CHEP'07 VICTORIA, BC
International Conference on Computing
in High Energy and Nuclear Physics
2-7 Sept 2007 Victoria BC Canada

The Geant4 Virtual Monte Carlo

I. Hřivnáčová

IPN, Orsay

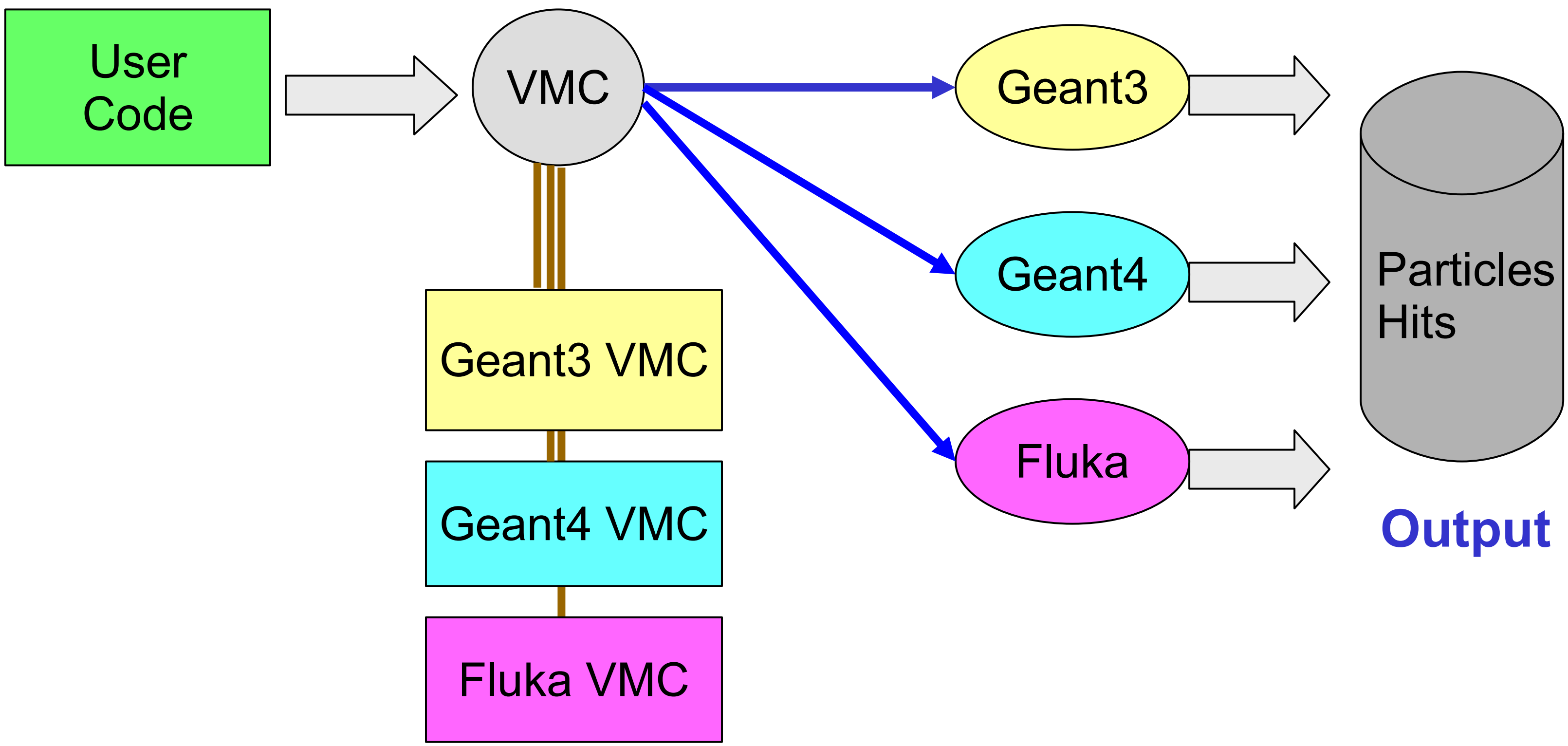
International Conference On Computing in High Energy and Nuclear Physics,
Victoria BC, 2 - 7 September 2007

