

# Status of Generic Biasing

*Parallel 5B - Biasing & Channeling*

Fermilab Geant4 Collaboration Meeting  
29<sup>th</sup> September 2015

Marc Verderi  
LLR/Ecole polytechnique

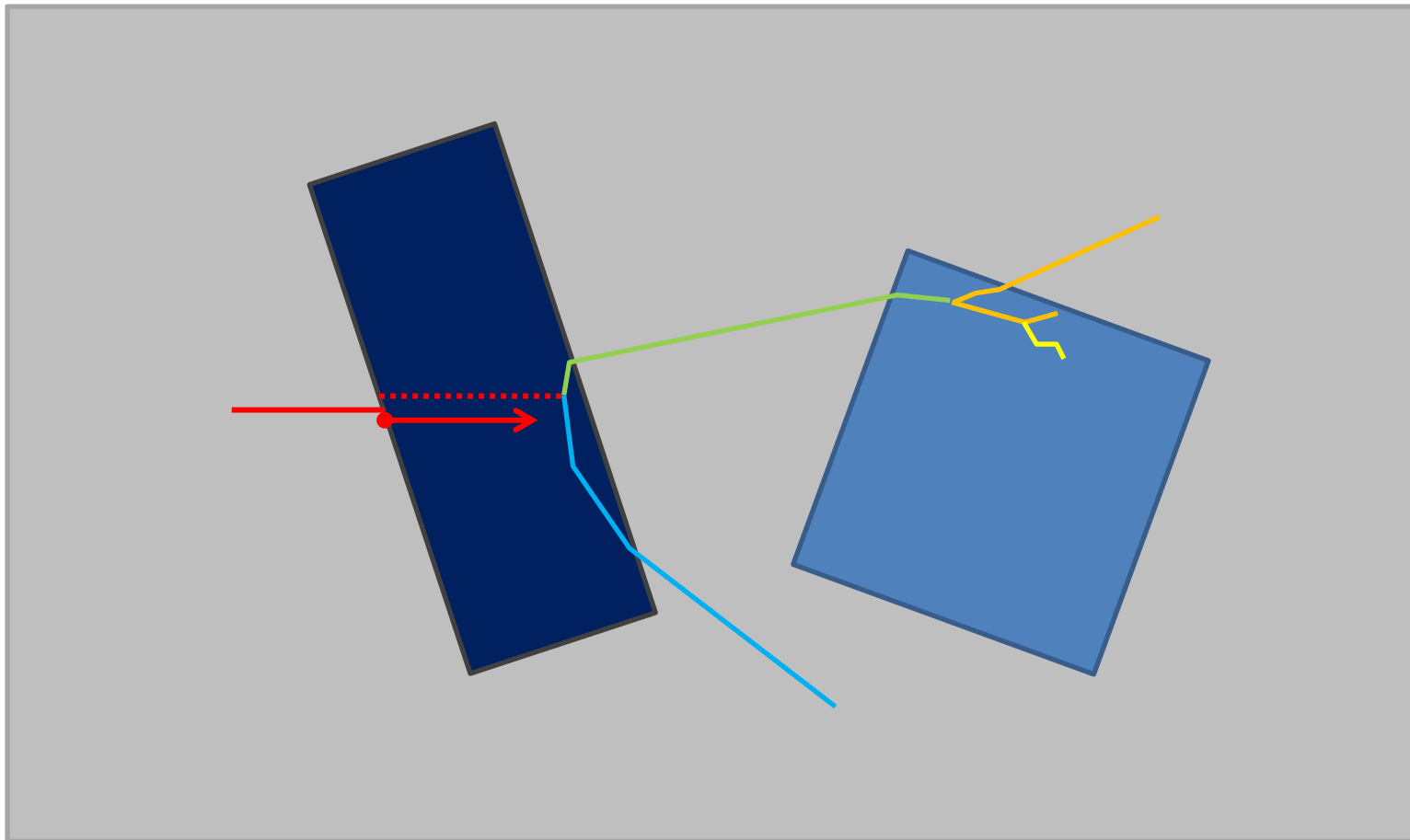
# Overview

- What's new:
  - Introduction of G4VAuxiliaryTrackInformation for robust track fate bookkeeping.
    - And related revision in generic biasing classes
  - [Note : G4VAuxiliaryTrackInformation might have many applications.]
- Still...:
  - The long list of items which need progress...

