Geant4 Introductory Course

Instituto de Estudos Avançados – Instituto Tecnológico de Aeronáutica,
São José Dos Campos, Brazil

July 28th – August 1st, 2014

User defined commands

J. M. Quesada, Universidad de Sevilla
Based on notes prepared by M. Asai (SLAC)

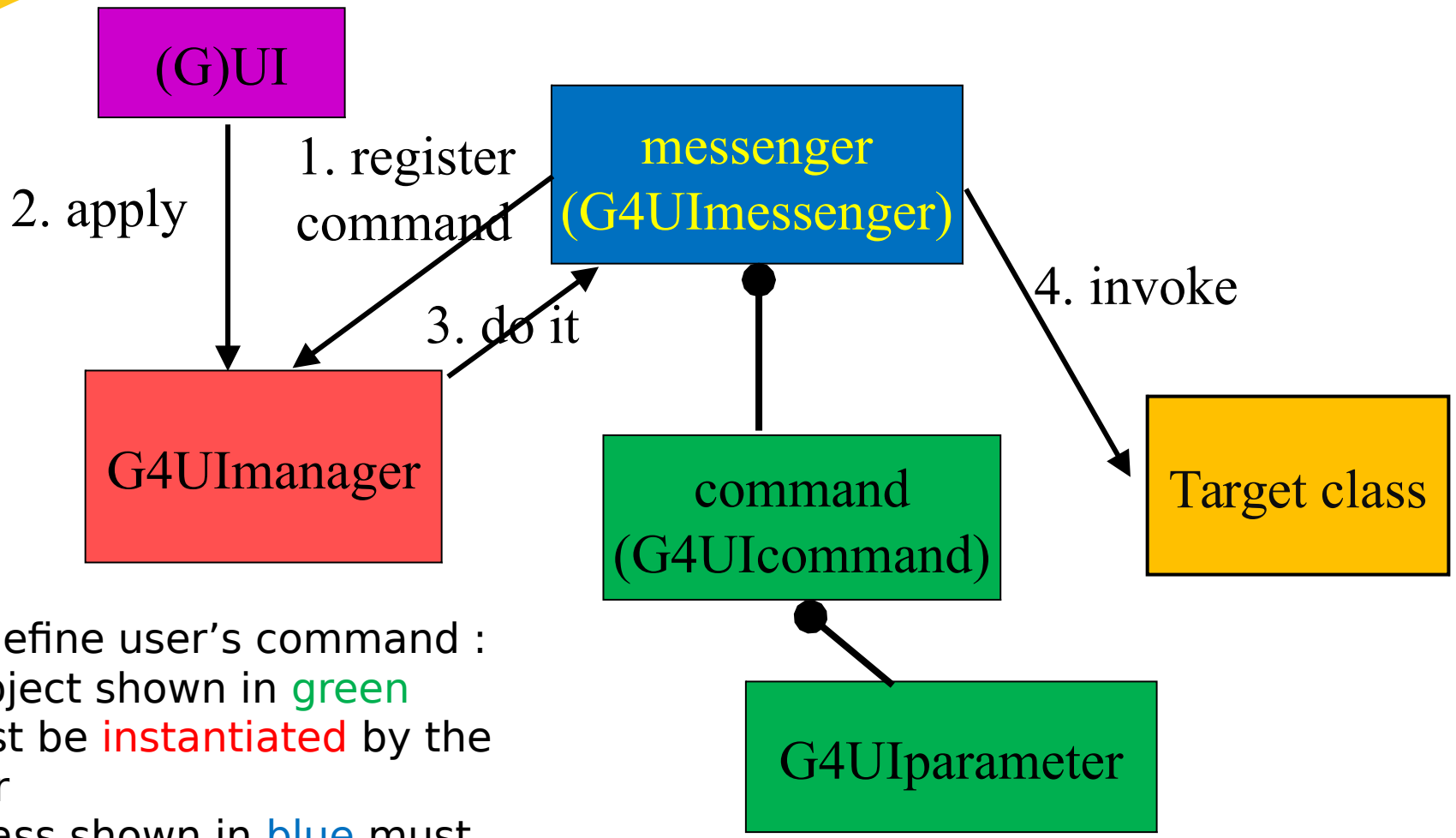
Geant 4

Contents

- Mechanism of UI command
- Defining basic UI command

Mechanism of UI command

Mechanism of UI command



To define user's command :
- Object shown in green must be instantiated by the user
- Class shown in blue must be implemented and instantiated by the user