Outline

- The VMC concept
- Geant4 VMC as a Geant4 application
  - Geometry definition & navigation
  - Selection of physics
  - Primary generator, stacking of particles
- Geant4 VMC specific features
  - User run configuration
  - User interface
- Tests

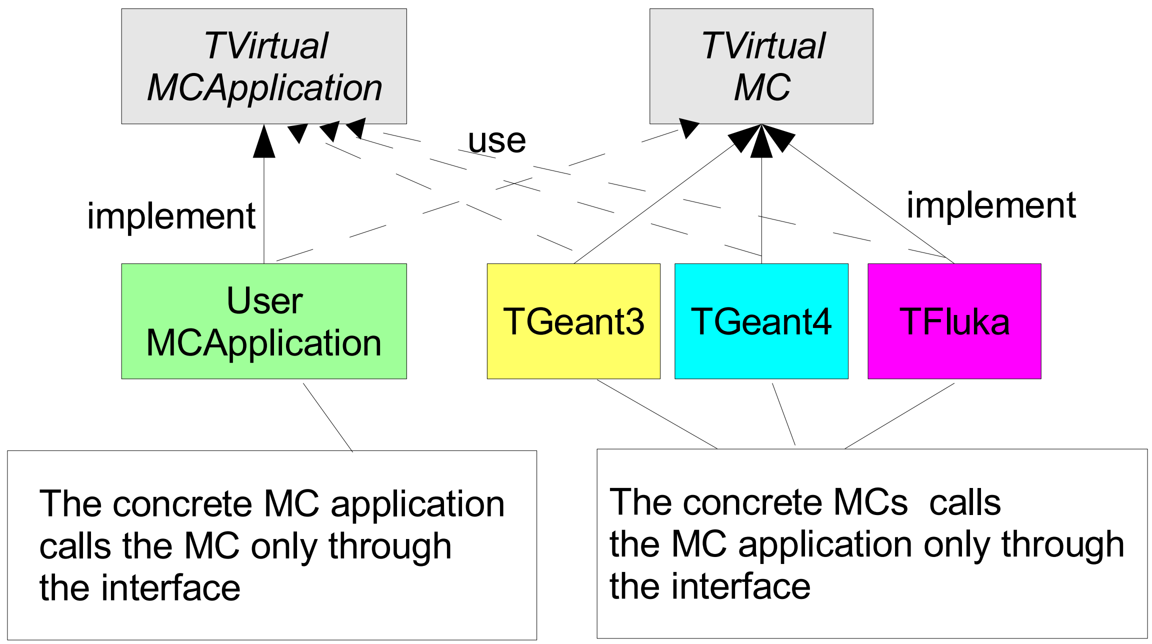
The VMC Concept

Thanks to an abstract VMC layer to Monte Carlo transport codes, the same user application code can be run with different simulation programs



The VMC Design

- In VMC, we introduce the abstract interface both for the MC simulation program and for the user application
- In this way we decouple the dependence between the user code and the concrete MC