**WHAT'S NEW**

# Bookkeeping issue

- Need for correlated track fates
  - And for a safe way to keep related information
- Illustrated with the “force collision” scheme:



# G4VAuxiliaryTrackInformation

- Class to “add data members” to the G4Track
  - Several “clients” being able to add their own.
  - (generalization of G4VUserTrackInformation)
- Clients for G4VAuxiliaryTrackInformation:
  - Generate an ID per client
  - Mechanism to generate ID provided by the G4PhysicsModelCatalog.  
ChannelingID = G4PhysicsModelCatalog::Register(“channeling”);
    - Call made in process PreparePhysicsTable(...)
    - Ensure that ID’s will be common to threads.
- Class for extension is G4VAuxiliaryTrackInformation
  - In track category
- Track receives extra information by:  
track->SetAuxiliaryTrackInformation(someID, someInfo);
- Information got by:  
someInfo = track->GetAuxiliaryTrackInformation(someID);
- Functionality provided in 10.2-beta.

# Revision of Generic Biasing classes

- (processes/biasing/)management/**G4VBiasingOperator**:
  - Had methods to manage the information associated to track fate:
    - RememberSecondaries(...):
      - Used by force collision operator (generic/**G4BOptrForceCollision**) after “splitting” occurred
    - ForgetTrack(...):
      - Again used by force collision operator when biasing scheme is completed for current track
  - Remarks:
    - 1) Sort of methods that participate to making the class interface “fat”
      - Want to be “kind” providing utility methods...
      - ... but tend to generate loss of readability by mixing up important and minor things
    - 2) These methods did not need to be in the base class (in principle)
      - Are used by G4BOptrForceCollision only up to now
    - 3) Can now be dropped using instead in the proper places:
      - For RememberSecondaries(...):  
secondary->SetAuxiliaryTrackInformation(someBiasingID, someBiasingInfo);
      - ForgetTrack(...):  
track->RemoveAuxiliaryTrackInformation(someBiasingID);
  - Question:
    - Do we keep however RememberSecondaries(...) and ForgetTrack(...) for backward compatibility ?
      - I would prefer not... (would be again a kindness excess ;)

# Revision of Generic Biasing classes

- management/**G4BiasingTrackData**
  - Second victim of G4VAuxiliaryTrackInformation
  - Are the data kept for remembering fate to apply to track
    - But more have been put in this class, intended for kind of general usage.
    - Again used by generic/**G4BOptrForceCollision** only
  - This class should:
    - Inherit from G4VAuxiliaryTrackInformation
    - Be moved from management -> generic
    - Be renamed to make clear it corresponds to **G4BOptrForceCollision**
    - All together : sounds like a disappearance ;) and replacement
- Same question:
  - Do we care about backward compatibility in this case ?
    - I would prefer not... (it is not a “major” user interface).
- Remark:
  - In the G4VAuxiliaryTrackInformation scheme, biasing operators may need their own track information data set
  - Have to allow them to request an ID to the G4PhysicsModelCatalog class at proper time:
    - Ie at the run initialization time
    - The G4BiasingProcessInterface will make the call to Operators

# Revision of Generic Biasing classes

- generic/**G4BiasingProcessInterface**:
  - BiasingTrackData deletion:
    - The process was taking care that the previous G4BiasingTrackData objects were deleted when:
      - The current track was ending
      - The fKillTrackAndSecondaries signal was used, deleting the objects for secondaries.
    - Could have been an expensive operation for a large application, as being called for *\*each\** track ending...
    - Now, this has simply to be dropped
      - And G4Track will take care of these deletions.
  - Additional call to operators:
    - As mentioned before, for requesting ID's to the G4PhysicsModelCatalog singleton
    - In the PreparePhysicsTable(...) method of the process
- All together, introduction of G4VAuxiliaryTrackInformation induces some amount of changes, but I believe that the generic biasing will gain:
  - In robustness
  - Readability
  - And maybe speed (less overhead).



**STILL...**

# Progress needed in:

- Items for 2015 work plan
- Improvements:
  - Allow/demonstrate use of parallel worlds (with generic biasing)
- Feasibility studies/prototyping:
  - Biasing of charged particles, with cross-section changing over the step
    - Formalism exists, not easy to apply
    - Might not be a problem in several concrete cases, where the physics processes actually don't take into account this change over a step
  - Use of occurrence biasing to allow continuous density change inside a same volume
    - Should be provided under the form of an example
  - DXTRAN-like biasing
    - Involve the uneasy question of biasing dependence upon other physics packages
    - Or of “generic classes” for final state description to avoid these dependences (a dream).
  - Material/isotope biasing
    - Option to enhance the contribution of an isotope as being the only one generating a measurable signal at the end
  - Woodcock tracking
    - A reimplementaion of a technique already present in GATE for example
- Testing:
  - Statistical test suite to verify correctness of biasing wrt to analog