Geant4 VMC 3.0

I. Hřivnáčová[1], A. Gheata[2]

[1] Institut de Physique Nucléaire (IPNO), Université Paris-Sud, CNRS-IN2P3;
[2] CERN

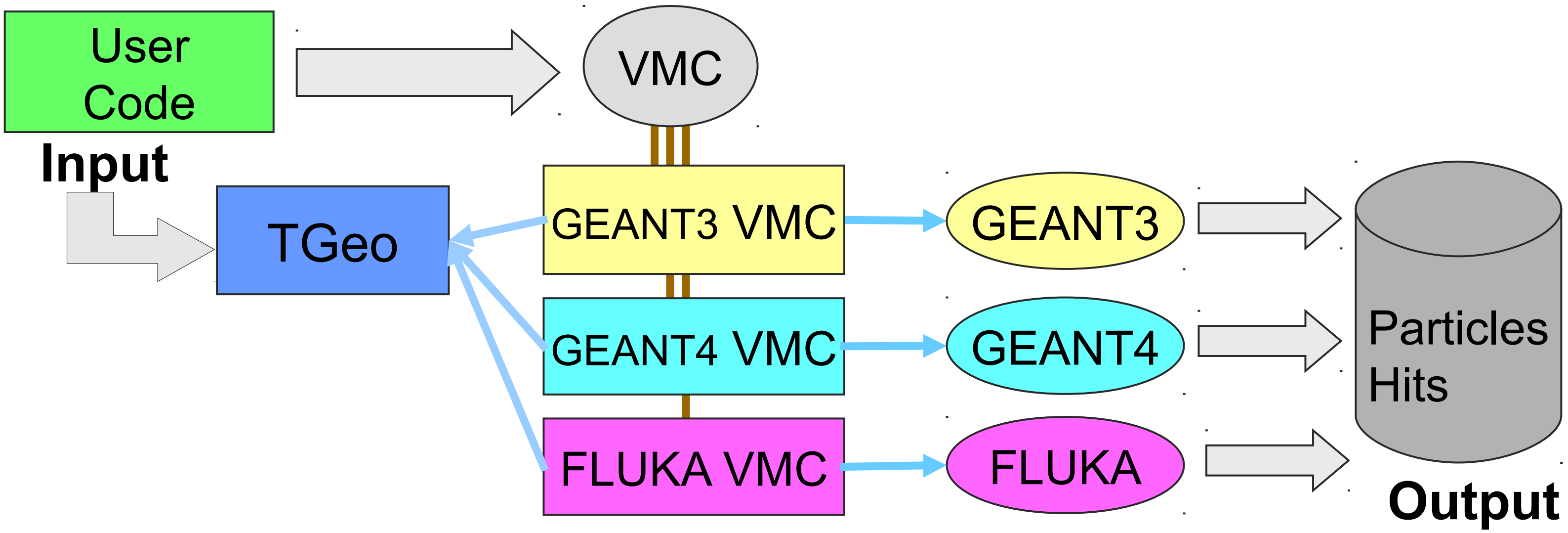
21st International Conference on Computing in High Energy and Nuclear Physics,
13 -17 April 2015, Okinawa

Outline

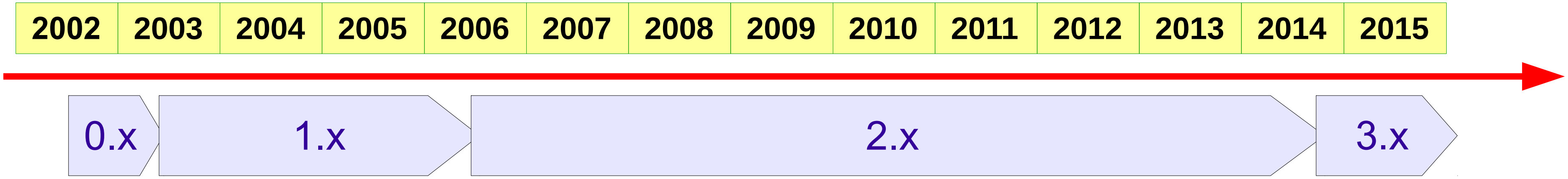
- VMC framework
- Geant4 VMC history
- New features in 3.0
  - Multithreading
  - CMake build
  - Migration to Root 6
  - Improved tests

