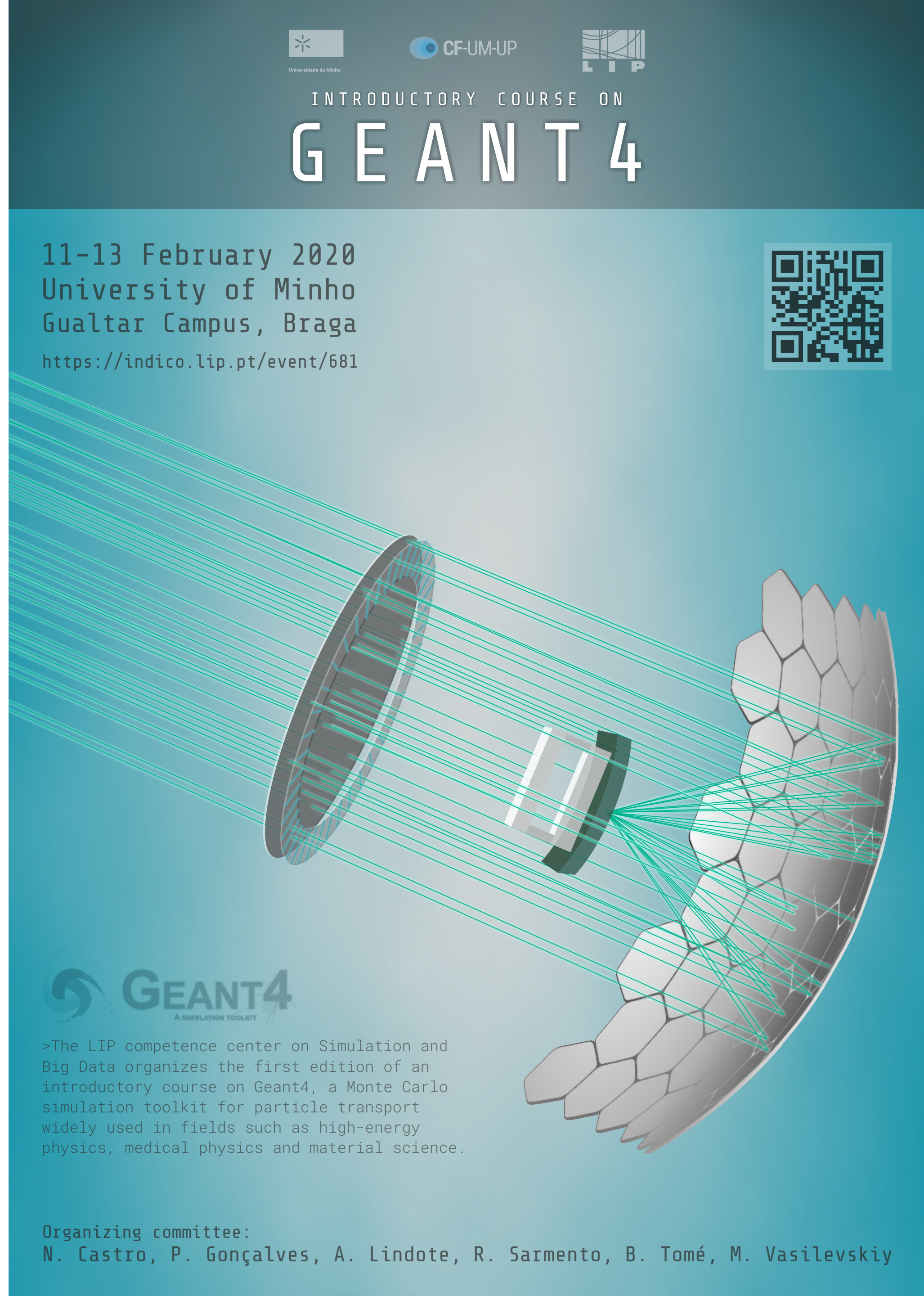
Optical Photon Processes

Bernardo Tomé

Slides adapted from slides produced by :
Marc Verderi,Dennis Wright, Vladimir Ivantchenko, Mihaly Novak
http://cern.ch/geant4



Optical processes

- Propagation of optical photons and their interaction with materials is treated separately from regular electromagnetic processes :
  - Wavelengths are much larger than atomic spacing
  - Treated (partially) as waves; no smooth transition to gammas
  - Energy/momentum not generally conserved in G4 optics !