Messenger class

- Each messenger class must be derived from G4UImessenger base class. A messenger class can handle more than one UI commands.
- A messenger class should be instantiated by the constructor of the target class to which commands should be delivered, and should be deleted by the destructor of the target class.
- Methods of messenger class
  - Constructor
    - Define (instantiate) commands / command directories
  - Destructor
    - Delete commands / command directories
  - void SetNewValue(G4UIcommand* command, G4String newValue)
    - Convert "newValue" parameter string to appropriate value(s) and invoke an appropriate method of the target class
    - This method is invoked when a command is issued.
  - G4String GetCurrentValue(G4UIcommand* command)
    - Access to an appropriate get-method of the target class and convert the current value(s) to a string
    - This method is invoked when the current value(s) of parameter(s) of a command is asked by (G)UI.

The deliverable classes (ready to be instantiated):

- G4UIcmdWithoutParameter
- G4UIcmdWithAString
- G4UIcmdWithABool
- G4UIcmdWithAnInteger
- G4UIcmdWithADouble
- G4UIcmdWithADoubleAndUnit
- G4UIcmdWith3Vector
- G4UIcmdWith3VectorAndUnit

Range, unit and candidates

void SetRange(const char* rangeString)

- Available for a command with numeric-type parameters.
- Range of parameter(s) must be given in C++ syntax.

aCmd->SetRange("x>0. && y>z && z<(x+y)");

- Not only comparison with hard-coded number but also comparison between variables and simple calculation are available.
- Names of variables must be defined by SetParameterName() method.

void SetDefaultUnit(const char* defUnit)

- Available for a command which takes unit.
- Once the default unit is defined, no other unit of different dimension will be accepted.
- Alternatively, you can define a dimension (unit category) without setting a default unit.

void SetUnitCategory(const char* unitCategory)

void SetCandidates(const char* candidateList)

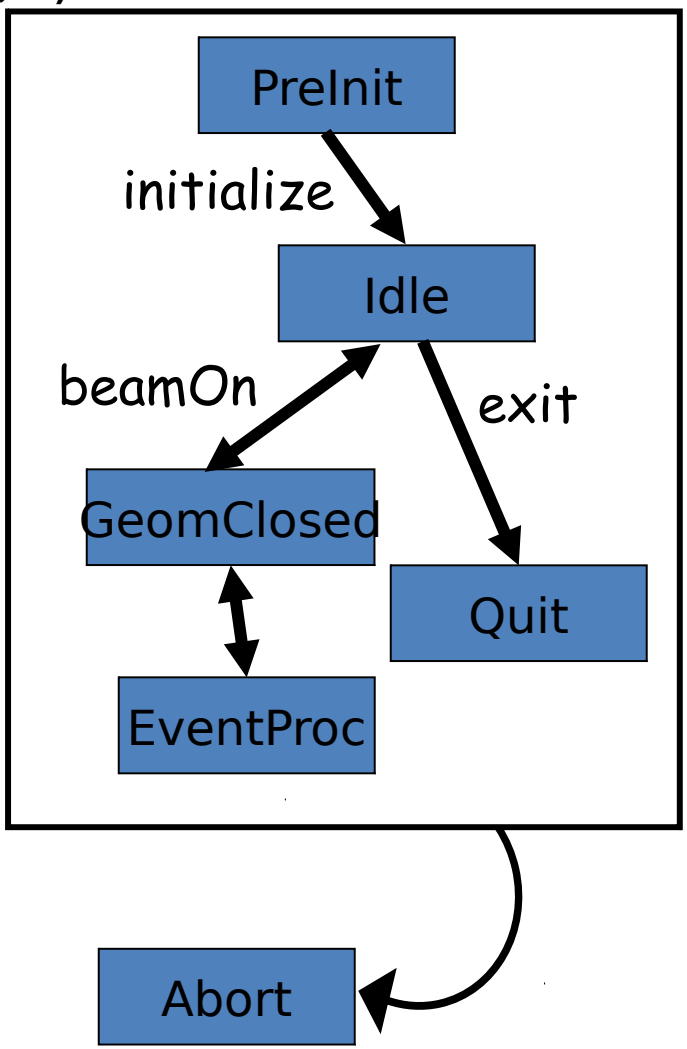
- Available for a command with string type parameter
- Candidates must be delimited by a space.
- Candidates can be dynamically updated.