Virtual Monte Carlo

- The Virtual Monte Carlo interface (VMC) allows to run simulation with different Monte Carlo codes from the same user application
  - Developed by the ALICE Offline Project and further included in ROOT
  - Reached stability during the last decade and have become a foundation for other detector simulation frameworks, the FAIR facility experiments framework being among the first and largest
  - http://root.cern.ch/drupal/content/vmc



Geant4 VMC History



2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015

0.x 1.x 2.x 3.x

Oct 2002	Jul 2003	Dec 2006	Nov 2014
v0.1	**v1.0**	**v2.0**	**v3.0**
First version	Geometry convertors (roottog4, …)	G4Root Navigation	Multithreading CMake Root 6
.. v0.5	.. v1.9	.. 2.15	.. 3.1 ..
Root 3.03/09 Geant4 4.1	Root 3.05/06 Geant4 5.2	Root 5.14/00 Geant4 8.2	Root 5.34/23 and 6.02/01 Geant4 10.00.p03

Towards Multithreading

- Development of Geant4 VMC MT started at **end of 2011**
  - Based on Geant4 MT prototype
- The prototype was described in CHEP 2012 publication:
  - I. Hřivnáčová 2012 J. Phys.: Conf. Ser. **396** 022024
- Geant4 VMC code was adapted for multithreading using the same approach as in Geant4 MT
  - Replacement of the singleton objects in Geant4 VMC with singletons per thread, including the main classes: TVirtualMC and TVirtualMCApplication
  - Applied modifications, as described in the Geant4MT User's Guide, to the Geant4 VMC classes, mainly in geometry and run categories
  - ROOT output implemented per thread and adapted to multithreading
- G4Root was not yet migrated in this prototype

Geant4 VMC 3.00.b01

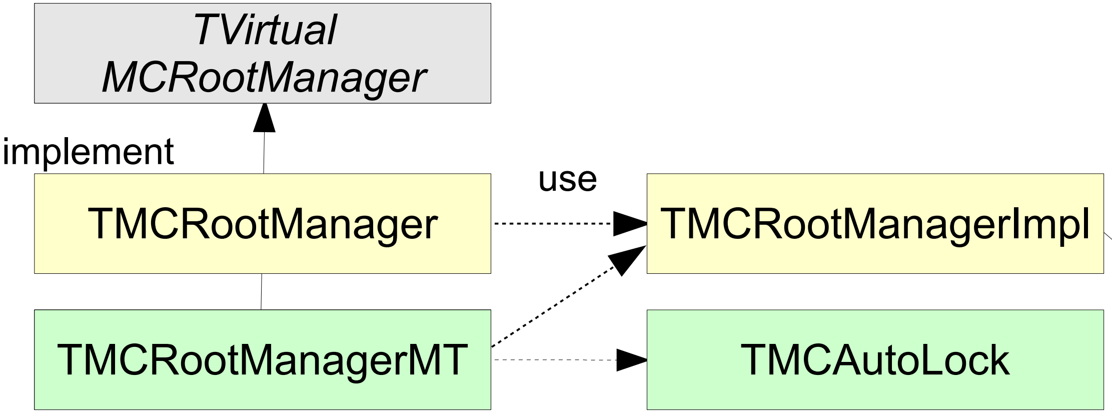
- First (beta) Geant4 VMC version providing support for Geant4 multithreading mode was released **on 14 March 2014**
  - Based on Geant4 10.00.p01
  - The Geant4 VMC prototype was adapted to changes in the Geant4 interfaces for user application
- Migration of G4Root to multithreading
- Single source code for both sequential and multithreading modes
- All (5) VMC examples were migrated to MT
- Added examples main functions and CMake configuration files for building examples programs linked with all libraries
  - With previous versions the VMC examples can be run only from the ROOT session via a ROOT macro