- Optical photons produced directly by three processes :
  - `G4Cerenkov`
  - `G4Scintillation`
  - `G4TransitionRadiation`

http://geant4-userdoc.web.cern.ch/geant4-userdoc/UsersGuides/ForApplicationDeveloper/html/TrackingAndPhysics/physicsProcess.html#optical-photon-processes

Optical photon transport

- Optical photons can undergo:
  - Refraction and reflection at boundaries
  - Wavelength shifting
  - Bulk absorption
  - Rayleigh scattering

- Material **optical** properties can be specified in a **`G4MaterialsPropertiesTable`** attached to **`G4Material`**
  - transmission efficiency, dielectric constants, surface properties
  - including binned wavelength/energy dependences

- Spectral properties can be also specified in **`G4MaterialsPropertiesTable`**
  - scintillation yield, time structure (fast, slow components)

Note : an optical photon will not be propagated (killed) in a material without index of refraction

Specification of material optical properties

```
const G4int NUMENTRIES = 3;

G4double ppckov[NUMENTRIES]     = {2.034*eV, 3.*eV, 4.136*eV};
G4double rindex[NUMENTRIES]     = {1.3435, 1.351, 1.3608};
G4double absorption[NUMENTRIES] = {344.8*cm, 850.*cm, 1450.0*cm};

G4MaterialPropertiesTable* MPT = new G4MaterialPropertiesTable();

MPT->AddConstProperty("SCINTILLATIONYIELD",100./MeV);

MPT->AddProperty("RINDEX",ppckov,rindex,NUMENTRIES}->SetSpline(true);
MPT->AddProperty("ABSLENGTH",ppckov,absorption,NUMENTRIES}->SetSpline(true);

scintillator -> SetMaterialPropertiesTable(MPT);
```

in Geant4 Application Developers Manual

Optical boundary processes

- Refraction and reflection handled by **`G4OpBoundaryProcess`**

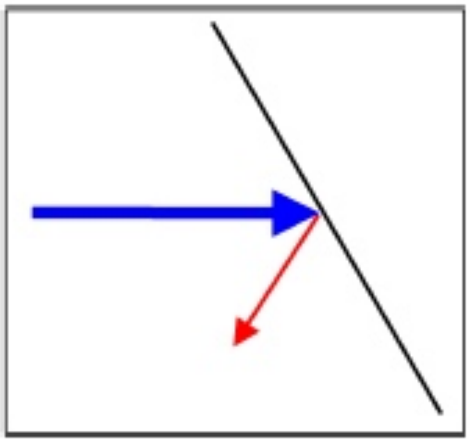
- Boundary properties
  - dielectric-dielectric
  - dielectric-metal
  - dielectric-black material

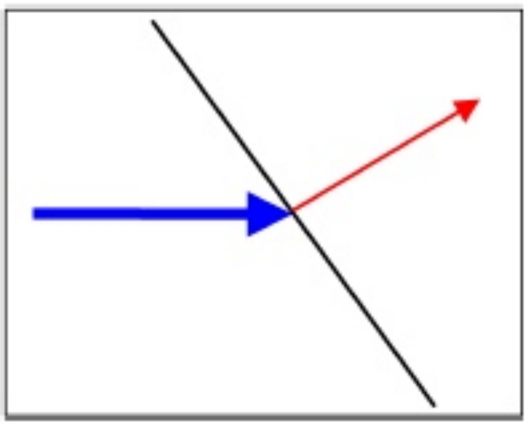
- Surface properties
  - polished
  - ground
  - front- or back- painted

User must supply surface properties through `G4OpticalSurfaceModel`

Reflection and Refraction

- Geant4 reflects “particle-like” behavior – no “splitting” of tracks





- Probability for transmission/reflection computed from Fresnel’s relations

- Interference effects, important in layers of thickness comparable with the wavelength, can only be accounted for by the empirical reflectivity factor for the surface

