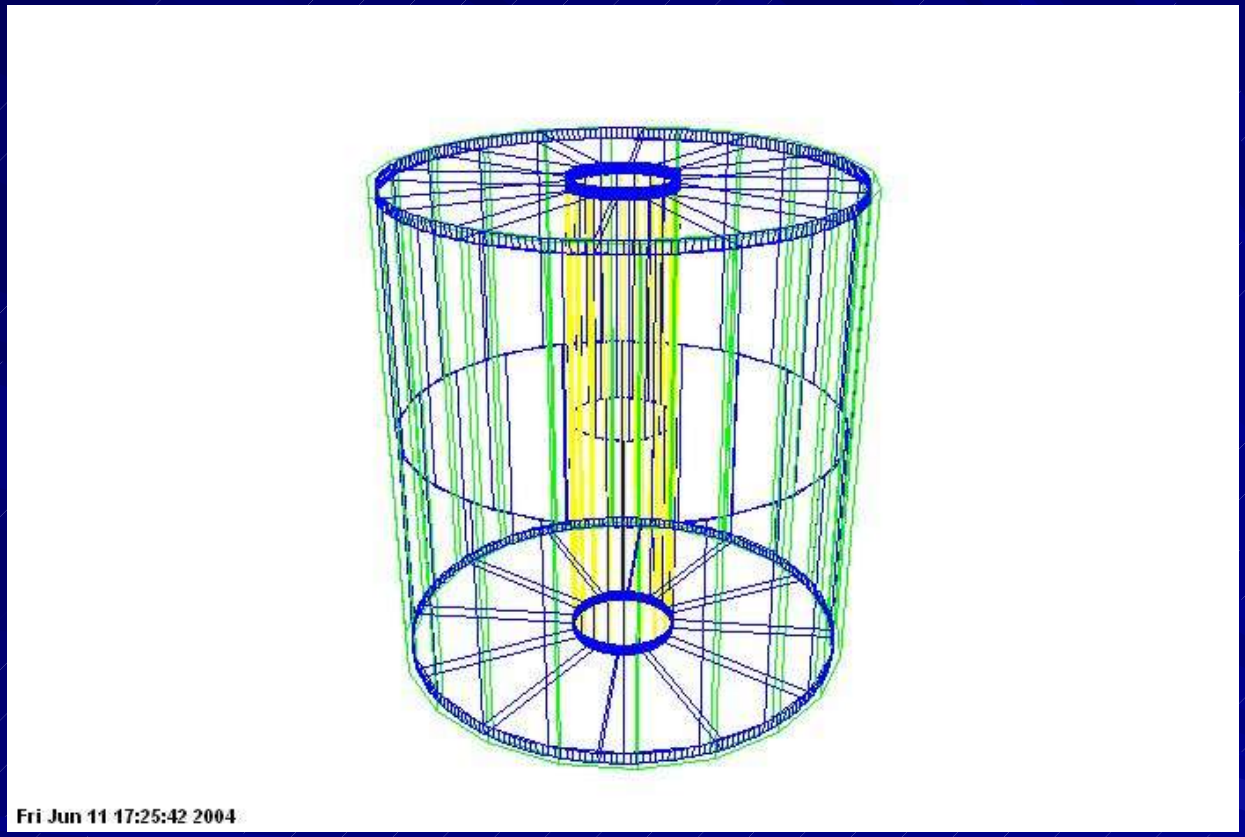
Virtual Monte Carlo and new geometry description in STAR



Maxim Potekhin

STAR Collaboration Meeting, BNL

July 17, 2004

Overview

- What is Virtual Monte Carlo (VMC) and its origins? ⇨
- Why are we interested in VMC and alternative geometry models? ⇨
- Current status of the Star VMC and Detector Description project ⇨
- Plans for the VMC ⇨

Geant 3 and 4 as precursors of the Virtual Monte Carlo (VMC)

