Geant4 VMC

Overview & Present Status

I. Hrivnacova

IPN, Orsay

VMC Workshop, CERN

29-30 November 2004

Introduction

- Geant4 VMC was presented many times to ALICE during its development, it was presented several times to Geant4 collaboration, however it was not yet presented to new VMC users
  - This presentation should fill this gap now
- The talk will not concentrate on the design or implementation but rather on pointing to
  - Differences between Geant3 VMC and Geant4 VMC
  - Differences between Geant4 VMC and Geant4
- It will also give guidelines and recommendations for writing VMC application reflecting the current status of the Geant4 VMC implementation

Outline

- Introduction
- Overview & Present status
  - Geometry definition, Physics list, Event generator
  - User actions – where VMC is different from Geant4
  - Run configuration, UIs, Visualization
  - Geometry Browser, Geometry XML exporter
  - Categories overview
- Performance test
- Geant4 VMC extensions
  - Converters: RootToG4, G4ToXML
  - Virtual Geometry Model
- Conclusions

Introduction

Geant4 Application

- Geant4 application consists of user implementation of Geant4 user abstract base classes:
  - Mandatory:
    - Detector construction, Physics list, Primary generator
  - Optional:
    - Run, Event, Tracking, Stacking, Stepping actions
- Analogy with VMC Application
  - The difference is that while VMC collects all interfaces in a single class, Geant4 defines a set of classes for this

Introduction
Geant4 VMC

- Geant4 VMC
  - Implements the TVirtualMC interface for Geant4
  - Implements Geant4 user mandatory/action classes
    - Represents an “empty” Geant4 application which is filled in by a user VMC application
- Implementation is based on the Geant4 framework
  - Some classes inherit from G4 classes
  - Use of Geant4 typedefs
  - CINT dictionaries only for a few selected classes which we want to access from ROOT UI

Outline

- Introduction
- Overview & Present status
  - Geometry definition, Physics list, Event generator
  - User actions – where VMC is different from Geant4
  - Run configuration, UIs, Visualization
  - Geometry Browser, Geometry XML exporter
  - Categories overview
- Performance test
- Geant4 VMC extensions
  - Convertors: RootToG4, G4ToXML
  - Virtual Geometry Model
- Conclusions

Geometry (1)
Geometry Definition via VMC

- Currently 2 possibilities of geometry definition
- Old way – via VMC functions:
  - Geometry is built by calls to TvirtualMC
  - VMC functions for geometry constructions are G3-like
  - Implemented with use of G3toG4 tool (in Geant4)
  - Suitable for users starting from existing G3 application

Geometry (2)

Geometry Definition via TGeo

- New way – using TGeo geometry modeller in ROOT
  - More user friendly than the old VMC interface
  - Geometry verification tool, visualization and persistency
  - Recommended for new users starting from scratch
  - Implemented via RootToG4 converter
    - Provided within Geant4 VMC
    - Using the converter - ROOT geometry is converted in memory in G4 geometry objects and then the standard G4 geometry is used in simulation
  - In plan: direct use of TGeo with Geant4 via a specific implementation of G4Navigator
    - This direction will be discussed tomorrow

Geometry (3)
Critical Issues

- Be careful
  - If using "MANY"
    - Geometry defined via VMC geometry – the overlaps have to be resolved manually by specifying the overlapping volumes using TVirtualMC::Gsbool() function
      - Limit use of MANY as much as possible
    - Geometry defined via TGeo – the MANY option is neglected by RootToG4 converter => Geant4 geometry will not be correct
      - Use composite shapes instead of MANY
  -

Geometry (4)
Critical Issues

- Be careful if using reflections
  - In both ways, the volumes placed with a transformation including reflection are in Geant4 handled in a special way (via G4ReflectionFactory)
  - Do not use reflection if a simple rotation can do the same
    - (Eg. reflecting a tube along Z-axis can be done via simple rotation)
  - Avoid reflections in definition of composite shapes
    - Reflecting of Boolean solids is not yet supported in Geant4