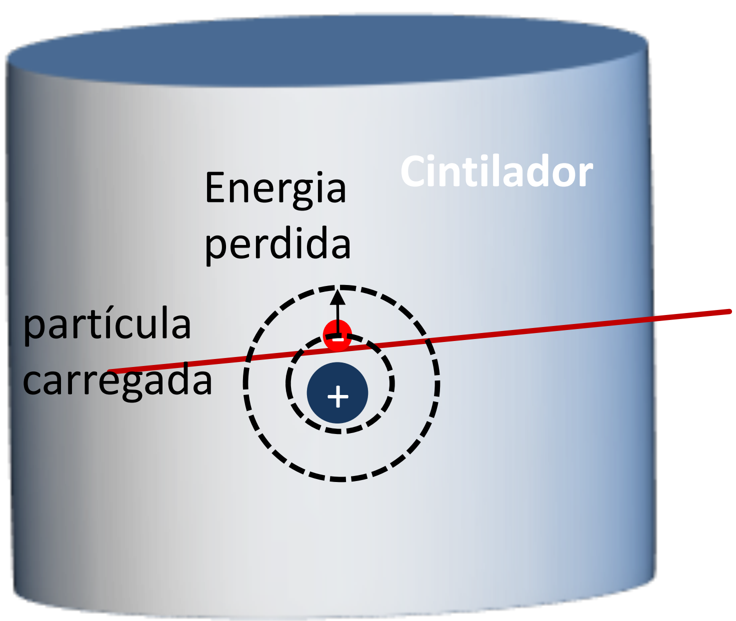
Optical bulk processes

- `G4OpAbsorption`
  - uses photon attenuation length from material properties to get mean free path
  - photon is simply killed after a selected path length
- `G4OpRayleigh`
  - elastic scattering including polarization of initial and final photons
  - builds its own physics table (for mean free path) using **`G4MaterialTable`**
  - may only be used for optical photons (a different process provided for gammas)

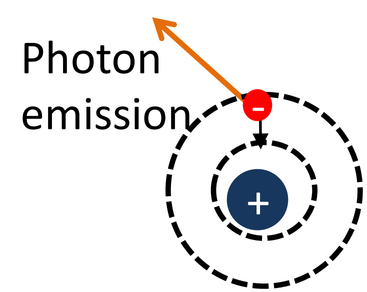
Optical processes : Scintillation

- In a scintillator material light is emitted when excitation energy is transferred to the material
  - isotropic emission
  - random linear polarization