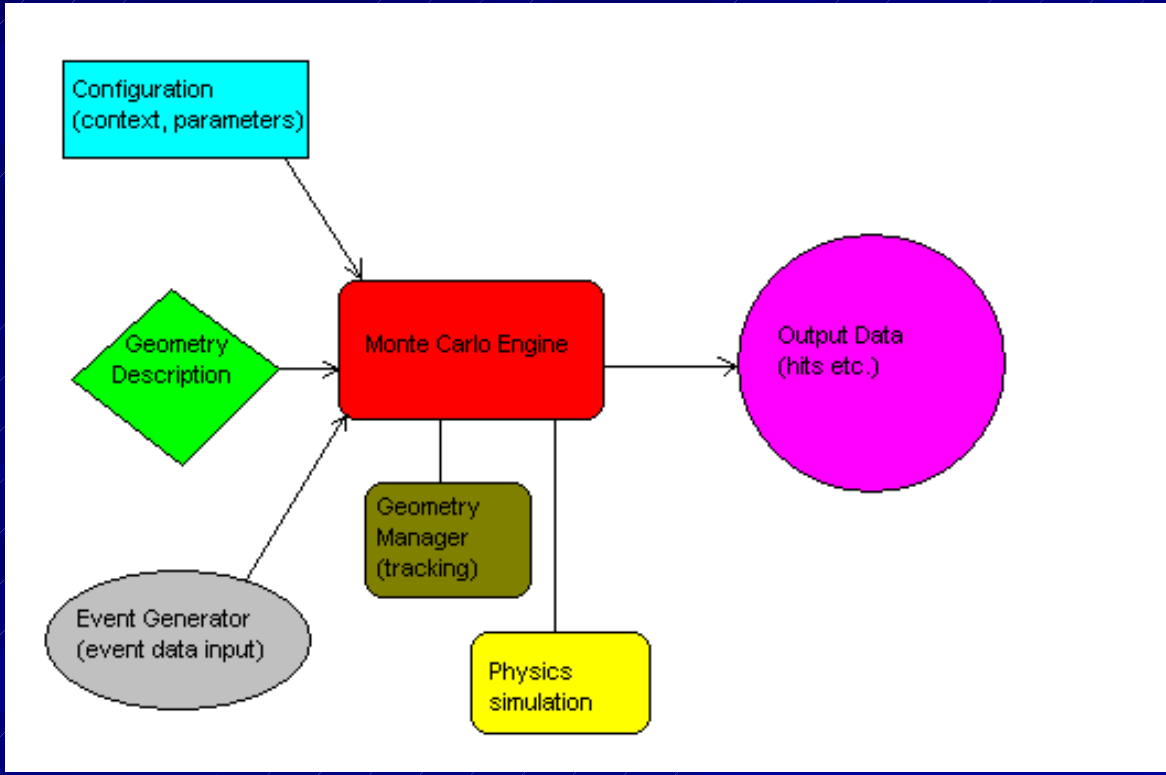
- Geant 3: still the mainstay of many simulation applications
- Geant 4 gaining steam
- Different language platforms and approaches (tracking and physics)
- Each represents a fairly closed system:
  - different geometry models
  - lack of platform neutral geometry description
  - no interchangeable components between the two
  - difficulty of interfacing any other component such as a third party geometry manager
- Transition from 3 to 4 inevitable but a major (and expensive) project for any collaboration
  - geometry (re) definition a major problem (re-coding? conversion?)
  - other infrastructure development

The VMC concept

Virtual Monte Carlo is based on the modular approach to the application design.

Individual components (in G3, G4) are identified and given appropriate interfaces.
The user code can be run, with minimal or no changes, with

- different geometry management software,
- different physics simulation etc.



What is the VMC, in practical terms?

In the narrow sense, as a software currently in existence, the VMC

- is a set of ROOT classes meant to implement the design goals stated above. Large part of the code is dedicated to hiding the implementation detail of the concrete MC engine
- has continuing support of the core ROOT development team
- has been tested with at least two underlying simulators, Geant 3 and 4, and there is a promise of eventually supporting Fluka
- Features the important option to use different geometry models:
  - has a generalized geometry building and tracking interface similar to Geant 3
  - does a good job of geometry encapsulation
  - has been tested with an alternative geometry manager, based on the ROOT geometry system (often referred to as TGeo)
- appears to fulfill the promise of having a single user application that would work with different underlying components

Our interest the VMC

- the VMC is a hedge against our simulation framework becoming obsolete in terms of user expertise and platform change.
- we want to increase transparency to the users, in part due to benefits of OO programming and the more familiar language platform (C++).
- VMC can be seen as a bridge between Geant 3 and 4, and future MC platforms
- Has the potential to become a long term MC solution for STAR, with a stable API, and a geometry description system that facilitates the ongoing detector development and upgrades

Our interest the VMC

- it is a step to make the STAR software more robust by implementing a unified geometry description, i.e. finally establishing the SAME and CONSISTENT geometry in simulation and reconstruction (still an outstanding issue)