Geant4 Physics Lists

- Geant4 has no "default physics"
- Physics List = user's selection of particles and physics processes
- Physics list samples provided by Geant4
  - Electromagnetic physics list
    - 1 physics list for standard EM physics
  - Educated Guess Hadronic Physics Lists
    - ~ 20 physics lists
    - the entry points to the various major use-cases of geant4 hadronic physics
    - http://www.geant4.com/hadronics/GHAD/HomePage

Geant4 VMC Physics List

- Geant4 VMC has defined its default physics list
- Implemented as modular physics list (derived from G4VModularPhysicsList), composed of:
  - Geant4 novice examples physics constructors:
    - EM (standard), Muon, Hadron, Ion, Optical photon physics
    - With modifications (additions) for VMC:
      - G4 physics processes mapped to
        - VMC process codes (TMCProcess enum values)
        - VMC process controls ("PAIR", "COMP", ...)
  - Special constructors for controlling physics via VMC:
    - SpecialControls, SpecialCuts
    - Not activated by default

Geant4 VMC Physics List
Process controls

- VMC – process controls in G3 style:
  - TVirtualMC::SetProcess(flagName, flagValue),
  - TVirtualMC::Gstpar(medId, flagName, flagValue),
    - FlagName = "PAIR", "COMP", "BREM", ...
- Geant4 – process activation is fully controlled by user physics list
- Geant4 VMC
  - Not all G3 flags are applicable to Geant4
  - Correspondence between VMC flags and G4 physics processes is commented in TG4G3Control enum
  - Applied by activating/inactivating the G4 physics processes corresponding to the given flag and a "special control" process associated with relevant particles

Geant4 VMC Physics List
Cuts

- VMC – cuts in G3 style (cuts in kinetic energy, per medium)
  - TVirtualMC::SetCut(cutName, flagValue),
  - TVirtualMC::Gstpar(medId, cutName, cutValue),
    - CutName = "CUTGAM", "CUTELE", "CUTNEU", ...
- Geant4 - cuts in range per particle, cuts per regions (recent)
- Geant4 VMC
  - All G3 cuts are applicable to Geant4
  - Implented via a special cuts process and user limits
  - Correspondence between VMC cuts and G4 particles/physics processes concerned is commented in TG4G3Cut enum
  - Cuts applied as tracking cuts, not threshold
  - Not by default, activated by invoking the command /mcPhysics/setSpecialCuts true

Geant4 VMC Physics List

Warning at run time:

```
###
You are using the Geant4 VMC hadronics physics
equivalent to ExN04HadronPhysics from Geant4
(version 6.0)
Note that this hadronic physics list is not optimized
for any particular usage. If you wish to have a starting
point tailored for a particular area of work, please use
one of the available physics lists by use-case.
More information can also be found from the Geant4
HyperNews.
###
```

… saying that Geant4 VMC physics list is probably not the best choice of Geant4 physics for your use case

User Defined Physics List (1)

- Geant4 VMC can be configured with a user own physics list
- Use own physics list or "educated guess" physics list especially when using hadronic physics
- To get the VMC functions associated with the physics list working, do:
  - For VMC cuts – register TG4PhysicsConstructorSpecialCuts
  - For VMC process controls – map physics processes to VMC flags; register TG4PhysicsConstructorSpecialControls
  - For VMC process codes - map physics processes to these
- Activating VMC cuts does not require to modify the physics list, however the other functions do

User Defined Physics List (2)

- If using a standard Geant4 physics list with no modifications specific for VMC
  - Avoid use of VMC process codes in your VMC application
  - Make sure that the composition of physics processes is coherent with global VMC process controls defined via TVirtualMC::SetProcess()
  - Avoid use of VMC process controls per tracking medium (via TVirtualMC::Gstpar)

Geant4 User Actions

Geant4	VMC
BeginOfRunAction()	0
BeginOfEventAction()	BeginEvent()
0	BeginPrimary()
PreUserTrackAction()	PreTrack()
~~	~~~~
PostUserTrackingAction()	PostTrack()
0	FinishPrimary()
EndOfEventAction()	FinishEvent
EndOfRunAction	0

- Most of Geant4 user action classes are implemented just via calls to the VMC Application

Geant4 User Actions

Stacking Of Particles (1)