t0



t0 + Δt



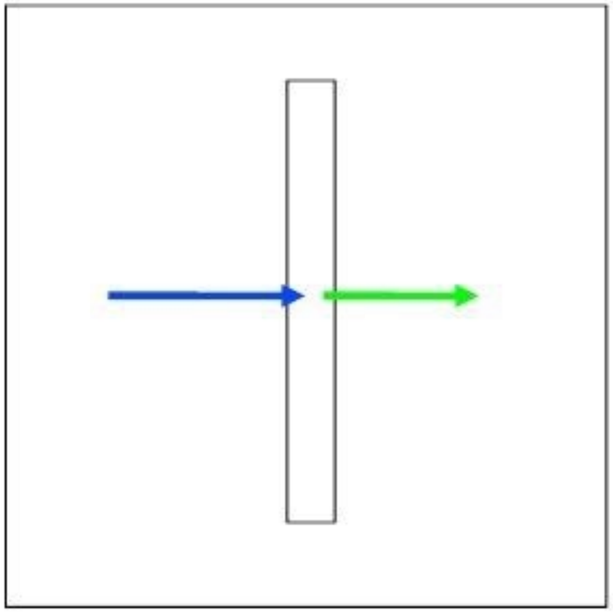
Specification of scintillation properties

```
const G4int NUMENTRIES = 9;
G4double Scnt_PP[NUMENTRIES] = { 6.6*eV, 6.7*eV, 6.8*eV, 6.9*eV,
                                 7.0*eV, 7.1*eV, 7.2*eV, 7.3*eV, 7.4*eV };

G4double Scnt_FAST[NUMENTRIES] = { 0.000134, 0.004432, 0.053991, 0.241971,
                                   0.398942, 0.000134, 0.004432, 0.053991,
                                   0.241971 };
G4double Scnt_SLOW[NUMENTRIES] = { 0.000010, 0.000020, 0.000030, 0.004000,
                                   0.008000, 0.005000, 0.020000, 0.001000,
                                   0.000010 };

G4Material* Scnt;
G4MaterialPropertiesTable* Scnt_MPT = new G4MaterialPropertiesTable();

Scnt_MPT->AddProperty("FASTCOMPONENT", Scnt_PP, Scnt_FAST, NUMENTRIES);
Scnt_MPT->AddProperty("SLOWCOMPONENT", Scnt_PP, Scnt_SLOW, NUMENTRIES);

Scnt_MPT->AddConstProperty("SCINTILLATIONYIELD", 5000./MeV);
Scnt_MPT->AddConstProperty("RESOLUTIONSCALE", 2.0);
Scnt_MPT->AddConstProperty("FASTTIMECONSTANT",  1.*ns);
Scnt_MPT->AddConstProperty("SLOWTIMECONSTANT", 10.*ns);
Scnt_MPT->AddConstProperty("YIELDRATIO", 0.8);

Scnt->SetMaterialPropertiesTable(Scnt_MPT);
```

in Geant4 Application Developers Manual

Optical processes : Wavelength Shifting