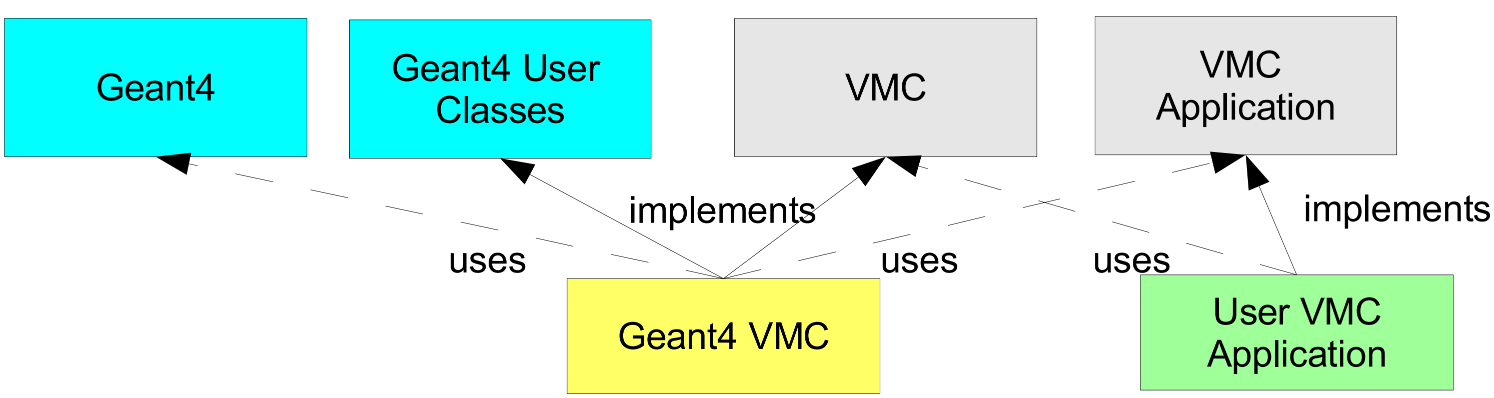


Virtual Monte Carlo

- VMC has been developed by the ALICE Offline project in the close collaboration with the ROOT team
- Since 2002 the VMC is distributed with ROOT
  - http://root.cern.ch/root/vmc/VirtualMC.html
- Now in use in more experimental frameworks

Geant4 VMC As A Geant4 Application

- To use the Geant4 toolkit, the user has to define their application based on the Geant4 user classes
  - Mandatory: detector construction, physics list, primary generator
- Geant4 VMC implements these Geant4 user classes via calls to the VMC application interface
  - Besides, it implements also the functions defined in the TVirtualMC interface



Geometry Definition

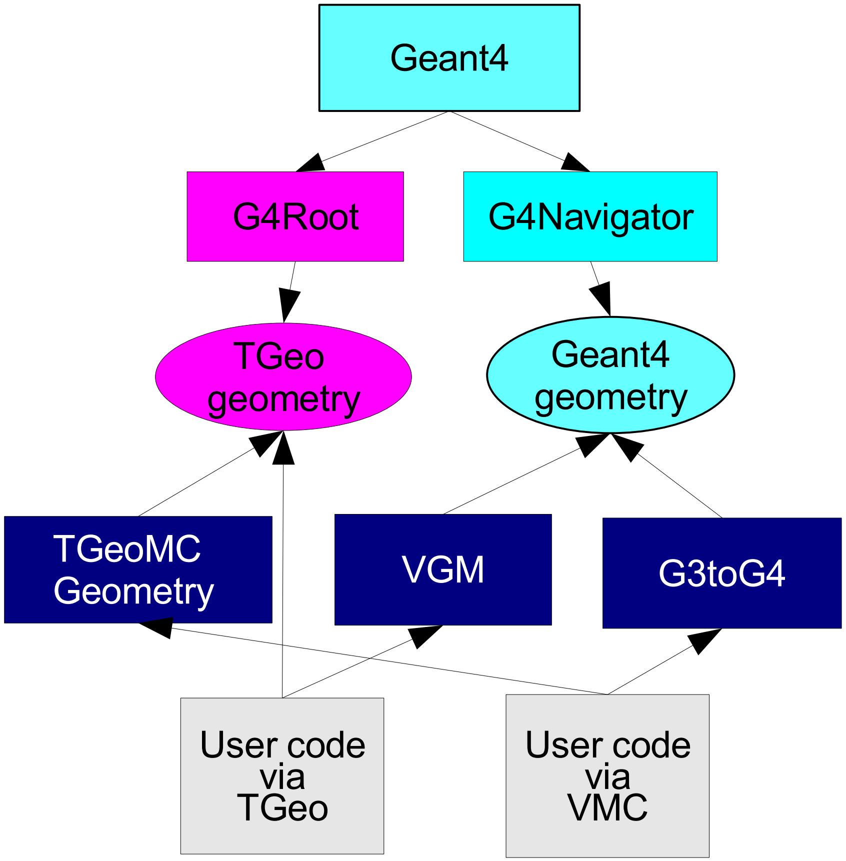
- Detector construction class in Geant4 VMC handles several ways of user geometry definition
- The geometry definition in the VMC has evolved:
  - First, using the VMC Geant3-like functions for building geometry,
    - This facilitated move to VMC for Geant3 users
  - The new approach came with the introduction of the own geometrical modeler in Root, TGeo
    - Independent from existing simulation tools
    - Provides IO, visualization and verification tools
  - Since the validation of TGeo with Geant3 VMC, the old way of geometry definition via VMC functions is deprecated