- Is in VMC defined by user VMC stack, however it is used differently in Geant4 VMC than in Geant3 VMC:
  - G3: stacking mechanism fully defined by user
    - Both primary and secondary particles are popped from the stack as they are provided by TVirtualMCStack::PopNextTrack()
  - G4: stacking mechanism disconnected from VMC stack
    - by default - default Geant4 stacking mechanism
      - Primary particles are popped using TVirtualMCStack::PopPrimaryForTracking()
      - Secondary particles are handled by G4kernel and VMC stack only monitors this stacking
    - optional - special stacking mechanism implemented for ALICE (TG4SpecialStackingAction)

Geant4 User Actions

Stacking Of Particles (2)

- Stack classes in VMC examples
  - Provide the same stacking mechanism for both G3, G4
- Use one of the stacks from VMC examples rather than implementing a new own one
- Limitation: The particles added in the VMC stack in other functions then TVirtualMCApplication::GeneratePrimaries() are not passed to G4 tracking
  - ALICE is concerned with this

Geant4 User Actions
Stepping

Geant4:

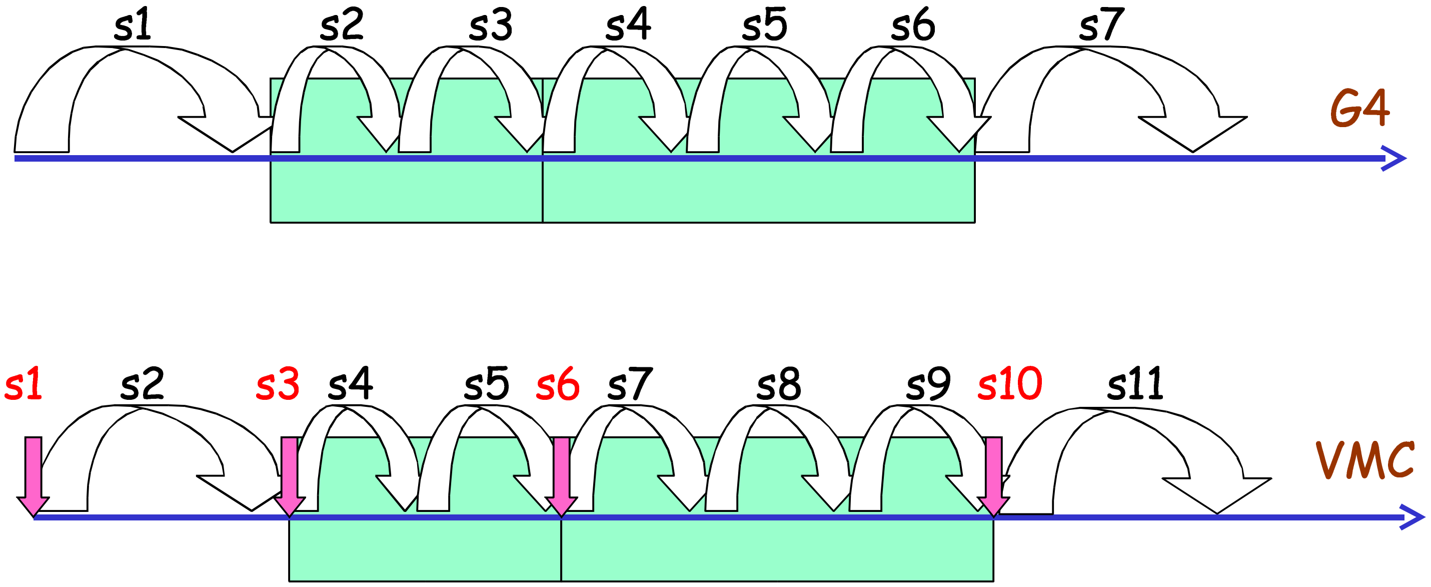
- UserSteppingAction:: UserSteppingAction()
  - Called at each step,
  - Not dependent on the current volume
- SensitiveDetector:: ProcessHits()
  - Called if the step is processed in the volume associated with this SD

VMC

- VMC application Stepping() function
- To be compatible with Geant3:
  - Called at each G4 step
  - Plus additional calls:
    - At starting a new track
    - At geometry boundary
- Volume selection in user code via volume ID

Geant4 User Actions

Stepping



Run Configuration

- TG4VRunConfiguration
  - Abstract class
  - Defines composition of all Geant4 user mandatory and user action classes
- TG4RunConfiguration
  - Concrete class
  - Default Geant4 VMC configuration
    - Instantiates Geant4 VMC classes that implement Geant4 user mandatory and user action classes
    - TG4DetConstruction, TG4ModularPhysicsList, ... etc.