Geant4 VMC 3.00

- Geant4 VMC version providing the support for Geant4 multithreading mode was released **on 17 November 2014**
  - Based on Geant4 10.00.p03
- Consolidated migration to Geant4 multithreading
- Complete move to CMake build system
  - Removal of "old" build system
- Migration to Root 6.02/xx
  - This involved building against C++11 standard

MTRoot

- A new set of classes which take care of locking critical operations (registering ROOT objects to trees etc.) in multithreading mode is introduced in new **mtroot** package
  - http://root.cern.ch/drupal/content/mtroot
- In previous Geant4 VMC versions, a single class Ex02RootManager for Root IO management was provided in examples



Reusing the implementation in both sequential and multithreaded managers

G4Root

- G4Root allows usage of ROOT geometry navigation in sequential and now also multithreading mode
  - Each worker thread has its own navigator
  - Fixed propagation of navigators pointers to stepping manager in multithreading mode
- Replaced the old G4Root test based on Geant4 example N06 with one based on OpNovice
  - Working also in multithreading mode
- Fixed G4Root navigation in geometries using ROOT assembly volumes
  - Leading to stuck particles getting killed

Migration of VMC Applications

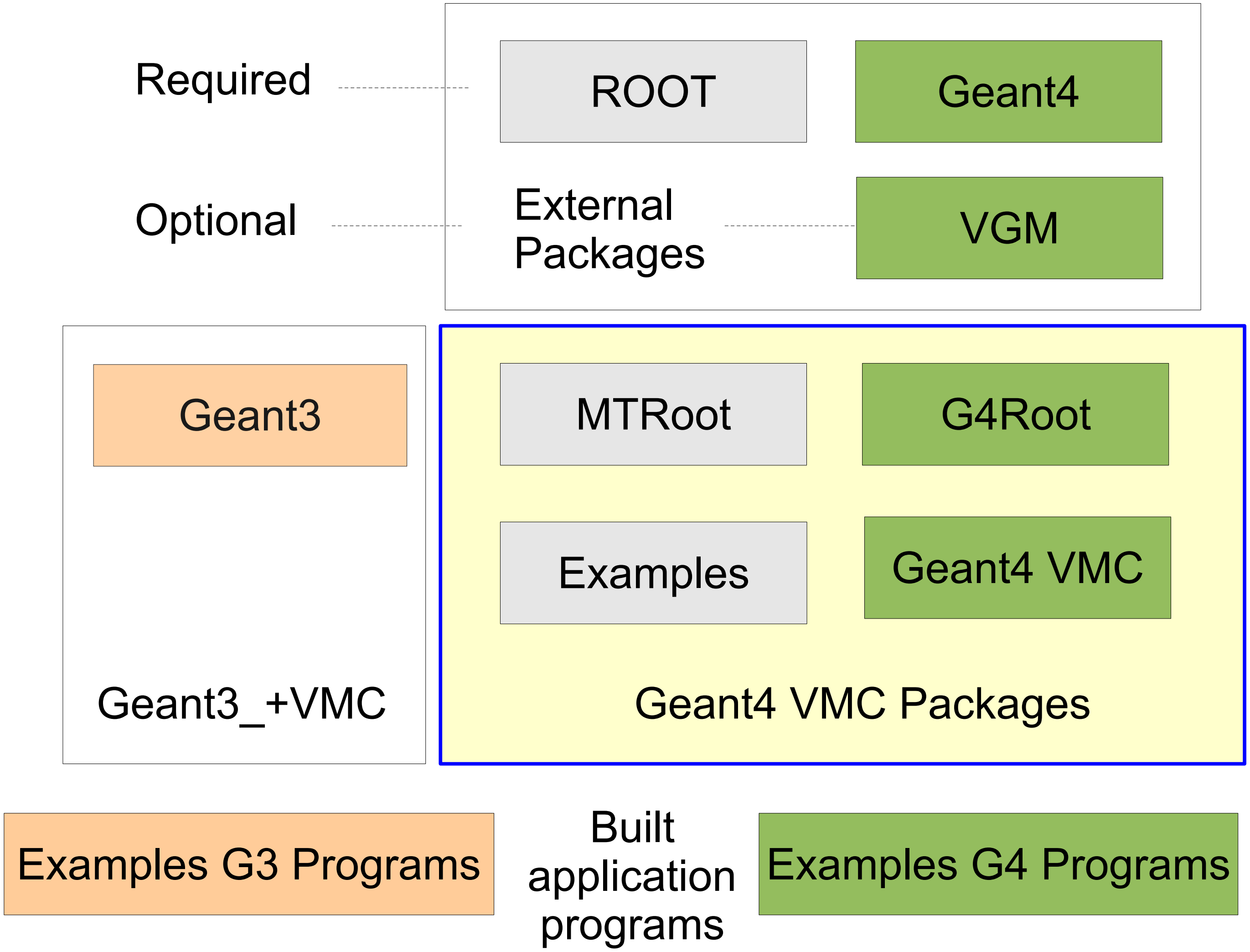
- Users need to implement new functions of TVirtualMCApplication which are then used to clone the application and its containing objects on workers
- Creating of the objects on worker threads is then triggered from the Geant4 VMC classes
  - Examples and more detailed instructions are available from the VMC Web site