Geometry Definition (2)

- Support for Geant3-like VMC function in Geant4 VMC
  - With use of the G3toG4 package in Geant4
- Support for TGeo in Geant4 VMC
  - By geometry conversion from TGeo to Geant4
    - In Jul 2003 – roottog4 converter
    - Since Mar 2005 – moving to Virtual Geometry Model
      - See: The Virtual Geometry Model, Id 124 at this conference
  - In Dec 2006 – integration of G4Root
    - The interface between TGeo navigation and Geant4
      - See: An interface for GEANT4 simulation using ROOT geometry navigation, by A. Gheata and M. Gheata, Id 38 at this conference

Geometry Options

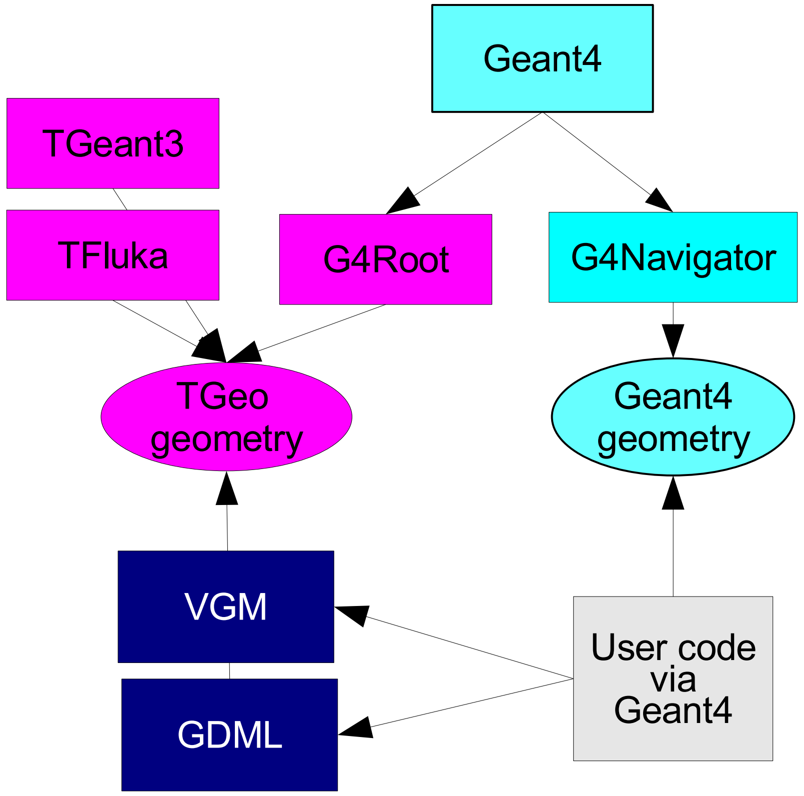
- User has a possibility to choose between Geant4 native navigation and G4Root navigation in both cases of geometry definition (via VMC or via TGeo)
- Possible selections:
  - VMCtoGeant4
  - VMCtoRoot
  - RootToGeant4
  - Root
  - Geant4
- The first word means geometry input, the second one navigator to be used



Geant4 VMC connects automatically necessary packages and activates selected navigator

Geometry options (2)

- With geometry definition via Geant4 user has a possibility to use the external converters to pass geometry to other MCs