- In a WLS material light is absorbed and reemitted with a longer wavelength
  - isotropic emission
  - random linear polarization
- Handled by `G4OpWLS` process
  - initial photon is killed, one with new wavelength is created
- `User must supply :`
  - absorption length in function of photon energy
  - emission spectra parameters as function of energy
  - time delay between absorption and re-emission



Specification of WLS properties

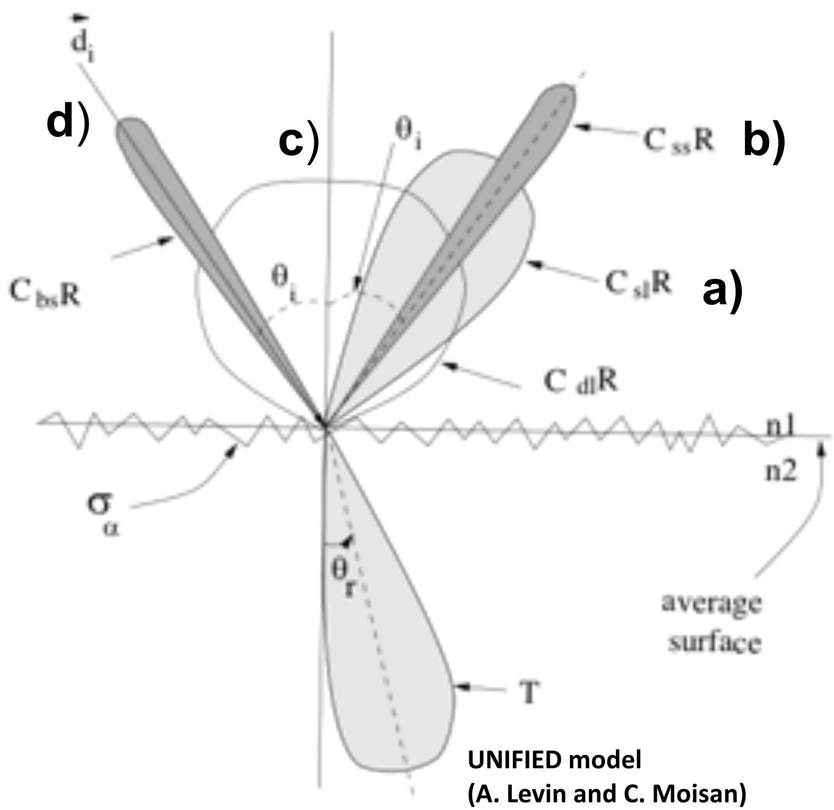
```
const G4int nEntries = 9;

G4double PhotonEnergy[nEntries] = { 6.6*eV, 6.7*eV, 6.8*eV, 6.9*eV,
                                    7.0*eV, 7.1*eV, 7.2*eV, 7.3*eV, 7.4*eV };

G4double RIndexFiber[nEntries] =
          { 1.60, 1.60, 1.60, 1.60, 1.60, 1.60, 1.60, 1.60, 1.60 };
G4double AbsFiber[nEntries] =
          {0.1*mm,0.2*mm,0.3*mm,0.4*cm,1.0*cm,10*cm,1.0*m,10.0*m,10.0*m};
G4double EmissionFiber[nEntries] =
          {0.0, 0.0, 0.0, 0.1, 0.5, 1.0, 5.0, 10.0, 10.0 };

 G4Material* WLSFiber;
 G4MaterialPropertiesTable* MPTFiber = new G4MaterialPropertiesTable();

 MPTFiber->AddProperty("RINDEX",PhotonEnergy,RIndexFiber,nEntries);
 MPTFiber->AddProperty("WLSABSLENGTH",PhotonEnergy,AbsFiber,nEntries);
 MPTFiber->AddProperty("WLSCOMPONENT",PhotonEnergy,EmissionFiber,nEntries);
 MPTFiber->AddConstProperty("WLSTIMECONSTANT", 0.5*ns);

 WLSFiber->SetMaterialPropertiesTable(MPTFiber);
```

