Radioactive Decay Developments

Dennis Wright
Geant4 Collaboration Workshop
24 September 2013

Outline

- MT-related changes

- Other changes

- Coming changes

MT-related Changes (1)

- In version 9.6
  - whenever an unstable ion is created, check to see if it has a decay table
    - search by ion name string
  - if not, load instance of G4DecayTable with info from RDM database files
  - insert decay table back into the particle definition
    - modifies shared data (not allowed for MT)
  - each decay table is a collection of the possible decay channels (G4NuclearDecayChannel) for a given nuclide
  - all subsequent access of nuclide decay information is through the particle definition

MT-related Changes (2)

- Most recent tags (radioactive_decay-V09-06-13 or later)
  - decay tables now de-coupled from particle definition
    - implemented as std::map<G4int, G4DecayTable*>
      - int key is Z*10000 + A*10 + metastable level
      - search on name strings eliminated
    - two versions:
      - local: one instance per thread with its own set of decay tables
      - static: a central library of decay tables from all threads

MT-related Changes (3)

- registration of isotope table to G4IonTable moved from BuildPhysicsTable to ctor in G4RadioactiveDecay
  - under MT, ion secondaries were being lost because BuildPhysicsTable was not being called soon enough
- on-demand ion creation still supported

Non-MT Changes

- Integer metastable levels to replace energy levels
  - faster for level comparison and list indexing
  - plan to maintain both integer and energy arguments
  - consequences to user: for now in macro files use
    - gun/ionL with integer excitation levels instead of gun/ion
    - this is temporary until 10.0 comes out
- For now, default list of nuclides is created
  - particle/createAllIsomer must be added at top of macro
  - won’t need to do this in 10.0 and after

Non-MT Changes

- Set minimum lifetime to define meta-stable = 1 ns
  - ~500 more IT levels to be added to RDM database
  - these levels already exist in PhotonEvaporation database

- Databases updated by Laurent Desorgher to August 2012 (previous update 1998!)
  - RDM -> RadioactiveDecay4.0
  - Gammas -> PhotonEvaporation3.0
  - testing now underway (checking for missing files, levels)

Coming Changes - Problems

- The current G4RadioactiveDecay is a “big class” solution
  - long code, lots of methods, difficult to debug and maintain
  - performs both analogue and biased decay using if-blocks

- G4NuclearDecayChannel also a “big class” with 7 decay channels derived from it
  - K, L, M shell EC transitions treated as separate classes

- Internal conversion not done correctly
  - only one gamma may be emitted when IC turned on
  - can leave nucleus in state for which there is no data
  - done so that shell index for IC probability can be calculated

Coming Changes - Solutions

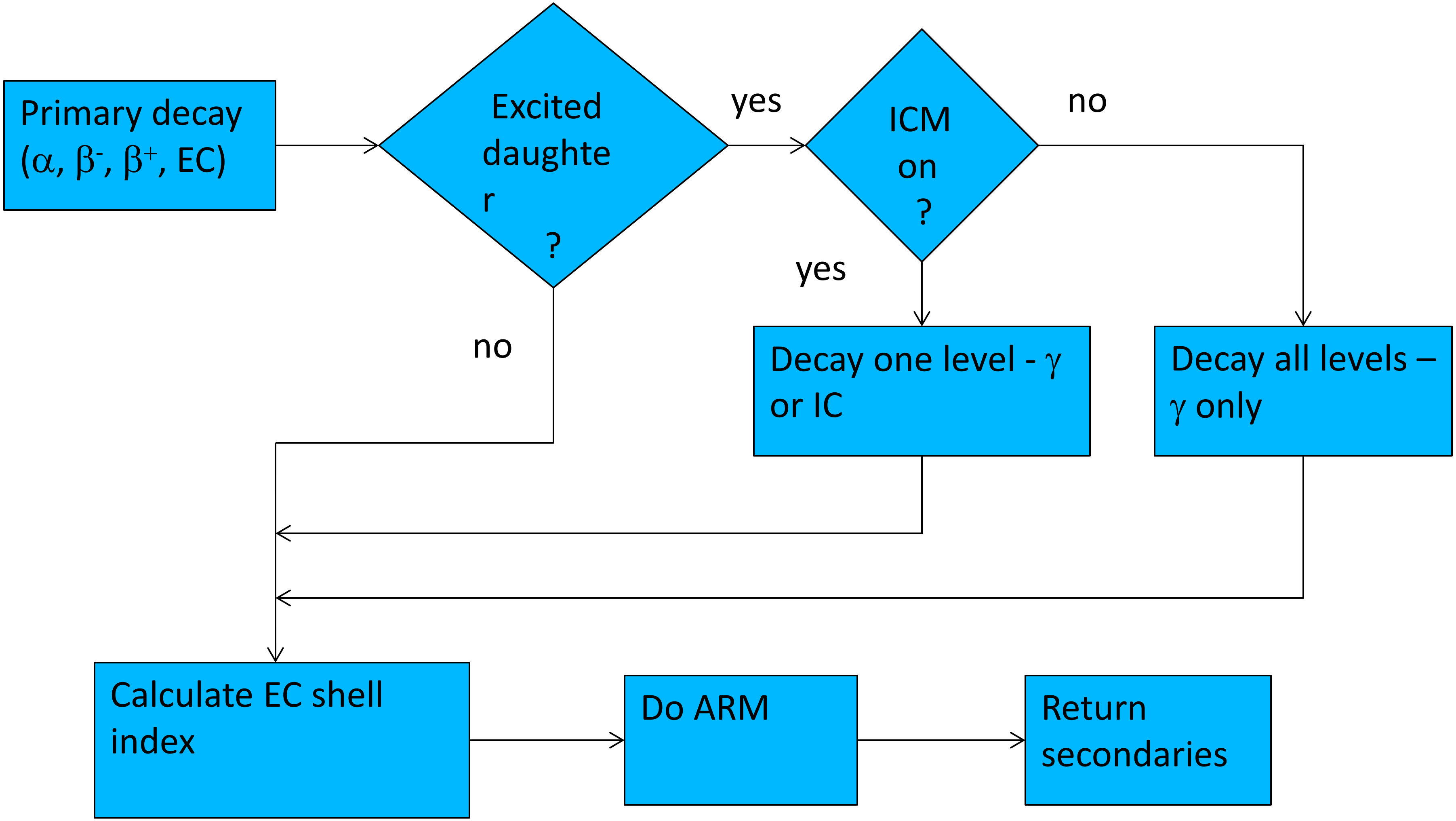
- Re-design required
  - make RDM smaller, split analogue and biased functionality into two classes
  - combine K, L, M EC transition classes into one
  - redo internal conversion to allow all eligible gamma/IC transitions to take place – leave IC on all the time

- Some good ideas for re-design in recent paper by Hauf and Kuster et al.
  - can use some of them

Backup Slides

-

Flow Chart for Past Implementation of G4NuclearDecayChannel


Primary decay (α, β^-, β^+, EC)
Excited daughte r ?
yes
ICM on ?
no
yes
no
Decay one level - γ or IC
Decay all levels – γ only
Calculate EC shell index
Do ARM
Return secondaries