User Run Configuration

- User can define his own RunConfiguration derived from TG4VRunConfiguration
  - However replacement of a TG4* class with a user one can disconnect some VMC function(s)
  - Recommended only for including user own physics list
  - In plan – to make possible to include user detector construction (will be discussed later in this talk)

Visualization

- VMC functions for drawing:
  - Interfaces from Geant3 (now outdated)
  - In TVirtualMC.h:

```
// functions for drawing
// to be removed with complete move to TGeo
```

- Geant4 visualization
  - Provides more functionality and interactivity (comparing to VMC)
  - Necessary for geometry verification
  - Even in the context of using TGeo with Geant4 VMC
- Geant4 graphics drivers in Geant4 VMC
  - OPENGLX, OPENGLXM – supported by default
  - Other drivers can be included after modifying Makefile following the comment lines in it

(G)UIs

- VMC UI = Root UI
  - VMC application is called from the Root session
  - Access to VMC and ROOT objects, ROOT browser
  - However Geant4 objects are not accessible from the Root UI (Geant4 classes are not processed by CINT)
- User can switch between Root UI and Geant4 UI
- Geant4 UI:
  - Access to Geant4 via Geant4 built-in commands
  - Geant4 VMC commands – start with “mc” prefix
- Possibility to process a foreign macro/command in both UIs
  - Eg. To open ROOT browser from G4 UI:

    /mcControl/rootCmd TBrowser b;

(G)UIs

Geant4 VMC Geometry Browser

- Visualization of the geometry volumes tree, drawing volumes, browsing material parameters, user limits, applied cuts
  - Analogy to Geant3 VMC Geometry browser
- Browsing of the geometry volumes tree and interactive drawing volumes now available via Geant4 or external tools
- Browsing VMC parameters (user limits, cuts) only here
- To open the browser from Root session:
  - root[0] new TG4GeometryGUI();
- By D. Adamova, NPI Rez

Geant4 VMC

Categories

Global	Common definitions
Geometry	Geometry definition
Digits+hits	Volume IDs, Sensitive detectors
Physics	Physics list, Cuts, Controls, External decay
Event	Stepping, Tracking, Stacking, Event actions
Run	Primary particles, Run configuration, UIs
Visualization	Visualization settings
Interfaces	Geometry GUI

Outline

- Introduction
- Overview & Present status
  - Geometry definition, Physics list, Event generator
  - User actions - where VMC is different from Geant4
  - Run configuration, UIs, Visualization
  - Geometry Browser, Geometry XML exporter
  - Categories overview
- **Performance test**
- Geant4 VMC extensions
  - Convertors: RootToG4, G4ToXML
  - Virtual Geometry Model
- Conclusions

Performance test

- Based on the Geant4 novice N03 example which was rewritten in VMC framework (E03 example in VMC)
  - Enables to compare performance of G3, G4 VMC and G4 (native) on a simple geometry & physics problem
- Test:
  - 50 events
  - 1000 primary e-, 50 MeV
  - 10 layers: 1cm Lead + 0.5 cm Liquid Argon
  - Default material (world): vacuum changed to air
  - Uniform magnetic field 0.2 tesla
  - EM physics
    - G3: LOSS=1, DRAY=1, HADR=0
    - G4: Geant4 VMC EM Physics Constructor
  - Cuts in G3 set to correspond to a uniform cut 1 mm in G4

Performance test

- Geant4 VMC configurations:
  - Special stepping of zero length for new track and on boundary on/off
  - Saving secondaries to MC stack on/off
    - If comparing with native G4 simulation be aware that G4 does not make secondaries persistent

C1 fastest	No special steps of zero length Not saving secondaries in MC stack
C2	With special steps of zero length Not saving secondaries in MC stack
C3 slowest	With special steps of zero length With saving secondaries in MC stack

