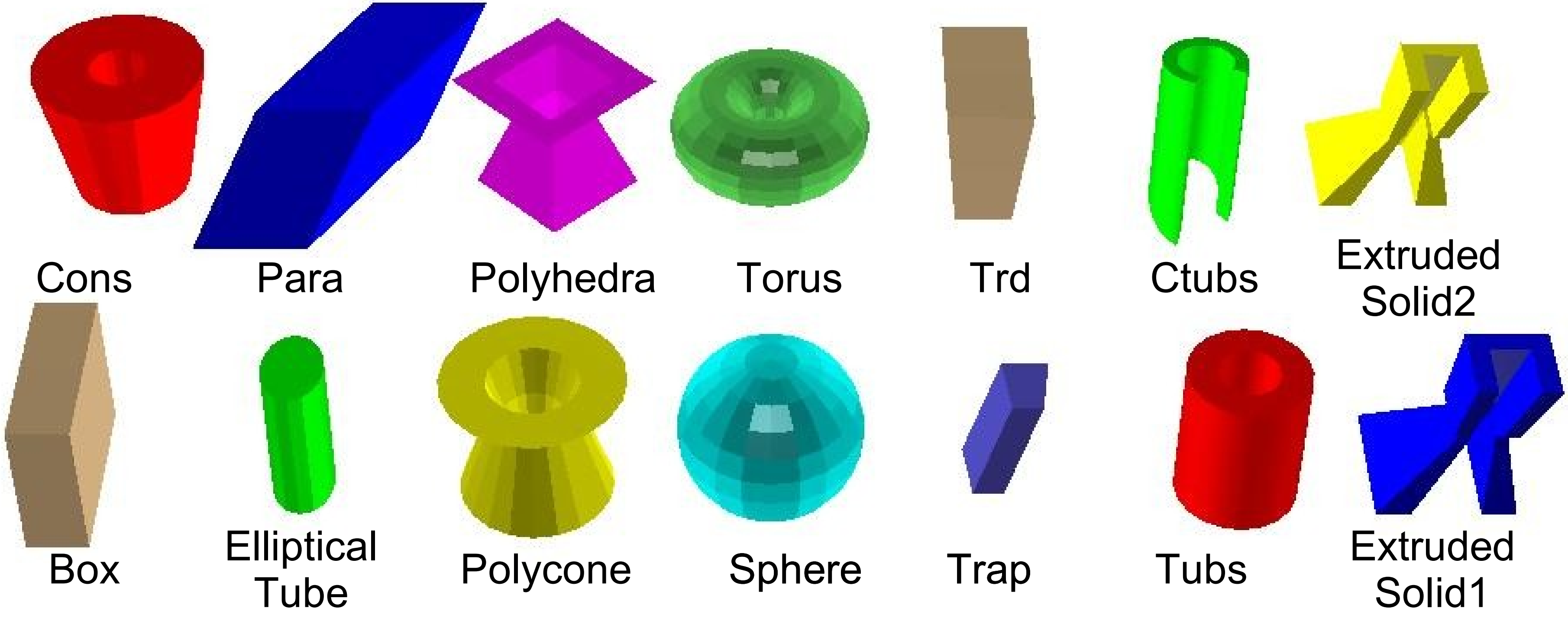
Export to XML - Example

```
#include "Geant4GM/volumes/Factory.h"
#include "XmlVGM/AGDDExporter.h"
#include "XmlVGM/GDMLExporter.h"

// Import Geant4 geometry to VGM
Geant4GM::Factory factory;
factory.Import(physiWorld);

// Export VGM geometry in AGDD
XmlVGM::AGDDExporter agddExporter(&factory);
agddExporter.GenerateXMLGeometry();

// Export VGM geometry in GDML
XmlVGM::GDMLExporter gdmlExporter(&factory);
gdmlExporter.GenerateXMLGeometry();
```

Testing

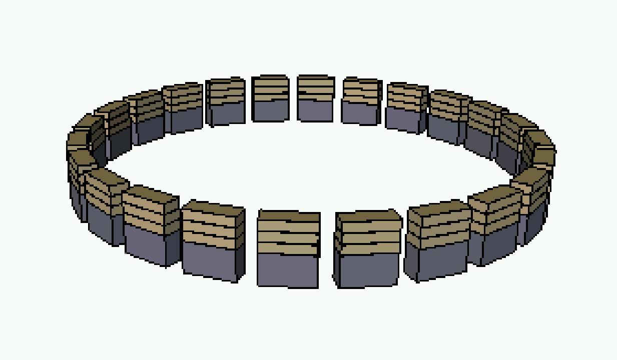
- The same simple geometry setups are defined via AGDD, Geant4, Root, VGM to test all different aspects of VGM:
  - Solids, Placements, Reflections, Boolean solids, Assemblies
  - Special – option to include user defined geometry

