Physics processes
Software process and OOAD
At the end of May, Sebastién, Ziad and me started a working group for the development of low energy processes in the water medium.

- Goal of the working group was the development of step-by-step electromagnetic interactions for:
  - Electrons between 7 eV and 10 keV
  - Protons and alpha particles between 1 keV and 10 MeV

- The processes considered are:
  - Elastic scattering (relevant only for electrons)
  - Excitation of liquid water molecules
  - Ionisation of liquid water molecules
Software process

- The development of this set of processes has adopted a rigorous software process
  - Requirements
    - Vision document
    - User requirement document
      - Both available in the Geant4 documentation CVS repository
  - Analysis and design
    - Design model
      - Available in the Geant4 documentation CVS repository
  - Implementation
    - Developed code is available in Geant4 CVS repository
  - Testing
    - Unit test
    - Validation (will be planned)
  - Traceability (in progress)
Elastic scattering

- Step by step model
- Relevant only for electrons
- Analytical total cross section between 7.5 eV and 10 keV
- Two analytical angular distribution models
  - Brenner model between 7.5 and 200 eV
  - Emfietzoglou model between 200 eV and 10 keV
- Electrons below 7.5 eV must be killed and energy must be deposited (UR)

- Water molecule tracking after interaction is neglected

\[ e^- + H_2O \rightarrow e^- + H_2O \]

<table>
<thead>
<tr>
<th>e^- Total cross section</th>
<th>e^- Angular distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 eV</td>
<td>7.5 eV</td>
</tr>
<tr>
<td>Energy deposit</td>
<td>Rutherford + screening factor</td>
</tr>
</tbody>
</table>

Riccardo Capra
Ionisation processes

- Step by step model
- Six incoming particles considered
  - e\(^-\), p, H, α\(^{++}\), α\(^+\), He
- 5 different ionisation levels
  - Energy deposit depends on the ionisation level
- Total cross sections (for each ionisation level) are tabulated
  - Tables are precalculated and stored in files
- Angular distribution of the outgoing electron is analytical
  - Each incoming particle has a different model
- Energy distribution of the outgoing electron is
  - Analytical for protons below 300 keV
  - Based on tables for protons above 300 keV and other particles
    - Tables for outgoing electrons must interpolate on two parameters
      - Energy of the incoming particle
      - Energy of the outgoing electron
- Ionized water molecule tracking after interaction is neglected

\[ p + H_2O \rightarrow p + H_2O^+ + e^- + \text{Energy Deposit} \]
Electrons exchange

- Step by step model
- Three incoming particles considered
  - $p, \alpha^{++}, \alpha^+$
  - $\alpha^{++}$ can strip one or two electrons to the water molecule
- Total cross sections
  - Analytical for protons
  - Tables for alphas
- Energy of the outgoing particle and the energy deposit depend on the incoming particle and on the number of electrons stripped
- Direction changes are neglected
- Ionized water molecule tracking after interaction is neglected

Three incoming particles considered
- $H, \alpha^+, He$
  - $\alpha$ can release one or two electrons
- Tabulated total cross section
- Energy of the outgoing particle and the energy deposit depend on the incoming particle and on the number of electrons stripped
- Direction changes are neglected for both particles and electrons
- Water molecule tracking after interaction is neglected

\[ p + H_2O \rightarrow H + H_2O^+ + \text{Energy Deposit} \]

\[ H + H_2O \rightarrow p + H_2O + e^- + \text{Energy Deposit} \]
Excitation

- Step by step model
- Six incoming particles considered
  - $e^-$, $p$, H, $\alpha^{++}$, $\alpha^+$, He
- Excited water molecules are not described
  - Excitation energy is deposited
- 5 different excitation levels
  - Energy deposit depends on the excitation level
- Total cross sections (for each excitation level)
  - Analytical for protons
  - Tables for other particles
- Direction changes are neglected

- Excited water molecule tracking after interaction is neglected
Constraint requirements

- **UR1**: The user shall be able to activate different processes in different geometry regions
  - Feature present in Geant4
- **UR2**: To reduce computation time, particles below a chosen cut-off energy could be dumped
- **UR3**: The user must be warned if processes are used on other materials than liquid water
- **UR4**: Cross section tables must be stored in a file in order to reduce the computation time
- **UR5**: Cross section tables must be interpolated
  - Feature present in Geant4 Low Energy package
The design presented in next slides heavily uses policy based classes

- This class structure was designed by me in the past weeks during the second design iteration of the package
  - The design iteration was necessary, as we discovered some new requirements that needed more flexibility
- The code implemented on this new design in mainly a rearrangement of what was done together with Sebastién and Ziad in May
Policy based classes

- Policy Based classes are parametrized classes that implement an algorithm delegating some operations to their class parameters, called policies
  - Policy based classes have something in common with Strategy Design Pattern

- But with some difference
  - Do not require that policies inherit from specific base classes
    - This means a weaker dependency of the policy and the policy based class from the policy interface. In complex situations this makes a design more flexible and open to extensions
  - They are compile bound
    - In other words there is no need of virtual methods, resulting in faster execution code and classes that allocate less memory
Different algorithms will be able to use same policies. The same algorithm will be able to use different policies.