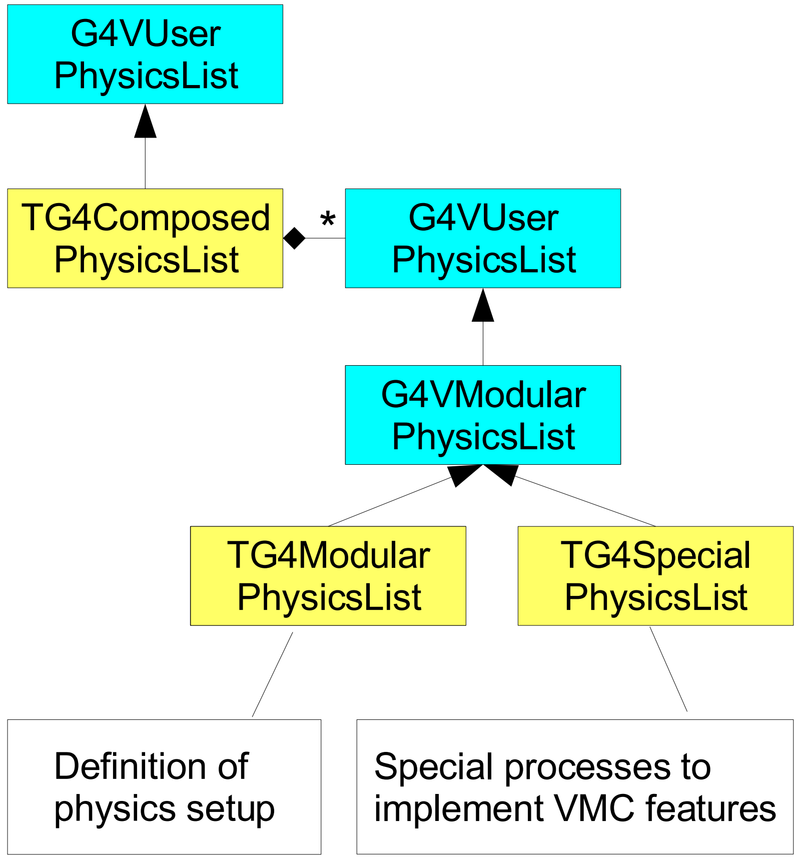
Geant4 VMC naturally supports geometry defined via Geant4 geometry model

Tracking Media

- In VMC
  - Tracking medium defined as in Geant3, it represents a set of tracking parameters associated to a material:
    - Sensitivity flag, parameters for magnetic field, maximum step, ...
- In Geant4 VMC
  - Geant4 did not adopted the Geant3 concept of tracking media
  - Use of G4UserLimits derived class to hold the relevant information:
    - The step limit, the vector of cut values, the vector of process controls
    - Some parameters are not applicable for Geant4
  - The user limits are used by the special processes to apply user defined values in tracking

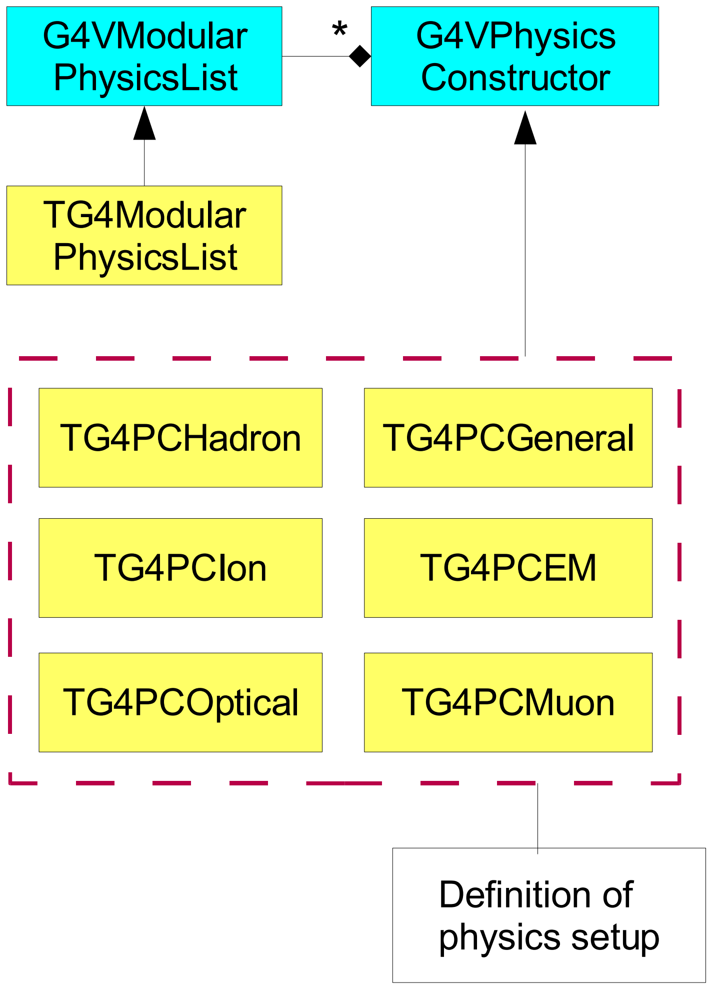
Physics Selection

- Geant4 does not have any default particles or processes.
  - User has to define them explicitly in their application
- Geant4 VMC provides a composed physics list including
  - The default physics setup, it can be replaced with own user physics list
  - The special processes implementing the VMC features



Modular Physics List

- The default physics setup has been collected from Geant4 novice examples and it is not tuned to any particular physics problem
- That is why the user is invited to implement their own physics list class.