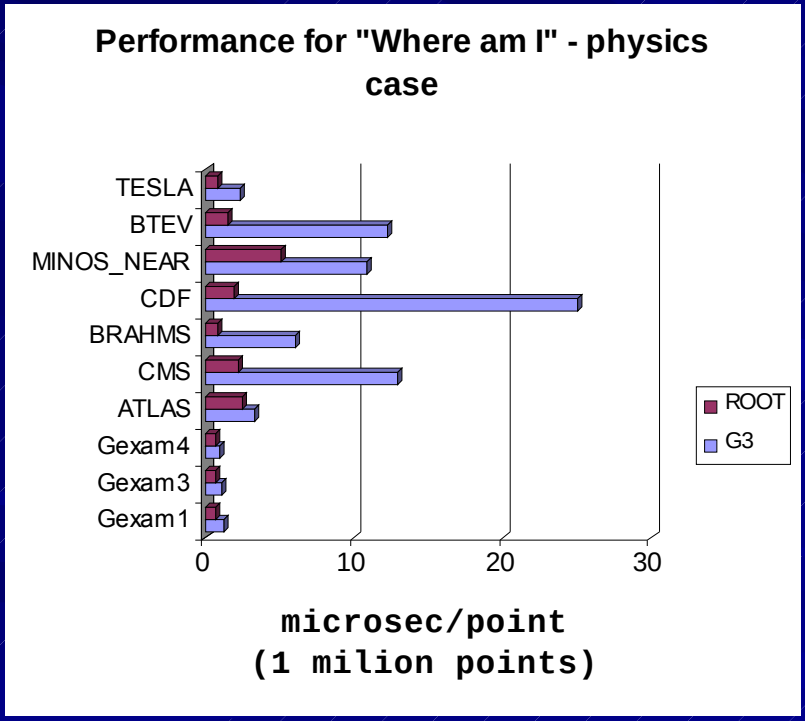
- VMC offers the possibility of using an alternative geometry manager (TGeo), which opens the possibilities of:
  - easier geometry navigation by the user, from C++ code
  - consistent interface with reconstruction
  - more flexible, and portable, geometry description: geometry persistent in ROOT files, XML (and its derivatives), possible CAD systems interface, various TGeo based interfaces, event displays
  - more practical description of the “real” geometry, e.g. based on surveys; hence “better simulation”

The TGeo component and its role in VMC

- the TGeo is a broad colloquial term used to describe a set of geometry related ROOT classes
- it is feature-rich and allows one to build complex geometries
- it is suitable for building a Geant-like application such as VMC
- can be used in event displays and potentially offers sophisticated graphic functionality (confer recent Qt work by V.Fine)
- can be considered “platform neutral” as it is not really tied to any particular application such as Geant, and can be productively used to navigate geometry in STAR reconstruction
- an attractive component to have in any future simulation framework

Our interest the VMC and TGeo

- there is a potential of substantial performance gain (subject still open to testing, validation etc). Confer the following diagram from CHEP'03 (authors:René Brun, Andrei Gheata, Mihaela Gheata)



Current status of the STAR VMC project

- the team: V.Fine, Y.Fisyak, V.Perevoztchikov, M.Potekhin

- Accomplished steps:
  - analysis of the existing VMC applications including ALICE
    - possibility of software reuse considered
    - an example of a large detector implementation
  - initial design of core classes for STAR
  - Geant 3 to ROOT geometry converter built into ST_geant_Maker (work in progress)

Current status of the STAR VMC project

- Accomplished steps:
  - validation of the ROOT geometry manager in the VMC context:
    - same set of tracks injected into a realistic STAR geometry,
    - regressions test: histograms of volumes and materials hits, track lengths in different volumes
    - differences identified and mostly understood (work in progress)
    - No “show stopper” problem with the TGeo geometry manager found so far
    - Issues requiring close attention do exist (divisions, “many” etc)
  - test application created
  - unexpected benefit: validation of the actual STAR geometry (some bugs found and fixed)

Detector Description in STAR VMC

- Detector Description is the method by which the detector geometry model is defined by the user. It can be a piece of Fortran code, a CAD system file, a piece C++ code, a XML file etc.

- *Now the Detector Description component should be viewed in a broader context of the VMC, as one of its components*

Detector Description in the context of STAR VMC

- What do we have now, and what can we hope to have in terms of tools and formats for the Detector Description in STAR?
  - Backward compatibility and building a bridge between the old and new applications:
    - the STAR software creates a Geant 3 based geometry model and exports it as a ZEBRA file
    - existing g2root converter opens a possibility of reuse; reads ZEBRA and exports geometry as a C++ (cint) macro
    - a new method in the St_geant_Maker does a similar conversion in memory
  - Persistent feature of the ROOT geometry tested and used (geometry saved to, and reloaded from, .root files)

Detector Description

- Directions for the Detector Description in STAR
  - it appears likely that we shall use one (or all) of the following:
    - C++ code with a proper API to interface the database
    - A XML (or its derivative such as GDML etc), which would necessitate the creation of the parser
    - geometry components obtained by conversion from Geant/Zebra
- The beauty of VMC is that it allows us to do that!
- We will try of course to find a uniform solution (XML solution being of particular interest)

Detector Description

Open issues in Detector Description:

- Have to make a choice of a particular language (schema), such as GDML, AGDD etc
- None of the existing parsers is useable right away in the TGeo context, so effectively this needs to be implemented almost from scratch
- Separation of geometrical structure and the data component: current lack of a standard database interface in the existing implementations (will probably have to do it ourselves and/or establish a joint project with interested parties)

STAR VMC plans

- Short term (2-10 weeks)
  - finalize the design of the core VMC classes
  - create a testbed application that would allow the development team to collaborate on the project
  - build an interface with reconstruction (i.e. persist the MC data in the form that can be fed into reco)
- Medium term (2-12 months)
  - make a choice of the Detector Description and implement it
  - build a fully functional VMC application prototype
  - conduct physics tests
  - work out integration with reco
- Long term (12-18 months)
  - have the STAR detector fully described within the new framework
  - have the TGeo interfaced and used in reco
  - phase out staf/gstar/starsim