Boundary processes :The Unified model

- Applies to dielectric-dielectric and dielectric-metal interfaces and tries to provide a realistic simulation, dealing with many aspects of surface finish and reflector coating :
  - surface may be assumed as smooth and covered with a metalized coating representing a specular reflector with given reflection coefficient
  - painted with a diffuse reflecting material where Lambertian reflection occurs.
  - surfaces may or may not be in optical contact with another component
  - a rough surface made up of micro-facets with normal vectors that follow given distributions around the nominal normal for the volume at the impact point.

Boundary processes :The Unified model

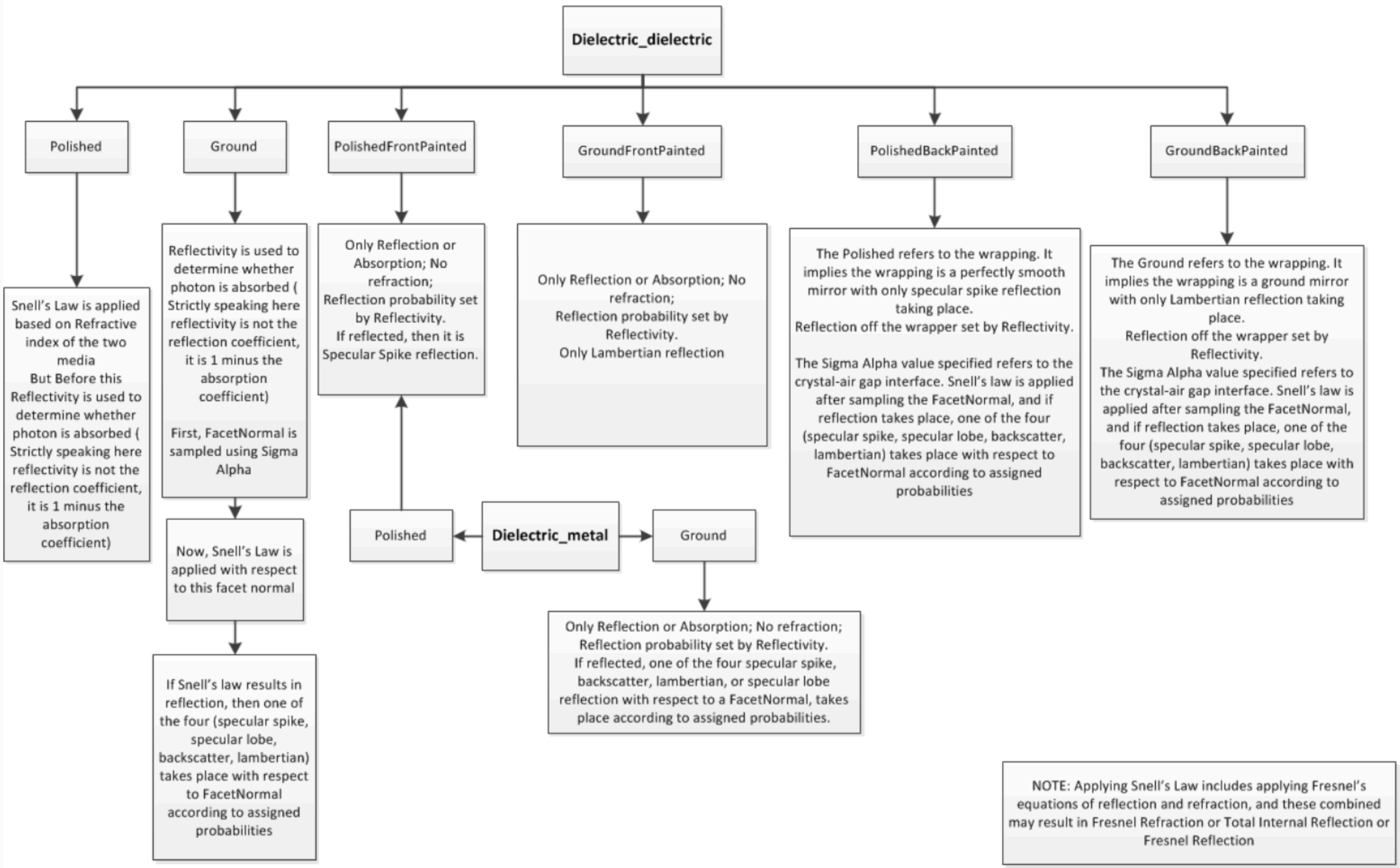
Provides for a range of different reflection mechanisms.



UNIFIED model
(A. Levin and C. Moisan)

a) specular lobe constant : reflection probability about the normal of a micro facet
b) specular spike constant : the probability of reflection about the average surface normal.
c) diffuse lobe constant is for the probability of internal Lambertian reflection
d) back-scatter spike constant : several reflections within a deep groove with the ultimate result of exact back-scattering

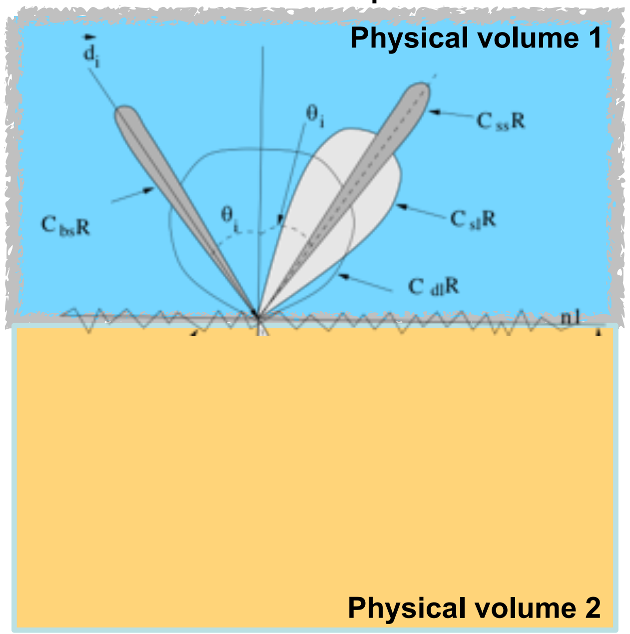
Surface finishes



http://geant4-userdoc.web.cern.ch/geant4-userdoc/UsersGuides/ForApplicationDeveloper/html/TrackingAndPhysics/physicsProcess.html#optical-photon-processes