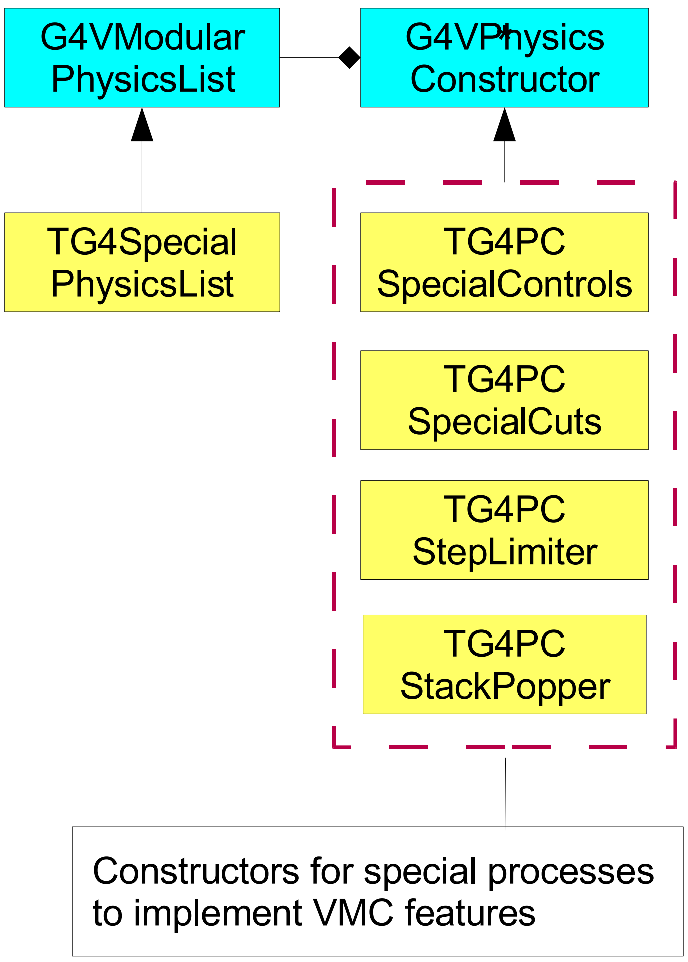


Special Processes

- The special processes are not activated by default, they have to be activated by user explicitly
- They can be re-used with a user physics list
- Special controls process
  - Implements the activation and inactivation of selected processes via VMC:

```
TVirtualMC::SetProcess(const char* flagName, Int_t flagValue);
```

  - Eg. `gMC->SetProcess("COMPT", 1);`
    - The flag names and meaning of the flag values as in Geant3
  - The user physics list has to fill in the map between the VMC flags and defined processes



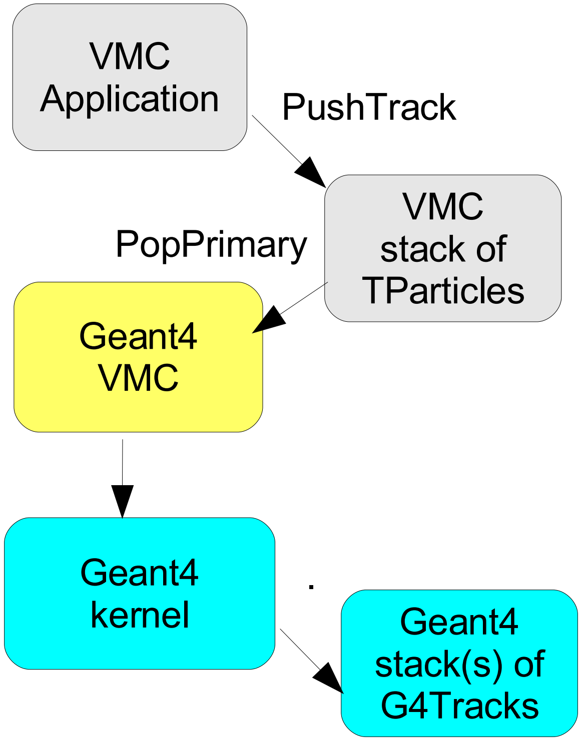
Special Processes (2)