Available state

void AvailableForStates(G4ApplicationState s1,…)

Define command's applicability for Geant4 application states.

- Geant4 has six application states.
  - G4State_PreInit
    - Material, Geometry, Particle and/or Physics Process need to be initialized
  - G4State_Idle
    - Ready to start a run
  - G4State_GeomClosed
  - Geometry is optimized and ready to
    - process an event
  - G4State_EventProc
    - An event is processing
  - G4State_Quit, G4State_Abort
    - UI command unavailable



Conversion between string and values

- Derivatives of G4UIcommand with numeric and boolean parameters have corresponding conversion methods.
- From a string to value

G4bool GetNewBoolValue(const char*)

G4int GetNewIntValue(const char*)

G4double GetNewDoubleValue(const char*)

G4ThreeVector GetNew3VectorValue(const char*)

  - To be used in SetNewValue() method in messenger.
  - Unit is taken into account automatically.

- From value to string

G4String ConvertToString(...)

G4String ConvertToString(...,const char* unit)

  - To be used in GetCurrentValue() method in messenger.

SetNewValue and GetCurrentValue


```
void A01DetectorConstMessenger
::SetNewValue(G4UIcommand* command,G4String newValue)
{
  if( command==armCmd )
  { target->SetArmAngle(armCmd->GetNewDoubleValue(newValue)); }
}

G4String A01DetectorConstMessenger
::GetCurrentValue(G4UIcommand* command)
{
  G4String cv;
  if( command==armCmd )
  { cv = armCmd->ConvertToString(target->GetArmAngle(),"deg"); }
  return cv;
}
```

PrimaryGeneratorMessenger.cc


```
[..]
PrimaryGeneratorMessenger::PrimaryGeneratorMessenger(PrimaryGeneratorAction* gun)
:m_action(gun)
{
  m_gunDir = new G4UIdirectory("/tutorial/gun/");
  m_gunDir->SetGuidance("PrimaryGenerator control");

  m_rndmCmd = new G4UIcmdWithAString("/tutorial/gun/rndm",this);
  m_rndmCmd->SetGuidance("Shoot randomly the incident particle.");
  m_rndmCmd->SetGuidance("Choice : on, off");
  m_rndmCmd->SetCandidates("on off");
  m_rndmCmd->SetParameterName("flag",false);
}

//....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo........oooOO0OOooo......

PrimaryGeneratorMessenger::~PrimaryGeneratorMessenger()
{
  delete m_rndmCmd;
}


//....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo........oooOO0OOooo......


void PrimaryGeneratorMessenger::SetNewValue(G4UIcommand* command, G4String newValue)
{
  if( command == m_rndmCmd )
   { m_action->SetRndmFlag(newValue);}
}


//....oooOO0OOooo........oooOO0OOooo........oooOO0OOooo........oooOO0OOooo.....
```

In PrimaryGeneratorAction ...

PrimaryGeneratorAction.hh:

```
 class PrimaryGeneratorAction : public G4VUserPrimaryGeneratorAction
{
[..]
    void SetRndmFlag (G4String flag) { m_rndmFlag = flag;};
[..]
    PrimaryGeneratorMessenger* m_gunMessenger;
    G4String                   m_rndmFlag;
[..]
}
```

PrimaryGenaratorAction.cc:

```
PrimaryGeneratorAction::PrimaryGeneratorAction(DetectorConstruction* det)
:m_detector(det)
{
[..]
  m_rndmFlag = "on";

  //create a messenger for this class
  m_gunMessenger = new PrimaryGeneratorMessenger(this);
}

//....oooOO0OOooo.......oooOO0OOooo.......oooOO0OOooo.......oooOO0OOooo.
.....

PrimaryGeneratorAction::~PrimaryGeneratorAction()
{
 [..]
  delete m_gunMessenger;
}
```