class TVirtualMCApplication

```
// required for running in MT
virtual TVirtualMCApplication* CloneForWorker() const;
// optional
virtual void InitForWorker() const;
virtual void BeginWorkerRun() const;
virtual void FinishWorkerRun() const;
virtual void Merge(TVirtualMCApplication* localMCApplication);
```

VMC Packages

- Geant4 VMC besides its “own” set of classes (built in geant4_vmc library) includes also
  - G4Root
  - MTRoot
  - Examples
  - Each with different dependencies



CMake Build

- Replaced the “old” build system based on Makefiles in ROOT
- Inspired by Geant4 and VGM projects
- Build options:

Geant4VMC_BUILD_P	P: G4Root, MTRoot, Geant4VMC, EXAMPLES	ON
Geant4VMC_USE_P	P: G4Root, VGM [OFF], GEANT4_UI, GEANT4_VIS, GEANT4_G3TOG4 [OFF]	ON
Geant4VMC_INSTALL_P	P: EXAMPLES	ON

- Configuration files, PConfig.cmake, for all included packages
  - Can be then used directly in the client projects, no need to define FindP.cmake
- Find packages FindP.cmake, P = Geant4, ROOT
- And some more utility files

Examples CMake Files

- Find and Use files for VMC and MC
  - FindP.cmake, UseP.cmake, P = VMC, MC
- Find [V]MC
  - Find all needed packages according to configuration options
- Use [V]MC
  - Set compiler definitions, includes and libraries for all packages according to configuration options
- The examples CMake file can then contain only examples specific setting
- Provided in both geant4_vmc/cmake and geant3/cmake and can be used in users VMC applications

Migration to ROOT 6

- Examples macros for loading libraries, load_g3|4.C, separated from run macros, run_g3|4.C
  - Required for Cling
- C++11
  - Minor fixes were needed in Geant4 VMC code to get it compiling with C++11 standard, required with ROOT 6
- CMake configuration files
  - Updated generation of dictionaries for changes in ROOT 6
  - Added generation of “rootmaps” which enable library autoloading: no need for explicit library loading via ROOT gSystem object