- Special cuts process
  - In Geant4: cuts in range per particle and region
  - In VMC: G3-like cuts in energy, global or per tracking medium

    `TVirtualMC::SetCut(const char* cutName, Double_t cutValue);`
  - Eg. `gMC->SetCut("CUTGAMA", 1e-03);`
    - The cut names as in Geant3, the cut values in GeV
  - Applied as tracking cuts, not as threshold
- Step limiter process
  - The Geant4 G4StepLimiter process is used to limit the step if step limitation is selected by user.
- Stack popper process
  - Implements adding user defined secondary particles to tracking

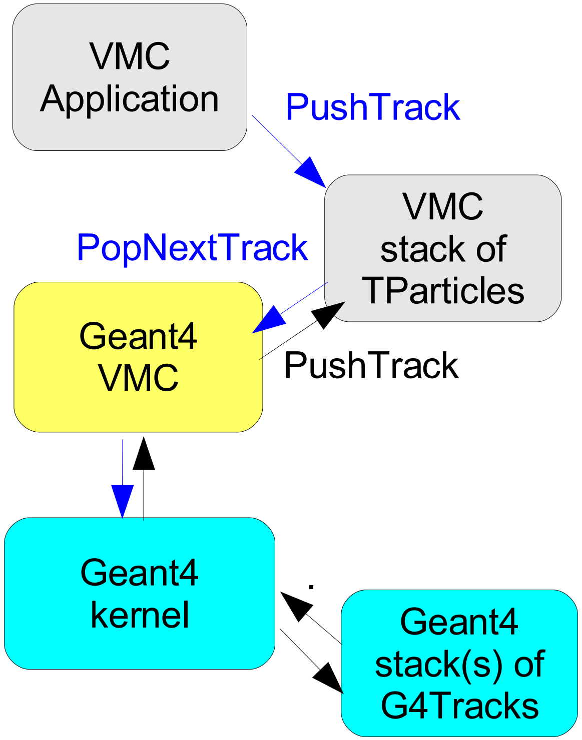
Primary generator & VMC Stack

- The Geant4 primary generator action is implemented with use of the VMC stack
- The VMC provides the interface for the stack of particles, which has to be implemented by user
  - User can also re-use the stack implementations from the VMC examples
  - The particles in the VMC stack are of the Root TParticle type
- The primary particles are first filled in the user VMC stack as TParticle objects, then they are transformed in the Geant4 objects and passed to Geant4 kernel



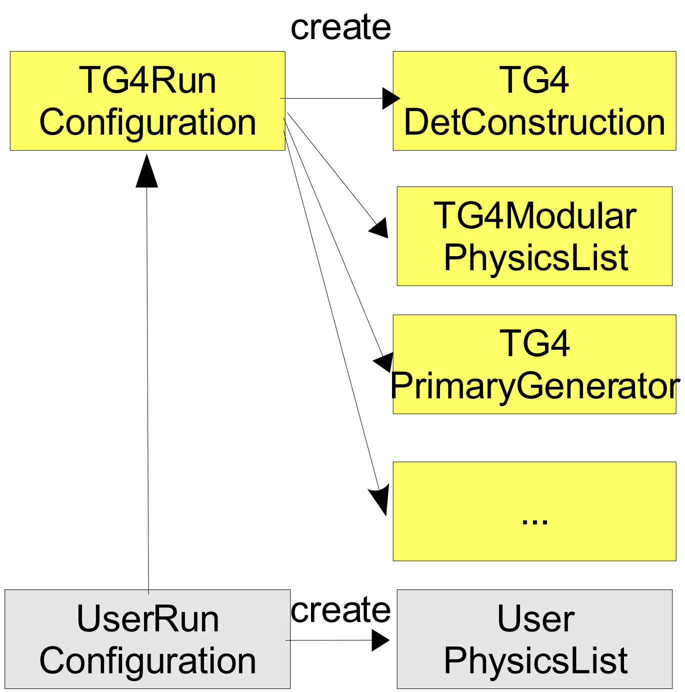
Secondaries & VMC Stack

- The secondary particles, created by Geant4 physics processes, are stored in the user VMC stack
  - By default, storing is done in the beginning of a track
  - Optionally, user can choose storing in the step of a parent track or no storing
- User can also activate a special StackPopper process, which monitors the VMC stack and pops particles defined by the user during tracking and which are then passed in Geant4 tracking
  - Implemented on the requirement by ALICE for simulation of the generation of feedback photons in an the avalanche close to a wire



