UI Command and Messenger

Geant4 Collaboration

KEK/CRC

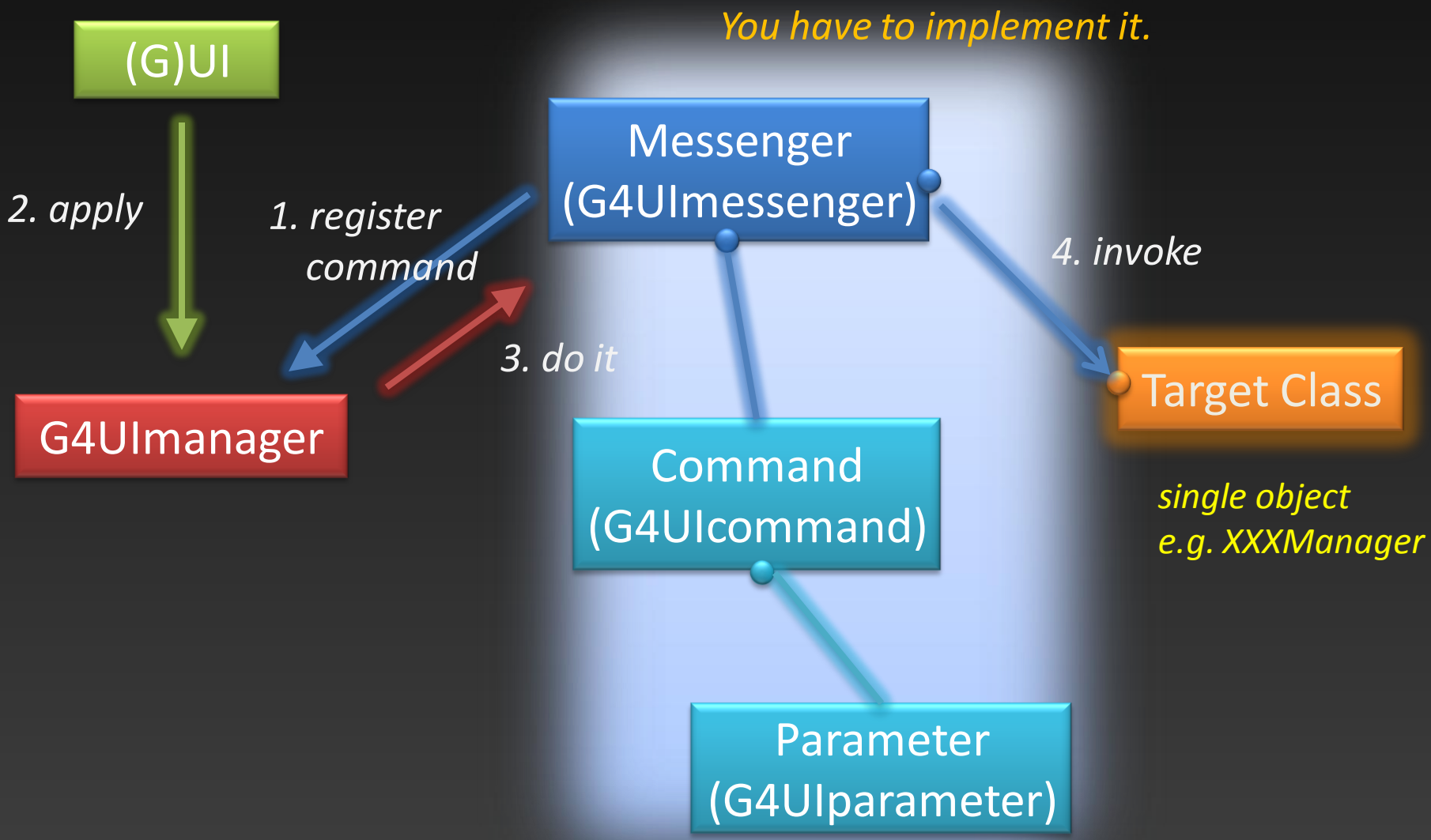
User-defined Commands

If built-in commands are not enough, you can make your own command.

Geant4 provides several command classes, all derived from *G4UIcommand*:

- *G4UIcmdWithoutParameter*
- *G4UIcmdWithABool*
- *G4UIcmdWithADouble*
- *and many more*
- See also Application Developers Guide, Chapter 7.2

Mechanism of UI command



Messenger class

You have to implement your UI commands which handle the corresponding G4 class.

- Each concrete messenger class is derived from *G4UImessenger* base class.
- A messenger class should be instantiated in the constructor of the target class.
- Target class should be *a single-object (singleton) class*
  - ✓ *e.g. XXXManager*

Implementation of your messenger classes

- Define command *directories* / *commands*.
- `void SetNewValue(G4UIcommand* command, G4String newValue)`
  - ✓ Convert "*newValue*" parameter string to appropriate value(s) and invoke a method of the target class
- `G4String GetCurrentValue(G4UIcommand* command)`
  - ✓ Access to a get-method of the target class and convert the current values to a string

An example of command definition

```
A01DetectorConstMessenger::A01DetectorConstMessenger
  (A01DetectorConstruction* tgt):target(tgt)
{
  mydetDir = new G4UIdirectory("/mydet/");
  mydetDir-> SetGuidance("A01 detector setup commands.");

  armCmd = new
   G4UIcmdWithADoubleAndUnit("/mydet/armAngle",this);
  armCmd-> SetGuidance("Rotation angle of the second arm.");
  armCmd-> SetParameterName("angle",true);
  armCmd-> SetRange("angle>=0. && angle<180.");
  armCmd-> SetDefaultValue(30.);
  armCmd-> SetDefaultUnit("deg");
}
```

G4UIcommand and its derivatives

G4UIcommand is a class which represents a UI command.
G4UIparameter represents a parameter.