Solids

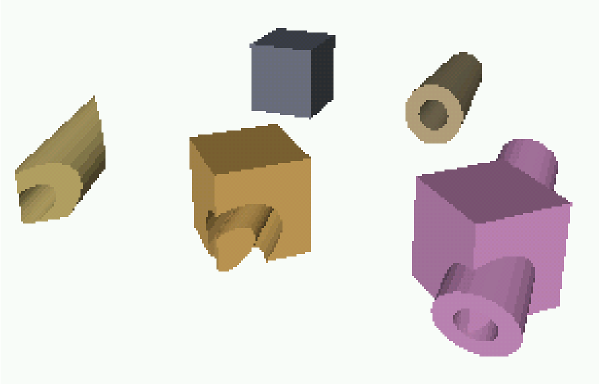


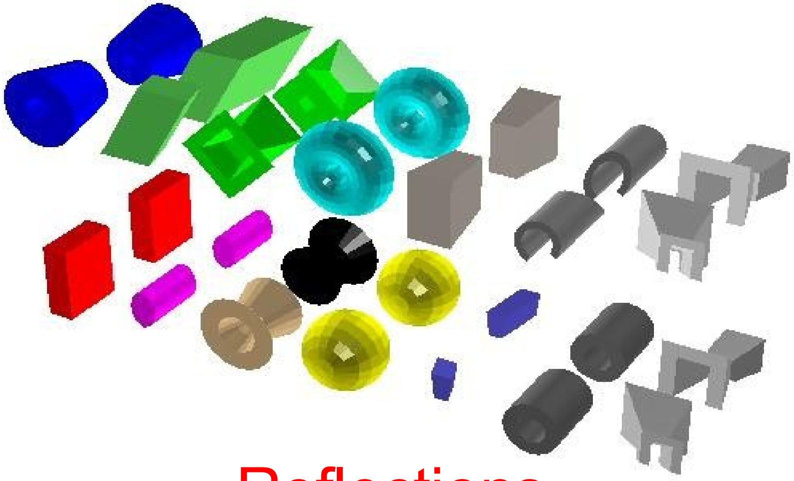
Testing (2)

Placements

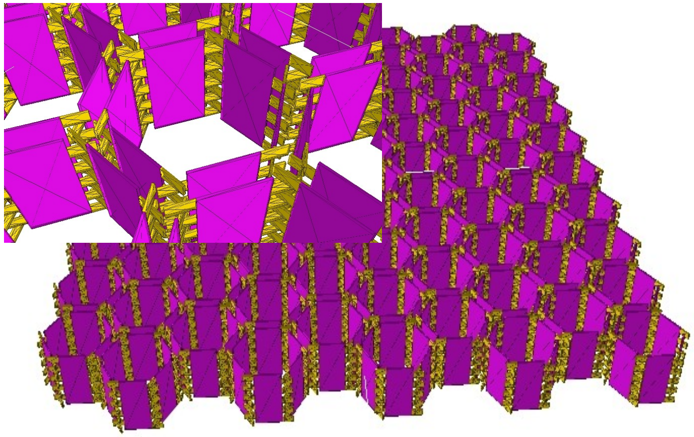


Boolean solids





Reflections



Assemblies

(from Root tutorial assembly.C)

Test Program

- Test program:
  - `vgm_test  inputType inFactory outFactory outXML selectedTest`
    `[run] [rootNavig] [...]`
    - `inputType` = AGDD, Geant4, Root, VGM
    - `inFactory, outFactory` = ADDD (only input), Geant4, Root, None (only output)
    - `outXML` = AGDD, GDML, noXML
    - `selectedTest` = Solids, Placements, Reflections, BooleanSolids, Assemblies, Special
    - `run` = option to run a test with Geant4 tracking
    - `rootNavig` = option to run a test with Geant4 tracking with G4Root navigation

Test Suites

- Test suites
  - All possible combinations of input/output/selectedTest included
  - Output from the test can be compared to the reference output
- Suites:
  - Suite1 – verbosity output from geometry conversion
  - Suite2 – XML export from all possible inputs
  - Suite3 – Geant4 tracking with Geant4 native navigation and G4Root navigation with verbose output
    - Using the GPS source of geantinos tuned for each geometry setup with a fixed random number seed
    - Very efficient for geometry debugging
- Special setup + run option
  - For verification of user geometry (comparing outputs from Geant4 native and G4Root navigation)

Examples

Demonstrate use of VGM, much simpler than the extensive test program

	Use case	Geometry source
E1	G4 -> Root	Geant4 novice example N03
E2	Root -> G4	Root file with geometry generated in E1
E3	G4 -> XML	Geant4 novice example N03
E4	Root -> XML	Root tutorial
E5	AGDD -> Root	AGDD test file

Building systems

- The VGM provides three independent building systems which make it easy to include it in various frameworks
- Building system based on GNU makefiles
  - Adapted makefiles from Geant4
- CMT
  - Thanks to Laurent Garnier, LAL, for porting VGM to CMT, maintenance of the CMT requirements files and testing on MacOS
- Autoconf
  - Introduced with adding AgddGM package

Supported Features

- Supported features
  - Large number of solids – all CSG solids and number of 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)
  - Assemblies (Root)
- Unsupported features
  - Some “Exotic” solids
  - Parameterised volumes (Geant4)
  - Positions with “MANY” option (Root)

Present Status of Packages

- Geant4 GM, Root GM
  - All VGM interfaces implemented
- AGDD GM
  - Implementation started by the end 2006
  - Implemented the interfaces to the geometry objects and the factory import function
  - On to do list: complete the factory create function (used in export)
  - Based on the AGDD v6 (decision by Daya Bay), while the AGDD exporter is based on AGDD v7
- XML exporters
  - GDML – All solids supported, missing reflections, multiple placements
  - AGDD – All features implemented, except for some solids not supported in AGDD

Conclusions

- The VGM introduces a general approach for access to geometries of specific geometry models and for their conversion:
  - Geant4, Root TGeo, AGDD, GDML
  - This gives a possibility for a user of one specific package to use the tools supported by other packages
  - It also provides a gateway for a geometry based application independent from a concrete geometry model
- The VGM is used in Geant4 VMC to support user geometry defined via TGeo with Geant4 native navigation
- It has been also used in G4Root validation
- Available from
  - http://ivana.home.cern.ch/ivana/VGM.html