Virtual Monte Carlo

I. Hrivnacova,
IPN Orsay

http://root.cern.ch/root/vmc/VirtualMC.html

ROOT 2002 Workshop, CERN
15 October 2002
Outline

- Introduction
- Packages
- Interfaces
- Available MCs
- Examples
- Distribution
- Future
Introduction

- The concept of Virtual Monte Carlo has been gradually developed by the ALICE Software project
- Class TGeant3 providing access to GEANT3 data structures (commons) and functions
  - Enabled to move user code from FORTRAN to C++
- Interface to MC as generalization of TGeant3
  - Started development for Geant4
  - However the implementations of MC interface depend on the ALICE software
- Interfaces to a user MC application
  - Enabled to remove the dependence on the ALICE software
Packages

- **MC**
  - The core package; provided within Root
  - Directory: root/mc; Library: libMC.so
  - Interfaces

- **Geant321 + VMC**

- **Geant4 VMC**

- **Examples**
  - Provided within Geant4 VMC
Interfaces

- Why interfaces?
- To decouple the dependence of a user code on a concrete MC
  - Enable to run the same user application with all supported Monte Carlo programs
- Authors:
  - R. Brun, F. Carminati, I. Hrivnacova, A. Morsch
Virtual Monte Carlo

TVirtualMC Application

TVirtualMC

UserMCAplication

Concrete MC application address to MC only through the interface

TGeant4

Concrete MCs address to user MC application only through the interfaces

TGeant3
Interfaces (1)

- **TVirtualMC**
  - Interface to Monte Carlo program
  - Generalization of Geant3 functions for definition of simulation task
    - Provides methods for definition geometry and physics setup, for access to tracked particle properties during stepping, visualization
  - Implementations: TGeant3, TGeant4
    - Are provided to a user
Interfaces (2)

- **TVirtualMCApplication**
  - Interface to a user application
  - Mandatory
  - Implementation has to be done by a user
Interfaces (3)

- **TVirtualMCStack**
  - Interface to a user defined particles stack
  - Mandatory
  - User can use the concrete stack classes provided in the examples or implement his own stack class

<table>
<thead>
<tr>
<th>Virtual MC Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetTrack</td>
</tr>
<tr>
<td>GetNextTrack : TParticle*</td>
</tr>
<tr>
<td>GetPrimaryForTracking : TParticle*</td>
</tr>
<tr>
<td>SetCurrentTrack</td>
</tr>
<tr>
<td>GetNTrack</td>
</tr>
<tr>
<td>GetNPrimary</td>
</tr>
<tr>
<td>CurrentTrack</td>
</tr>
</tbody>
</table>
Interfaces (4)

- TVirtualMCDecayer
  - Interface to an external decayer
  - Eg. Pythia6
  - Not mandatory

Virtual MC Decayer

- Init
- Decay
- ImportParticles
- SetForceDecay
- ForceDecay
- GetPartialBranchingRatio
- GetLifetime
- ReadDecayTable
Available MCs

- Implementation for 2 MCs available:
  - *Geant3*, *Geant4*
  - Implementation for Fluka is in development by the ALICE collaboration
Available MCs

Geant3 VMC

- Provided within a single package together with Geant321 (Fortran)
- File structure
  - geant321/*, TGeant3, minicern - source code
  - Makefile
  - config - platform dependent makefiles
  - lib - created at compilation time
- Geant3 Geometry Browser
  - Provided within TGeant3
  - http://alisoft.cern.ch/ people/morsch/Geant3GUI.html
- Authors:
  - R. Brun, F. Carminati, A. Morsch
Geant3 VMC

Geant3 Geometry Browser
Available MCs

**Geant4 VMC (1)**

- Provided within a package `geant4_vmc`
  - Requires Geant4 installation

- **File structure**
  - README, history, source, config
  - lib, tmp - created at compilation time
  - examples - not dependent on Geant4

- **Access to Geant4**
  - Geant4 classes are not processed by CINT - G4 objects are not accessible from Root UI
  - Switching between Root UI and Geant4 UI is available
Available MCs

Geant4 VMC (2)

- **Geant4 VMC extensions:**
  - **Geant4 VMC Geometry Browser**
    - Analogy to Geant3 Geometry browser
  - **Geometry XML Convertor**
  - Enables to export Geant4 geometry to XML and then to browse and visualize using GraXML
    - **GraXML** = Tool for handling HEP experiment Detector Description & Event data in Java 3D (developed for Atlas by J. Hrivnac)

- **Authors:**
  - D. Adamova, V. Berejnoi, A. Gheata, I. Hrivnacova
Geant4 VMC

Geant4 Geometry Browser

Visualisation attributes set using G4 VMC commands, setting in browser not (yet) available
Geant4 VMC - Geant4 Geometry in GraXML
Geant4 VMC

Limitations

- TVirtualMC has been inspired by Geant3
  - lead to certain difficulties and left Geant4 VMC with some limitations
- Geant4 VMC geometry is based on G3toG4 tool
  - G3toG4 limitations
  - They have been minimized with Geant4 4.0:
    - Support for reflections
    - Limited support for “MANY” volumes positions
- A few more minor limitations
  - None of them a real obstacle for using the VMC
  - More details can be found on VMC web page
Examples