- **Testable process**
- **Water processes algorithms (policy based)**
- **Total cross section policies**
- **Angular distribution policies**
- **Analytical models**
- **Tables-based models**
- **1-dimensional interpolated datasets (cross sections)**
- **2-dimensional interpolated datasets (energy distributions for ionisation)**
- **Data files**
- **Files tables generation**
- **Electrons charged Ions Manager**
DNA Processes design

- **G4VLowEnergyTestableDiscreteProcess**
  - For unit testing purposes access to the protected `GetMeanFreePath` method is needed
  - All the classes inheriting from this class could be tested with a common unit test

- **G4VDNAProcessInWater**
  - Policy based class
  - Verifies if the medium is water
  - Calculates water atomic density

```cpp
G4VDNAProcessInWater(processName : const G4String &)  
G4VDNAProcessInWater();
•<virtual> PostStepDoIt(aStep : const G4Step &, aTrack : const G4Track &): G4ParticleChange *
•<virtual> BuildPhysicsTable(aParticleDefinition : const G4ParticleDefinition &): void
•<virtual> IsApplicable(aParticleDefinition : const G4ParticleDefinition &): G4bool
•<virtual> GetMeanFreePath(aTrack : const G4Track & , previousStepSize : G4double, condition : G4ForceCondition *): G4double
•<virtual> ValidateInWater(aTrack : const G4Track &)
G4VDNAProcessInWater();
•operator=(right : const G4VDNAProcessInWater &); G4VDNAProcessInWater&
```

```cpp
G4VLowEnergyTestableDiscreteProcess  
(from processes_management)
•G4VDiscreteProcess()
•G4VDiscreteProcess()  
•<virtual> PostStepGetPhysicalInteractionLength()
•<virtual> PostStepDoIt()
•<virtual> AlongStepGetPhysicalInteractionLength()  
•<virtual> ARestGetPhysicalInteractionLength()
•<virtual> ARestDoIt()
•<virtual> AlongStepDoIt()  
•<abstract> GetMeanFreePath()
G4VDiscreteProcess()
operator=()
```

```cpp
G4VDNAProcessInWater  
(from processes_management)
•G4VLowEnergyTestableDiscreteProcess()
•operator=(right : const G4VLowEnergyTestableDiscreteProcess &); G4VLowEnergyTestableDiscreteProcess&
```

---

Riccardo Capra
**G4VDNAProcessInWater**

- **G4VDNAProcessInWater** inherits from
  - G4VLowEnergyTestableDiscreteProcess
  - TotalCrossSectionPolicy (parameter)
    - *G4VDNAProcessInWater* uses the total cross section provided by this class in order to evaluate the mean free path in the water medium
    - The class must provide a `BuildTotalCrossSection` method and the incoming particle definition
  - FinalStatesPolicy (parameter)
    - At this level only generic methods (`BuildFinalStatesData`, `KillIncomingParticle`) are requested
    - Further constraints on this policy are set by classes inheriting from G4VDNAProcessInWater
Elastic scattering process

- **G4DNAElectronElasticScatteringInWater**
  - Inherits from G4VDNAProcessInWater
  - Requires an additional constraint to the FinalStatesPolicy
    - That policy must provide a RandomizeCosTheta method to obtain the \( \cos(\theta) \) variable distributed according to the considered model
  - Implements the PostStepDoIt
  - We want stress the following features
    - Generic implementation
      - Only one electron as outgoing particle
      - Uniform distribution along \( \varphi \)
      - No energy deposit
    - The model-dependent details are left to the policies
      - Total cross section
      - \( \theta \) distribution
Elastic scattering policies

- To create concrete elastic scattering processes, policy classes must be provided
  - **G4DNARutherfordTotalCrossSectionPolicy**
    - Provides the Rutherford formula corrected with a screening factor
    - This class is parametrized with the energy limit of the process
  - **G4DNABrennerAngularDistributionPolicy**
    - Implements the angular distribution of the Brenner model
  - **G4DNAEmfietzoglouAngularDistributionPolicy**
    - Implements the angular distribution of the Emfietzoglou model
I developed the code to test all the processes inheriting from G4VLowEnergyTestableDiscreteProcess that do not generate secondaries

- Methods tested
  - IsApplicable
  - PostStepDoIt
  - GetMeanFreePath
- Energy range, number of events, incoming particle, process name and material can be selected by command line
- It can be easily generalized to processes with secondaries
- Other non-DNA discrete processes will profit of this unit test
  - In particular this unit test code has been used for the polarized Rayleigh testing
Unit test results

- Total cross section
  - Solid line: our model

- Angular distribution
**G4DNAIonizationInWater**

- Inherits from G4VDNAProcessInWater
- There are 5 ionisation levels
  - TotalCrossSectionPolicy must have an additional method to decide on which ionisation level the interaction happens
- The process requires additional constraint to the FinalStatesPolicy
  - The energy of the stripped electron
  - The direction of the stripped electron
- Direction of incoming particle after the interaction will be calculated by conservation law
Ionisation final states policies