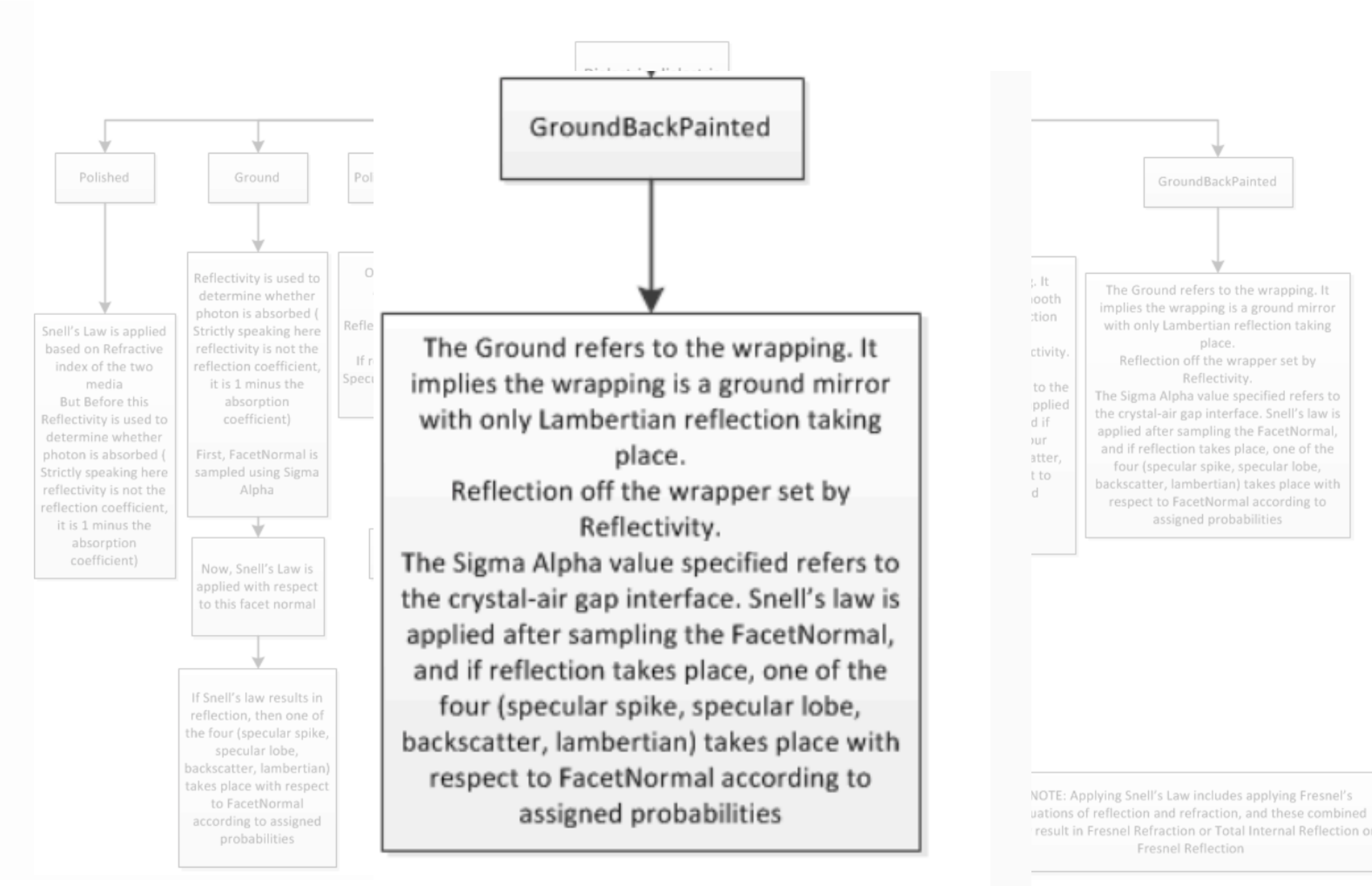
Example of specification of dielectric-dielectric surface properties