Performance test

MC	Time [s]
G3 VMC	117 (0.52)
G4 native (v5.0)	227 (1.00)
G4 VMC c1	245 (1.08)
G4 VMC c2	272 (1.20)
G4 VMC c3	349 (1.54)

Outline

- Introduction
- Overview & Present status
  - Geometry definition, Physics list, Event generator
  - User actions - where VMC is different from Geant4
  - Run configuration, UIs, Visualization
  - Geometry Browser, Geometry XML exporter
  - Categories overview
- Performance test
- Geant4 VMC extensions
  - Convertors: RootToG4, G4ToXML
  - Virtual Geometry Model
- Conclusions

Geant4 VMC Extensions
Converters

- Provided within geant4_vmc
- Module convertors with packages
  - roottog4
  - g4toxml
- Both modules
  - Independent from geant4_vmc, can be used standalone
  - Used in geant4_vmc
    - RootToG4 – needed to support TGeo geometry input
    - G4toXML – only adds functionality

Geant4 VMC Extensions
Geometry XML Converter

- Enables to export Geant4 geometry to XML files
- XML formats:
  - AGDD – Atlas Generic Detector Description in XML
  - GDML – Geometry Description Markup Language
    - AGDD now frozen in Atlas
    - GDML in the LCG simulation project
- The motivation for this work was use of the GraXML tool
  - Browsing and visualization of geometry
  - Advanced graphics
  - More on GraXML: http://home.cern.ch/hrivnac/Activities/Packages/GraXML

Geant4 VMC Extensions
Virtual Geometry Model (1)

- Generalization of the converters in Geant4 VMC
- Presented at CHEP'04
- Defines an abstract layer to geometry objects and maps the geometry models (Geant4, Root) to this generalized scheme
  - Geometry objects from mapped models can be then handled in the same way
- Conversions:
  - RootToG4, G4ToRoot
  - XML export: both Root and Geant4 to AGDD and GDML
- Motivation for this work
  - Interest of people in possibility to move to VMC starting from Geant4 geometry

Geant4 VMC Extensions

Virtual Geometry Model (2)

- More on VGM:
  - CHEP'04 talk
  - http://home.cern.ch/ivana/VGM.html
    - Source code, installation instructions, examples
- Plans:
  - Improvements of GDML exporter
    - to upgrade to GDML 2.0, add support for Boolean shapes
  - Design improvements

Geant4 VMC Extensions

Geometry Definition via Geant4 ?

- Geometry defined via Geant4?
  - Not in the VMC project main line
  - However for some projects/experiments this may be a key issue in adopting the VMC line
- Virtual Geometry Model
  - Makes possible to pass Geant4 native geometry to VMC by converting it to Root geometry
- Conversion G4->ROOT
  - Desirable for G3, and Fluka VMCs
  - But not really for G4 VMC
  - Support for Geant4 native input in Geant4 VMC will not require much effort, in plan

Geant4 VMC Extensions
Plans

- To replace converters in Geant4 VMC with VGM
  - Maintaining both would be duplication of work
- VGM - external module to Geant4 VMC
  - Optional download
  - Optional compilation from Geant4 VMC
- VGM distribution
  - Currently via own Web page
  - Is there an interest in distribution via Root CVS/Web site ?

Outline

- Introduction
- Overview & Present status
  - Geometry definition, Physics list, Event generator
  - User actions - where VMC is different from Geant4
  - Run configuration, UIs, Visualization
  - Geometry Browser, Geometry XML exporter
  - Categories overview
- Performance test
- Geant4 VMC extensions
  - Converters: RootToG4, G4ToXML
  - Virtual Geometry Model
- **Conclusions**

Conclusions

- Geant4 VMC in production since 2002
  - Maintenance - updating with Geant4 releases, bug fixing
  - Continuos implementation of new user requirements
  - Geant4 VMC Web page
    - http://root.cern.ch/root/vmc/Geant4VMC.html
    - Includes source code documentation generated by Doxygen
- Though VMC provides a common denominator for all implemented MCs
  - User has a possibility to tune "their" Geant4 via standard Geant4 commands as well as via their own G4 based code overriding the defaults in Geant4 VMC
  - However these possibilities should be used with moderation