A Geant4-based simulation of irradiation system for hadron therapy

Tsukasa Aso\textsuperscript{1,2}, A. Kimura\textsuperscript{2}, S. Kameoka\textsuperscript{3,2}, K. Murakami\textsuperscript{3,2}, T. Sasaki\textsuperscript{3,2}

\textsuperscript{1}Toyama National College of Maritime Technology
\textsuperscript{2}JST CREST
\textsuperscript{3}KEK

This research was supported by CREST, JST
Introduction

- Particle therapy facilities need “Simulation tools”.
  - Designing irradiation system according to facility specification.
  - Validation of treatment planning in different configuration.
- The “Simulation tool” has to allow users to setup their own irradiation system with minimum coding effort.
  - Usually, implementation of complex geometry is one of the issues.
  - Users want to concentrate on evaluation of physics results.

- Our Strategy for addressing to the requirement
  - A software toolkit for “common software parts” which is specially dedicated to particle therapy system.
    - We provide base/concrete classes for representing irradiation system.
      - (Reusability) The beam module classes may be utilized each other, because, in many case, same or similar geometry modules are used at different facilities.
      - (Extensibility) Users define their own beam modules on the top of the base class. It gives the user a guarantee that beam modules have basic functionalities.
  - The user can setup his/her own irradiation system geometry by combining those software parts.
  - Comparison of results becomes easier on the common framework using our software toolkit.
Overview of Design concept

Our simulation toolkit has three layers structure.

- **Geometry** represents the world volume of the irradiation system.
  - This is basically identical to `G4VUserDetectorConstruction`.

- **Particle Therapy System** represents a particular irradiation system. It consists of available beam modules at the facility.
  - i.e. HIBMC Gantry Nozzle, NCC Gantry Nozzle, etc.

- **Beam Module** represents individual beam module. It involves geometrical information.
  - i.e. Scatterer, Wobbler magnet etc.
**Defining Beam Module**

- We introduce the base class "G4MVBeamModule", where users define their own beam module geometry.
- This base class is responsible for handling the physical volume of the beam module.
  - Install(uninstall) the module in(from) the beam line.
  - Translate(rotate) the module in the beam line.
- The user may add new beam module classes on this base class.
  - The user has to implement two virtual functions.
    - `buildEnvelope()`
    - `buildNode()`.
  - The "Envelope" represents a master volume of the beam module, while the "Node" represents complex geometries inside the beam module.
Concrete class of Beam Module

- We have already implemented concrete classes of beam modules.
- These concrete classes are enough for describing
  - HIBMC gantry nozzle (Hyogo Ion Beam Medical Center)
  - NCC gantry nozzle (National Cancer Center)
  - NIRS experiment port for IHI (National Institute of Radiological Sciences)
Particle Therapy System

Defining Particle Therapy System

- We provide a base class "G4VParticleTherapySystem".
- The user must define their own particle therapy system on this base class. I.e. “HIBMCGantrySetup”, “NCCGantrySetup”, or “IHIPort” and so on.
- It mandates to implement three virtual methods.
  - `Setup()`
    - The user must register available beam modules in the particle therapy system.
  - `BuildDefault()`
    - At least, only the treatment room (world volume) has to be installed.
  - `UpdateEvent()`
    - The user must describe event by event action.
    - For example, this is a case for wobbling magnetic fields.

G4ParticleTherapySystemMessenger for manipulating beam modules.

- `/G4M/Module/install <Module Name>`
- `/G4M/Module/uninstall <Module Name>`
- `/G4M/Module/select <Module Name>`
  - `/G4M/Module/translate <X Y Z>`
  - `/G4M/Module/rotate <Ox Oy Oz>`
  - `/G4M/Module/typeid < Module’s parameter ID>`
Geometry Builder

- Geometry builder is responsible for selecting a particle therapy system.
  - We provide the base class "G4VGeometryBuilder".
  - It has a virtual method "SystemSelection()", where the particular particle therapy system is instantiated according to the given name.
  - The current particle therapy system object is obtained by static method, G4MVGeometryBuilder::GetSystem() . This is used for accessing functionalities of the particle therapy system and the beam modules.

- G4MGeometryMessenger
  - /G4M/System <PTSname> (Pre_init)
  - /G4M/ChangeSystem <PTSname> (Idle)

- At present, our implementation had realized following facilities.
  - HIBMCGantry (Hyogo Ion Beam Medical Center)
  - NCCGantry (National Cancer Center)
  - IHIPort (NIRS experimental port for IHI )
Demonstration

Contents

- Particle Therapy System Selection
  - HIBMCGantry, NCCGantry, IHIPort
- Installing / Uninstalling of Modules
- Translation / Rotation of Modules
Summary

- We have developed a simulation toolkit for irradiation system of particle therapy.
  - Beam components are modularized as software parts.
  - The toolkit is extended by adding new modules maintaining basic functionalities.
  - Our strategy is successfully applied for three particle therapy systems.
  - The physics validation using this toolkit comes soon.

- We are collecting geometry information of irradiation system.
  - It is welcome to give us your irradiation system geometry.
Parameter modification

Three type of modification

- `fSystem->GetModule("moduleName")`
  - Translation/Rotation
  - More functionality require cast to the module

- `fSystem->ApplyCommand("type");`
  - Catalogued parameter is loaded and rebuild the module geometry.

- `fSystem->UpdateEvent();`
  - Event by event modification such as wobbler field should be described.