User Run Configuration

- The run configuration class (TG4RunConfiguration) takes care of creating all Geant4 user defined mandatory and action classes that will be initialized and managed by Geant4 kernel
- The class can be extended in a user application by inheritance; this gives a user possibility to override or extend each Geant4 user defined class.
- Tested use cases (in the VMC example E03)
  - User physics list
  - User detector construction
  - User primary generator action



User Interface

Geant4:

- Implemented commands associated to selected Geant4 or user objects
  - Application state sensitive
  - Strong checking of user input

```
|Idle> /run/beamOn 1
```

Root:

- Based on the CINT interpreter
  - All Root and user objects which dictionary exists are accessible via the UI
  - Any object public function can be called at any time

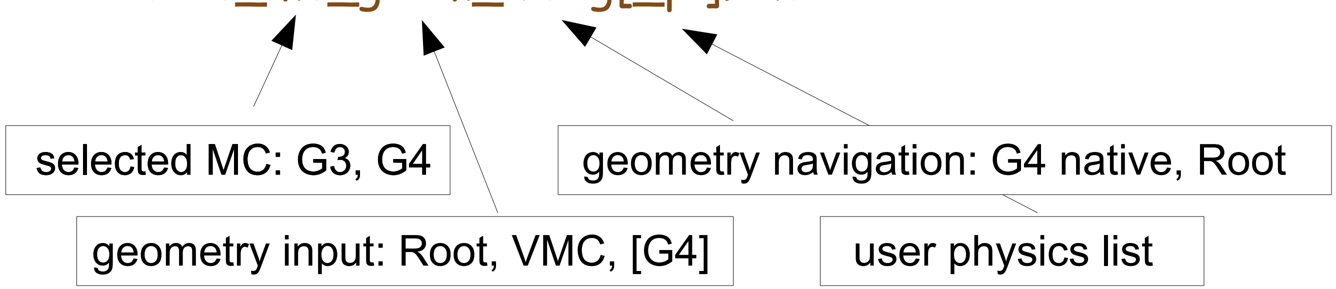
```
root [0] myApplication->Run(1);
```

Geant4 VMC:

- Both Geant4 UI and Root UI are available for the user
  - However it is not possible to access Geant4 objects from Root UI, as their dictionaries are not produced
  - Set of Geant4 commands implemented in Geant4 VMC, all start with 'mc'

VMC Examples &Test Suite

- VMC examples = Geant4 novice examples N01, N02, N03, N06 rewritten with use of Virtual Monte Carlo
  - Geometry defined via Root or optionally via VMC
  - E03 – user defined run configuration (with physics list and/or geometry defined via Geant4)
- In test suite script – all examples run with all options, the output is saved on the files and can be compared with the reference outputs stored in CVS
  - test_mc_geom_navig[_pl].out



The VMC Approach

Pros

- Possibility to run the VMC application with other transport codes with minimum effort and to compare the results
- Integration of VMC in Root framework
  - For the experiment frameworks based on Root it is seen as an advantage to develop simulation application fully in the same framework

Cons

- The Geant4 toolkit is constantly upgraded with new features, not all of them can be immediately interfaced in the VMC
  - Some of them may be difficult to be fulfilled with other MC implementations
- As the VMC represents an additional layer between the user application and Geant4, it brings some overhead in performance
  - It depends on the user application, but in average can be ~ 10 %

Conclusions

- Geant4 VMC is in production since 2002
- The present overview has covered the new features such as
  - Support for user defined Geant4 classes
  - Support for Root TGeo geometry definition and navigation
  - The user defined secondary particles
- The Geant4 VMC is under test by several collaborations who adopted the VMC approach
  - ALICE, CBM, PANDA, Opera, MINOS