Geant4 provides derived classes according to the types of associated parameters.

- *G4UIcmdWithoutParameter*
- *G4UIcmdWithAString*
- *G4UIcmdWithABool*
- *G4UIcmdWithAnInteger*
- *G4UIcmdWithADouble, G4UIcmdWithADoubleAndUnit*
- *G4UIcmdWith3Vector, G4UIcmdWith3VectorAndUnit*
- *G4UIdirectory*

A UI command with other type of parameters can be also defined.

Parameters

```
void SetParameterName(const char* parName,
                      G4bool omittable,
                G4bool currentAsDefault=false);
```

If "*omittable*" is true, the command can be issued without specifying a parameter value.

If "*currentAsDefalult*" is true, the current value of the parameter is used as a default value.

- The default value must be defined with *SetDefaultValue()* method.

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)");`
- 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.
- You can also define a dimension (*unit category*) without setting a default unit.
  - ✓ `void SetUnitCategory(const char* unitCategory)`

`void SetCandidates(const char* candidateList)`

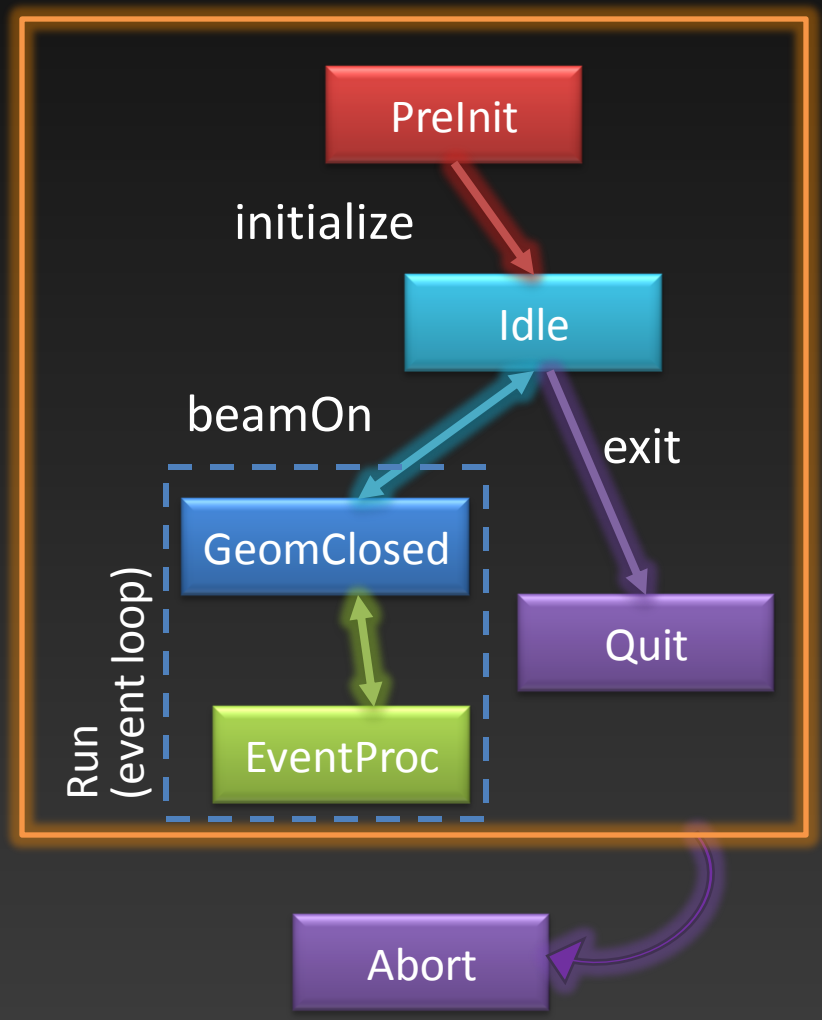
- Available for a command with string type parameter
- Candidates must be *delimited by a space*.

Available state

```
void AvailableForStates(
        G4ApplicationState s1,…)
```

Geant4 has *6 application states*.

- G4State_PreInit
  - ✓ Material, Geometry, Particle and/or Physics Process need to be initialized/defined
- 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
  - ✓ (Normal) termination
- G4State_Abort
  - ✓ A fatal exception occurred and program is aborting



Converting between string and values

Derivatives of *G4UIcommand* with numeric and boolean parameters have corresponding conversion methods.

From a string to value

- used in *SetNewValue()* method in a messenger
- *G4bool GetNewBoolValue(const char*)*
- *G4int GetNewIntValue(const char*)*
- *G4double GetNewDoubleValue(const char*)*
- *G4ThreeVector GetNew3VectorValue(const char*)*
- Unit is taken into account automatically.

From a value to string

- used in *GetCurrentValue()* method in a messenger
- *G4String ConvertToString(...)*
- *G4String ConvertToString(...,const char* unit)*

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;
}
```

Complicated UI command

"*Complicated*" UI command means a UI command with parameters which is not provided as the delivered classes.

A UI command with other type of parameters can be directly defined by *G4UIcommand* and *G4UIparameter*.

- `G4UIparameter(const char * parName,`
  `char theType,`
  `G4bool theOmittable);`
- "*theType*" is the type of the parameter.
  - ✓ 'b' (boolean), 'i' (integer), 'd' (double), and 's' (string)
- Each *parameter* can take *guidance*, a default value in case "*theOmittable*" is true, parameter range , and a candidate list .
- Parameters can be added to a command by G4UIcommand::SetParameter(G4UIparameter* const)