Messenger is instantiated

DetectorMessenger.cc


```
DetectorMessenger::DetectorMessenger(DetectorConstructio
n * Det)
:m_detector(Det)
{
  m_detDir = new G4UIdirectory("/tutorial/det/");
  m_detDir->SetGuidance("detector construction commands");


  m_materialCmd = new G4UIcmdWithAString
 ("/tutorial/det/spacecraftMaterial", this);
  m_materialCmd->SetGuidance("Select spacecraft
material");
  m_materialCmd->
SetParameterName("spacecraftMaterial",false);


  m_thicknessCmd = new
G4UIcmdWithADoubleAndUnit("/tutorial/det/spacecraftThickn
ess",this);
  m_thicknessCmd->SetGuidance("Select spacecraft
thickness");
m_thicknessCmd>SetParameterName("spacecraftThickness"
,false);
  m_thicknessCmd->SetRange("spacecraftThickness>0.");
  m_thicknessCmd->SetUnitCategory("Length");


  m_updateCmd = new
G4UIcmdWithoutParameter("/tutorial/det/update",this);
  m_updateCmd->SetGuidance("force to recompute
geometry.");
  m_updateCmd->SetGuidance("This command MUST be
applied before \"beamOn\" ");
  m_updateCmd->SetGuidance("if you changed geometrical
value(s).");
  m_updateCmd->AvailableForStates(G4State_Idle);
}
```

```
DetectorMessenger::~DetectorMessenger()
{
  delete m_materialCmd;
  delete m_thicknessCmd;
[..]
  delete m_updateCmd;

  delete m_detDir;
}


//....oooOO0OOooo.....oooOO0OOooo.....oooOO0

void DetectorMessenger::SetNewValue
(G4UIcommand* command,G4String newValue)
{
  if( command == m_materialCmd )
   { m_detector->SetSpacecraftMaterial(newValue);}

  if( command == m_thicknessCmd )
   { m_detector->SetSpacecraftThickness
(m_thicknessCmd->GetNewDoubleValue(newValue));}

if( command == m_updateCmd )
   { m_detector->UpdateGeometry();}
}
```

In DetectorConstruction ...

DetectorConstruction.hh:

```
class DetectorConstruction : public G4VUserDetectorConstruction
{
[..]
    void  SetSpacecraftMaterial(G4String);
[..]
void  SetSpacecraftThickness(G4double val) {m_spacecraftThickness = val;};
[..]
 void              UpdateGeometry();
```

DetectorConstruction.cc:

```
DetectorConstruction::DetectorConstruction()
{
   [..]
  // create commands for interactive definition of the detector
  m_detectorMessenger = new DetectorMessenger(this);
}
DetectorConstruction::~DetectorConstruction()
{delete m_detectorMessenger;  }
[..]
void DetectorConstruction::SetSpacecraftMaterial(G4String materialChoice)
{
  // search the material by its name
  G4Material* pttoMaterial = G4Material::GetMaterial(materialChoice);
  if (pttoMaterial) m_spacecraftMaterial = pttoMaterial;
}
[..]
void DetectorConstruction::UpdateGeometry()
{
  G4RunManager::GetRunManager()->DefineWorldVolume(ConstructVolumes());
}
```

Messenger is instantiated

And more possibilities (out of the scope of the present introduction):

- *Complicated* UI command
- G4GenericMessenger

Thanks for your attention

Backup slides

Defining complicated UI command

Complicated UI command

- Complicated UI command means a UI command with parameters which is not included in the deliverable classes.
  - G4UIcmdWithoutParameter, G4UIcmdWithAString, G4UIcmdWithABool, G4UIcmdWithAnInteger, G4UIcmdWithADouble, G4UIcmdWithADoubleAndUnit, G4UIcmdWith3Vector, G4UIcmdWith3VectorAndUnit
- A UI command with other type of parameters must be defined by G4UIcommand base class with G4UIparameter.
- G4UIparameter(const char * parName, char theType, G4bool theOmittable);
  - “parName" is the name of the parameter which will be used by the range checking and help
  - "theType" is the type of the parameter.
    - ‘b’ (boolean), ‘i’ (integer), ‘d’ (double), and ‘s’ (string)
  - Each parameter can take one line of guidance, a default value in case “theOmittable” is true, a range (for numeric type parameter), and a candidate list (for string type parameter).