Ground backpainted




```
G4VPhysicalVolume* volume1;
G4VPhysicalVolume* volume2;

G4OpticalSurface* OpSurface = new G4OpticalSurface("name");

G4LogicalBorderSurface* Surface = new
  G4LogicalBorderSurface("name",volume1,volume2,OpSurface);

G4double sigma_alpha = 0.1;

OpSurface->SetType(dielectric_dielectric);
OpSurface->SetModel(unified);
OpSurface->SetFinish(groundbackpainted);
OpSurface->SetSigmaAlpha(sigma_alpha);

const G4int NUM = 2;

G4double pp[NUM] = {2.038*eV, 4.144*eV};
G4double specularlobe[NUM] = {0.3, 0.3};
G4double specularspike[NUM] = {0.2, 0.2};
G4double backscatter[NUM] = {0.1, 0.1};
G4double rindex[NUM] = {1.35, 1.40};
G4double reflectivity[NUM] = {0.3, 0.5};
G4double efficiency[NUM] = {0.8, 0.1};

G4MaterialPropertiesTable* SMPT = new G4MaterialPropertiesTable();

SMPT->AddProperty("RINDEX",pp,rindex,NUM);
SMPT->AddProperty("SPECULARLOBECONSTANT",pp,specularlobe,NUM);
SMPT->AddProperty("SPECULARSPIKECONSTANT",pp,specularspike,NUM);
SMPT->AddProperty("BACKSCATTERCONSTANT",pp,backscatter,NUM);
SMPT->AddProperty("REFLECTIVITY",pp,reflectivity,NUM);
SMPT->AddProperty("EFFICIENCY",pp,efficiency,NUM);

OpSurface->SetMaterialPropertiesTable(SMPT);
```

Surface finishes



in Geant4 Application Developers Manual

Hands-On
Optical Physics Processes

- **We will use a new branch where part of the needed code is already inserted;**

- **Start by committing the modifications you inserted :**

```
git commit -am "My new code"
```

- **Fetch the new branch :**

```
git checkout step6-optics-0
```

Let's make the crystal shine

- We are going to use the personalised modular physics list (class **`PhysicsList`**) to include the optical physics processes category; modify the code as appropriate to use this class and include the optical physics category;
- In the **`DetectorConstruction()`** inspect the function **`SetMaterialOpticalProperties()`**; it defines the optical properties table to be set to the NaI crystal,
- Choose a light yield such that a few (~ 10 - 100) photons will be emitted for the typical energies of the radioactive sources being simulated.
- Run the simulation with the visualisation and the macro **`gamma.mac`**
- For a small number of gammas shot, you should see several photon tracks inside the crystal; if yes proceed to the next slide

Detecting the optical photons

We will assume that a photon is detected if it arrives at the back of the crystal :

- In the **`SteppingAction`** class check if a step occurred inside your volume of interest and if the particle is an **`opticalphoton;`**
- Additionally use the **`Pre`** or **`PostStep`** points as needed to check if the photon detection condition is met;
- Increment a variable in the EventAction to count the number of detected photons; kill the optical photon afterwards;
- At the end of the event, write the total number of detected photons (together with the already existing true and smeared energies) to the output file;

- Increase the crystal light yield (L.Y.) to about 3 photons / keV (of the order of the NaI L.Y. (1 photon / 26 eV) times a 10% overall efficiency;

Detecting the optical photons

- Run the program for a few thousands of events using **`gamma.mac`**;
- Plot the distribution for the number of detected photons;
- What do you think about the result ?

Improving the light collection

We will now wrap the NaI crystal with a white diffuser

- In the **`DetectorConstruction`** class the function **`SetSurfaceOpticalProperties()`** is used to set the optical properties of the wrapping material;

- At the end of the **`Construct()`** function you can find the placeholders to use the **`G4LogicalBorderSurface constructor`** to set the optical properties of the surface of the crystal;

Detecting the optical photons

- Run the program for a few thousands of events using **`gamma.mac`**;
- Plot the distribution for the number of detected photons;
- Try to fit a gaussian distribution to the total absorption peak and extract the resolution of our spectroscopy setup;
- Now run the macros with the isotopes and compare the results with what was obtained yesterday using the true energy deposition;