- The energy distribution of the stripped electron is tabulated for several ionisation processes
  - But not for low energy protons ionisation
- A common policy for all these processes can be developed
  - It will be parametrized with the energy limits of the dataset
**G4DNAChargeDecreaseInWater**

- Inherits from G4VDNAProcessInWater
- $\alpha^{++}$ can go into $\alpha^+$ or He

**TotalCrossSectionPolicy** must have an additional method to decide how many electrons are stripped during the interaction

**The process requires an additional constraint to the FinalStatesPolicy**

- The electrons binding energy balance $(E_{\text{Water}} - E_{\text{Particle}})$

**Energy deposit, direction and energy of the outgoing particle are calculated in the PostStepDoIt method**
**G4DNAChargeDecreaseInWater**
- Inherits from G4VDNAProcessInWater
- He can go into $\alpha^+$ or $\alpha^{++}$
  - TotalCrossSectionPolicy must have an additional method to decide how many electrons are released during the interaction
- The process requires an additional constraint to the FinalStatesPolicy
  - The electrons binding energy
- Energy deposit, direction and energy of the outgoing particles are calculated in the PostStepDoIt method
Excitation

- **G4DNAIonizationInWater**
  - Inherits from G4VDNAProcessInWater
  - There are 5 excitation levels
    - TotalCrossSectionPolicy must have an additional method to decide on which ionisation level the interaction happens
    - The process requires additional constraint to the FinalStatesPolicy
      - The excitation energy of each one of the 5 levels
      - Direction of incoming particle after the interaction will be calculated by conservation law
EM data set upgrades

- DNA processes are based on tables based on theoretical calculations
  - In order to have an acceptable initialisation time, tables are calculated separately and stored into files
- Tables generation must be part of the Geant4 code
  - For this reason I have upgraded EM datasets introducing Save and Set methods (up to now Load and Get methods were enough)
    - I designed and implemented the upgrades on all the affected all EM datasets
    - I tested G4DNACrossSectionDataSet with a specifically developed unit test
Tabulated total cross sections

- Total cross section of all the processes is calculated from tables (except protons processes and elastic scattering)
- **G4DNATotalCrossSectionFromFilePolicy** provides this capability to all these processes

- The class is parametrised with the energy limits of the data table and the name of the data table file
  - I have implemented the class, but not tested yet
H, $\alpha^+$, He are ions not officially provided by Geant4

Singleton G4DNAGenericIonsManager will provide an unique instance of these ions for all processes (policies) that need them

<table>
<thead>
<tr>
<th>G4DNAGenericIonsManager</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;&lt;public, static&gt;&gt; Instance()</td>
</tr>
<tr>
<td>&lt;&lt;public&gt;&gt; GetIon(...)</td>
</tr>
</tbody>
</table>
Design road map

• Current processes design still miss two problem domains
  
  • A system of classes to manage bidimensional datasets and bidimensional interpolation algorithms
    
    • Current low energy design covers data interpolation from $\mathbb{R}^1$ to $\mathbb{R}^n$
    • Ionisation processes requires to extend it from $\mathbb{R}^2$ to $\mathbb{R}^n$
    • Other processes in the Low Energy package require a similar extension (Bremsstrahlung and Ionisation)
    
    • It would be nice to extend current data management scheme in order to manage interpolations from $\mathbb{R}^m$ to $\mathbb{R}^n$

• The system of classes the generates data tables
  
  • Data tables are based on theoretical calculations and the code to generate them should be provided within Geant4 according to Geant4 coding standards
  • Currently tables are generated by Sebastién and Ziad with private C code
Implementation road map

- Inelastic processes will be developed soon
  - We will profit of this meeting week to continue the development on some of the inelastic processes
    - Ionisation processes cannot be implemented until bidimensional dataset classes are not designed
  - After this meeting week Sebastién and Ziad will be mainly in charge for the processes development
  - I will continue covering all the technical aspects
    - Design, bidimensional interpolation implementation, generic ions singleton, unit test code upgrade, coding support and review, unit testing support and review, physics validation
Testing and validation overview

- Elastic processes has been tested
- 1D Data management classes has been tested
- With small extensions current Unit test can be used for all the processes we plan to develop
- We should start studying how to validate these processes
  - Should be nice to provide a validation of the processes that will be inserted in the next December release
Summary

Since May a lot of valuable work has been done

- Requirements
  - URD and VD are written and published on CVS

- Design
  - Covers almost all the problem domains we have to face
  - Published on CVS

- Implementation
  - Liquid water medium testing
    - Needed by all processes
  - Elastic processes
    - Brenner model
    - Emfietzoglou model

- Data management upgrades
  - G4VEMDataset
  - G4EMDataSet
  - G4CompositeDataSet
  - G4ShellDataset

- DNA specific data management
  - G4DNACrossSectionDataSet

- Tabulated total cross sections
  - Policy used by several inelastic processes

- Testing
  - Elastic processes unit test
    - Highly reusable unit test
  - G4DNACrossSectionDataSet unit test