Complicated UI command

- A G4UIcommand object can take arbitrary number of G4UIparameter objects.
  - Names of parameter must be different to each other (within the command).
  - It takes arbitrary number of guidance lines.
  - Availability for Geant4 states can be set.
  - In addition to ranges defined to individual parameters, it may take another range definition where values of more than one parameters can be compared to each other.

/gun/ion command


```
ionCmd = new G4UIcommand("/gun/ion",this);
ionCmd->SetGuidance("Set properties of ion to be generated.");
ionCmd->SetGuidance("[usage] /gun/ion Z A Q");
ionCmd->SetGuidance("        Z:(int) AtomicNumber");
ionCmd->SetGuidance("        A:(int) AtomicMass");
ionCmd->SetGuidance("        Q:(int) Charge of Ion (in unit of e)");
ionCmd->SetGuidance("        E:(double) Excitation energy (in keV)");

G4UIparameter* param;
param = new G4UIparameter("Z",'i',false);
ionCmd->SetParameter(param);
param = new G4UIparameter("A",'i',false);
ionCmd->SetParameter(param);
param = new G4UIparameter("Q",'i',true);
param->SetDefaultValue("0");
ionCmd->SetParameter(param);
param = new G4UIparameter("E",'d',true);
param->SetDefaultValue("0.0");
ionCmd->SetParameter(param);
```

Parameters are registered along their orders.

Converting string to values

- For complicated command, convenient conversion method is not available. Please use G4Tokenizer to tokenize the string and convert each token to numerical values.

```
SetNewValue(G4UIcommand * command,G4String newValues)
{    G4Tokenizer next( newValues );
fAtomicNumber = StoI(next());
fAtomicMass = StoI(next());
G4String sQ = next();
if (sQ.isNull()) {
fIonCharge = fAtomicNumber;
} else {
fIonCharge = StoI(sQ);
sQ = next();
if (sQ.isNull()) {
fIonExciteEnergy = 0.0;
} else {
fIonExciteEnergy = StoD(sQ) * keV;
}
}
…
```

G4GeneralMessenger

G4GenericMessenger

- G4GenericMessenger is a short-cut for defining simple UI commands that are
  - associated with one target class,
  - grouped in one command directory, and
  - dealing only with native variable types.
- G4GenericMessenger is a concrete class you can just instantiate without implementing a dedicated messenger class.

```
class G4GenericMessenger : public G4UImessenger
{
public:
 G4GenericMessenger(void* obj, const G4String& dir = "", const G4String& doc = "");
 Command& DeclareProperty
   (const G4String& name, const G4AnyType& variable, const G4String& doc = "");
 Command& DeclarePropertyWithUnit
   (const G4String& name, const G4String& defaultUnit,
                          const G4AnyType& variable, const G4String& doc = "");
 Command& DeclareMethod
    (const G4String& name, const G4AnyMethod& fun, const G4String& doc = "");
 Command& DeclareMethodWithUnit
    (const G4String& name, const G4String& defaultUnit,
                          const G4AnyMethod& fun, const G4String& doc = "");
```

Target class object

Command directory

Target class data member

Target class public set method

Command name

example


```
B5PrimaryGeneratorAction::B5PrimaryGeneratorAction()
{
   fMessenger = new G4GenericMessenger(this,
                        "/B5/generator/",  "Primary generator control");

   // momentum command
   G4GenericMessenger::Command& momentumCmd
     = fMessenger->DeclarePropertyWithUnit("momentum", "GeV", fMomentum,
                                                  "Mean momentum of primaries.");
   momentumCmd.SetParameterName("p", true);
   momentumCmd.SetRange("p>=0.");
   momentumCmd.SetDefaultValue("1.");
}

B5PrimaryGeneratorAction::~B5PrimaryGeneratorAction()
{
   delete fMessenger;
}
```