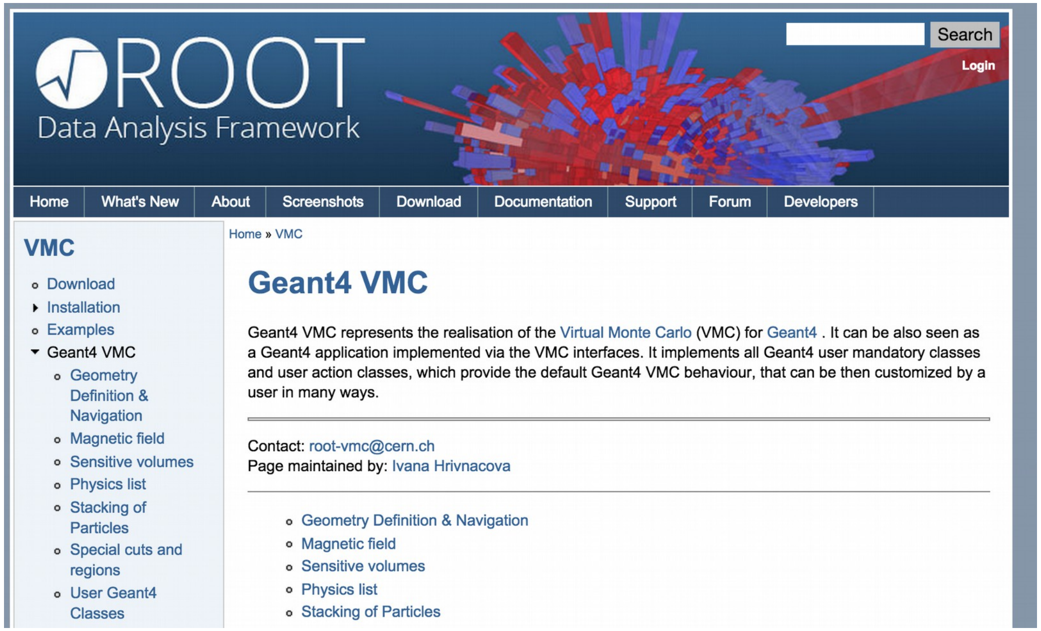
Tests

- Test suites shell scripts:
  - test_suite.sh – 21 test configurations run from the ROOT session
  - test_suite_exe.sh – the same tests run from examples programs
  - test_physics_lists.sh – test all Geant4 available physics lists
- Improvements in 3.0:
  - Added a possibility to select Geant3/Geant4 test only via arguments
  - Added summary messages and return codes (they allow automatically evaluate the result of the tests)
  - Possibility to define a build directory via arguments

- Thanks to Oliver Freyermuth, Physics Institute of the University of Bonn for testing and contribution to CMake build, examples test suites and ROOT 6 migration

User Support

- The Web site integrated in ROOT Drupal
  - https://root.cern.ch/drupal/content/geant4-vmc
- New documentation pages
  - Installing geant4_vmc
  - Installing and Running Examples
  - Multithreaded processing
- Release notes with a detailed description of changes for all versions
- The JIRA VMC project for bug reports since October 2014



Geant4 VMC

Geant4 VMC represents the realisation of the Virtual Monte Carlo (VMC) for Geant4 . It can be also seen as a Geant4 application implemented via the VMC interfaces. It implements all Geant4 user mandatory classes and user action classes, which provide the default Geant4 VMC behaviour, that can be then customized by a user in many ways.

Contact: root-vmc@cern.ch
Page maintained by: Ivana Hrivnacova

- Geometry Definition & Navigation
- Magnetic field
- Sensitive volumes
- Physics list
- Stacking of Particles

Conclusions

- Geant4 VMC 3.0, multithreading capable, is available since November 2014
  - A consolidated version 3.1 based on Geant4 10.01 since December 2014
- The interest in Geant4 multithreading in both ALICE and FAIR
  - Migration of FairRoot framework to multithreading in progress
  - Ongoing ALICE tests with a multithreading prototype based on a VMC example with realistic ALICE geometry, magnetic field, primary event generator and simplified TPC hits
- Improvements in build system and testing