Bruno: new developments and future prospects

A. Di Simone
INFN Tor Vergata

Outline

- Latest developments
  - Physics Recipes
  - Optical photons
- Plans for the future
  - Support for ongoing studies
  - Packaging (…)
  - The longer term

Physics Recipes

- Use case presented by Riccardo at last meeting
  - Switch on/off individual physics processes on a per-volume basis
- Similar functionality is now implemented in Bruno, by means of “Physics Recipes”
  - Allow to tune the physics list with needed granularity
  - Caveat: this is potentially very dangerous. Do not use it unless you know what you are doing
    - you may severely harm the reliablity of the simulation
- Granularity is defined by Regions, not by Volumes
  - Makes more sense, since all physics-related quantities (such as production cuts) are related to the regions

Physics recipes

- In order to switch off or on a process, just create a region and associate a recipe to it:

```
/regions/create_region DIRCRegion DircWorld
/regions/physicsRecipe DIRCRegion myBadProcess remove
/regions/physicsRecipe DIRCRegion myNiceProcess add
```

- Every time a particle enters the DIRCRegion, myBadProcesses is suspended and myNiceProcess activated
- When the particle exits the region, the changes are undone, and the ones corresponding to the next region are applied (if any)
- This kind of manipulation requires that both processes are already present in the physics list.
  - Always keep in mind that by adding or removing with this procedure you are actually just suspending or resuming the processes.

Optical processes

- Optical processes deal with special particles called "opticalphoton"
  - think of them as photons in the optical range
  - it is important to keep in mind that they are not just "gammas" with lower energies
  - no matter how much you lower the energy of a "gamma" it will never become an "opticalphoton".
- The present implementation of Bruno masks all optical simulation behind a command line flag ("-O", capital o, not zero).

```
./bin/Linux-g++/Bruno -O myOpProp.mac -m singleparticle.mac
```

Creating optical photons

- Optical photons are created in G4 by specific processes (such as Cerenkov or Scintillation)
- By default, the "-O" option in Bruno activates only the Cerenkov process
- In order for the Cerenkov process to work, you must have defined the refraction index for the radiator
- This must be done in a separated macro file (whose name is the argument of the "-O" flag).

Optical properties of materials

- It is important that optical properties are defined in a separated macro, and nowhere else
  - Optical properties defined in other macros are harmless, but will not have any effect
  - other commands may be dangerous if put in the optical property macros
  - the reason for this is that this macro is executed at a very early stage of Bruno initialization, and only the subset of commands related to optical properties is guaranteed to work.
    - make sure all your optical properties are defined in a separate macro
    - make sure that this macro does not contain any other commands
- Optical properties of surfaces and materials are defined in a specific G4 class (G4MaterialPropertiesTable). Bruno provides an interface to create G4MaterialPropertiesTables through macro files.
  - Examples on the SuperB simulation wiki

Optical surfaces

- Once created, optical photons undergo a special set of processes (absorption, reflection, refraction,...)
  - These require the optical properties of the surfaces surrounding a volume to be known, in addition to the properties of the bulk material
- Bruno has an interface to define such optical surfaces
  - Presently supports only G4LogicalSkinSurface
  - Support for G4LogicalBorderSurface can be added easily, if needed
  - Examples on the wiki

Supporting ongoing activities

- Supporting detector/bg studies is of paramount importance (for obvious reasons)
  - This means implementing the occasional extra feature, but also being sure that
    - The software stays flexible enough to accommodate most foreseeable scenarios
    - We should not succumb to haste and last-minute requests, and try not to implement suboptimal solutions (I have done it in the past...)
- I personally find it very interesting that Bruno is being used also for test-beam simulation
  - This is an unexpected use case
  - It may be helpful to have feedback concerning how useful Bruno has been, and what could be done to improve it

Packaging

- A recurring title in my slides
- No significant work since what I have presented at Elba last year
  - Now Eugenio has devised a new schema for gdml file distribution
  - Contacts with release team
    - try to commit what we have as it is now
    - then start with the process of synch-ing with the latest Bruno version

The longer term (1)

- We need to establish a policy for adoption of a new G4 release
- Steps may include
  - physics case
  - technical feasibility
  - prototype
  - validation
  - migration
- Validation is imvho by far the most critical one
  - We need a set of plots for each subdetector, with some standard physics observable, which can be easily (automatically) produced and compared to reference results
  - In addition, specific plots should clearly put in evidence that the originating physics case has been successfully addressed

The longer term (2)

- I believe we are hitting the limit of the standard G4 runtime configurability
  - Actually, I think we hit that limit quite some time ago, but implementation of the optical processes convinced me that it is time we move on
- The initialization sequence has become quite complicate, and the optical photons implementation showed that its steering needs more flexibility
- We need a powerful and user friendly “language” to define the configuration of a job (python, xml, perl, sh, …), and a better way to steer the job initialization
- My vote is for python, of course
  - However, I believe this is a decision to be taken at a higher level, and with a broader scope:
    - What job configuration patterns will SuperB software support?
    - Are we going to have a unique scripting language for all applications?
    - Which one?
- We should ask computing coordination to have an extensive discussion and reach some consensus as soon as possible
  - If this fails, I think we (FullSim) should move on anyway, following whatever path developers and users think is most appropriate

The longer term (3)

- A very interesting announcement made by R. Brun some weeks ago
- A project just started, whose goal is to prepare a new simulation toolkit
  - Plug the G4 physics within the functionality already provided bu ROOT
    - I/O
    - Interactivity
    - Math library
    - Geometry definition/navigation
- Should, eventually, become Geant5
- Status is of course still very very very very preliminary, but we are in the unique position to give a significant contribution to this project, at the very least by testing whatever is released
  - Our choice to use gdml for the geometry could make this relatively simple
  - Other “big” experiments (LHC) are probably too complicated, or too committed to G4 to really work on this
- Hard to understand what the development schedule will be, hence we really cannot plan anything as of now
  - I'll be keeping an eye on it, and we will decide what we want to do (if we want to do anything at all)
- The project has an official web site (very minimalistic) where you can find more info:
  - http://geant.cern.ch

Conclusions

- Bruno developments are mainly focused on supporting ongoing activities
  - Physics Recipes and optical photons are the latest news
  - More details in the SuperB simulation wiki
    - http://mailman.fe.infn.it/superbwiki/index.php/Geant4_SuperB_simulation_main_portal
- I hope we can move on with the packaging issue asap
- Following the G4 release schedule is an important and delicate task, which needs adequate validation procedures
- Bruno runtime configurability (and initialization steering) has hit the limits of the standard G4 mechanisms (macro/C++ files)
  - A better solution is needed
  - Would prefer this to be a SuperB-wide choice
- Newborn Geant5 project may be something interesting we want ot keep an eye on