- 2 examples provided in geant4_vmc
  - In spite of being provided within Geant4 VMC, they are built independently and do not require Geant4 installation in case a user wants to run them with Geant3 only

- Geant4 novice examples N01 and N02 rewritten with usage of Virtual Monte Carlo

- Implement MC Application and MC Stack

- Both examples are executed in the same way:
  - cd geant4_vmc/examples/E01 or E02
  - root
  - root[0].x run_g3.C or run_g4.C
Examples

E01

- Rewritten Geant4 novice example N01
- Builds simple geometry, tracks geantino and prints info at each step
- Classes:
  - Ex01MCApplication, Ex01MCStack, Ex01Particle
void Ex01MCApplication::Construct Geometry()
{
    //------------------------------ experimental hall (world)
    Double_t expHall[3];
    expHall[0] = 300.;
    expHall[1] = 100.;
    expHall[2] = 100.;
    gMC->Gsvolu("EXPH","BOX", fImedAr, expHall, 3);

    //------------------------------ a tracker tube
    Double_t trackerTube[3];
    trackerTube[0] = 0.;
    trackerTube[1] = 60.;
    trackerTube[2] = 50.;
    gMC->Gsvolu("TRTU","TUBE", fImedAl, trackerTube, 3);

    Double_t posX = -100.;
    Double_t posY = 0.;
    Double_t posZ = 0.;
    gMC->Gspos("TRTU", 1 ,"EXPH",
                posX, posY, posZ, 0, "ONLY");

    // ... etc
}
{
    // Load basic libraries
    gROOT->LoadMacro("../macro/basiclibs.C");
    basiclibs();

    // Load Geant3 libraries
    gROOT->LoadMacro("../macro/g3libs.C");
    g3libs();

    // Load Geant4 libraries
    gROOT->LoadMacro("../macro/g4libs.C");
    g4libs();

    // Load this example library
    gSystem->Load("libexample01");

    // MC application
    Ex01MCApplication* appl
        = new Ex01MCApplication("Example01", "The example01 MC application");

    appl->InitMC("g3Config.C");
    appl->InitMC("g4Config.C");

    appl->RunMC(1);
}

run_g3.C
run_g4.C
void Config() {
new TGeant3("C++ Interface to Geant3");
}

void Config() {

// RunConfiguration for Geant4
TG4RunConfiguration* runConfiguration = new TG4RunConfiguration();

// TGeant4
new TGeant4("TGeant4", "The Geant4 Monte Carlo", runConfiguration);
}
Examples

E02

- Rewritten Geant4 novice example N02
  - Builds parameterised geometry with magnetic field, draws geometry, tracks proton, registers hits in a tracker and saves hits and kinematics in a Root file
    - Persistence only in VMC (not in the Geant4 example)
  - Classes:
    - Ex02MCApplication, Ex02MCStack, Ex02Particle, Ex02DetectorConstruction, Ex02ChamberParameterisation, Ex02MagneticField, Ex02TrackerSD, Ex02TrackerHit, Ex02RootManager
Examples

AliRoot

- **AliRoot Framework**
  - Integrates simulation, reconstruction and analysis ALICE software
- **Simulation based on Virtual Monte Carlo**
  - A complex “example” of VMC usage
  - Users stay with Geant3
- **Each detector subsystem has one single package**
  (one directory, one library)
  - Dependence only on STEER
  - No MC specific code
I. Hrivnacova

AliRoot

Detectors
- ITS
- TPC
- MUON
- PHOS
- PMD
- RICH
- FMD
- TOF
- TRD
- START
- ZDC

STEER
- run management interface classes
- detector base classes
- data structure base classes

EVGEN

PYTHIA6

HIJING

... 

MICROCERN

VMC

ROOT

Geant4 VMC

Geant3 VMC

Geant4

Geant3

External packages
Examples

AliRoot Run With Virtual MC

- Concrete Monte Carlo is selected and loaded dynamically at run time

- Steps:
  - aliroot
    - The main program creates the application object gAlice
  - root [0] gAlice->Init("Config.C");  
    - G3 simulation
  - root [0] gAlice->Init("g4Config.C");  
    - G4 simulation
    - Application is initialized with either G3 or G4 configuration file that instantiates either G3 or G4
  - root [1] gAlice->Run();
    - Simulation run with selected MC
Distribution

- **ROOT CVS server**
  - `mc` - in `root`
  - `geant3, geant4_vmc`
    - 2 new independent modules (parallel with `root`):
      - `cvs -d:pserver:cvs@root.cern.ch:/user/cvs co geant3`
      - `cvs -d:pserver:cvs@root.cern.ch:/user/cvs co geant4_vmc`

- **ROOT Web**
  - Distribution of sources:
    - `geant321+_vmc.0.1.tar.gz`
    - `geant4_vmc.0.1.tar.gz`
Future

- **Fluka VMC**
  - In development by ALICE Software Project
  - A. Fasso, E. Futo, I. Gonzalez, F. Carminati, A. Morsch

- **TGeo + VMC**
  - It is foreseen to enable VMC geometry definition using the new geometrical modeller in Root
  - R. Brun, A. & M. Gheata
Summary

- **VMC** allows to run different simulation Monte Carlo from the same user code
  - *On the menu today:* Geant3, Geant4
  - *In preparation:* Fluka
- **The first release of VMC**, Geant3+VMC, Geant4 VMC and examples available: