UI command

Makoto Asai (SLAC Computing Services)

Geant4 Tutorial Course @ Fermi Lab

October 27th, 2003

Geant4

Contents

- Alias and loop
- Mechanism of UI command
- Messenger class
- Defining a command
- Implementing a messenger

Alias

- Alias can be defined by
  `/control/alias [name] [value]`
  - It is also set with **`/control/loop`** and **`/control/foreach`** commands
  - Aliased value is always treated as a string even if it contains only numbers.
- Alias is to be used with other UI command.
  - Use curly brackets, { and }.
  - For example, frequently used lengthy command can be shortened by aliasing.

```
/control/alias trv1 "/tracking/verbose 1"
{trv1}
```

  - Aliases can be used recursively.

```
/control/alias file1 /diskA/dirX/fileXX.dat
/control/alias file2 /diskB/dirY/fileYY.dat
/control/alias run 1
/myCmd/getFile {file{run}}
```

Loop

- **`/control/loop`** and **`/control/foreach`** commands execute a macro file more than once. Aliased variable can be used inside the macro file.
- **`/control/loop [`*`macroFile`*`] [`*`counterName`*`] [`*`initialValue`*`] [`*`finalValue`*`] [`*`stepSize`*`]**
  - *`counterName`* is aliased to the number as a loop counter
- **`/control/foreach [`*`macroFile`*`] [`*`counterName`*`] [`*`valueList`*`]**
  - *`counterName`* is aliased to a value in *`valueList`*
  - *`valueList`* must be enclosed by double quotes (" ")

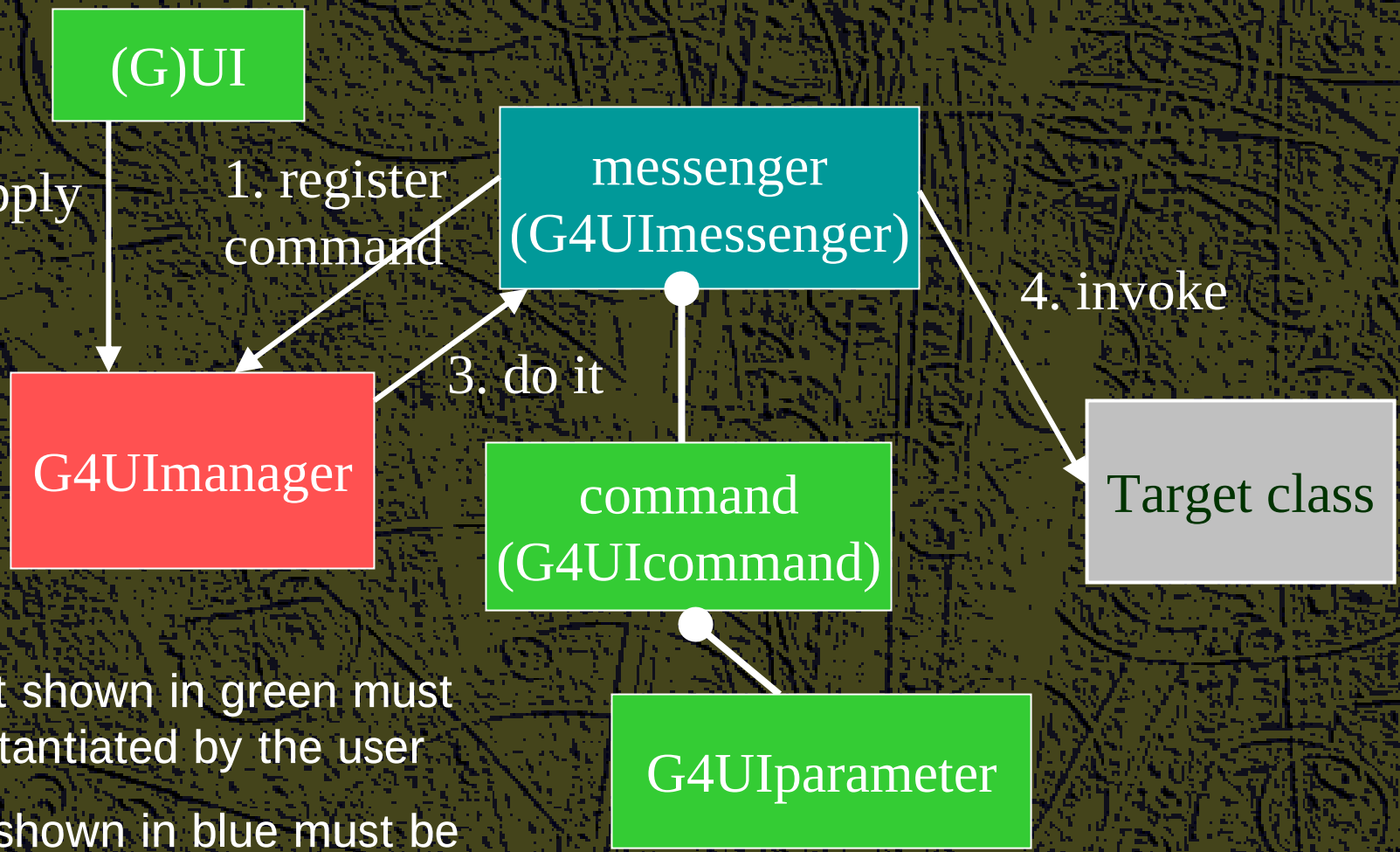
- on UI terminal or other macro file

```
/control/loop myRun.mac Ekin 10. 20. 2.
```

- in myRun.mac

```
/gun/energy {Ekin} GeV
/run/beamOn 100
```

Mechanism of UI 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 one or more 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 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

Definition (instantiation) of a command

- To be implemented in the constructor of a messenger class.

```
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");
}
```

- Guidance can (should) be more than one lines. The first line is utilized as a short description of the command.

G4UIcommand and its derivatives

- G4UIcommand is a class which represent a UI command. G4UIparameter represents a parameter.
- G4UIcommand can be directly used for a UI command. Geant4 provides its derivatives according to the types of associated parameters.
  - G4UIcmdWithoutParameter
  - G4UIcmdWithAString
  - G4UIcmdWithABool
  - G4UIcmdWithAnInteger
  - G4UIcmdWithADouble, G4UIcmdWithADoubleAndUnit
  - G4UIcmdWith3Vector, G4UIcmdWith3VectorAndUnit
  - G4UIdirectory
  - A UI command with other type of parameters must be defined by G4UIcommand base class.

Parameter name(s)

```
void SetParameterName(
   const char*parName,
   G4bool omittable,
   G4bool currentAsDefault=false);
void SetParameterName(
   const char*nam1, const char*nam2, const char*nam3,
   G4bool omittable,
   G4bool currentAsDefault=false);
```

- Parameter names are used in held, and also in the definition of parameter range.
- If "omittable" is true, the command can be issued without this particular parameter value.
- If "currentAsDefalult" is true, current value of the parameter is used as a default value, otherwise 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)");
```

- 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)
```

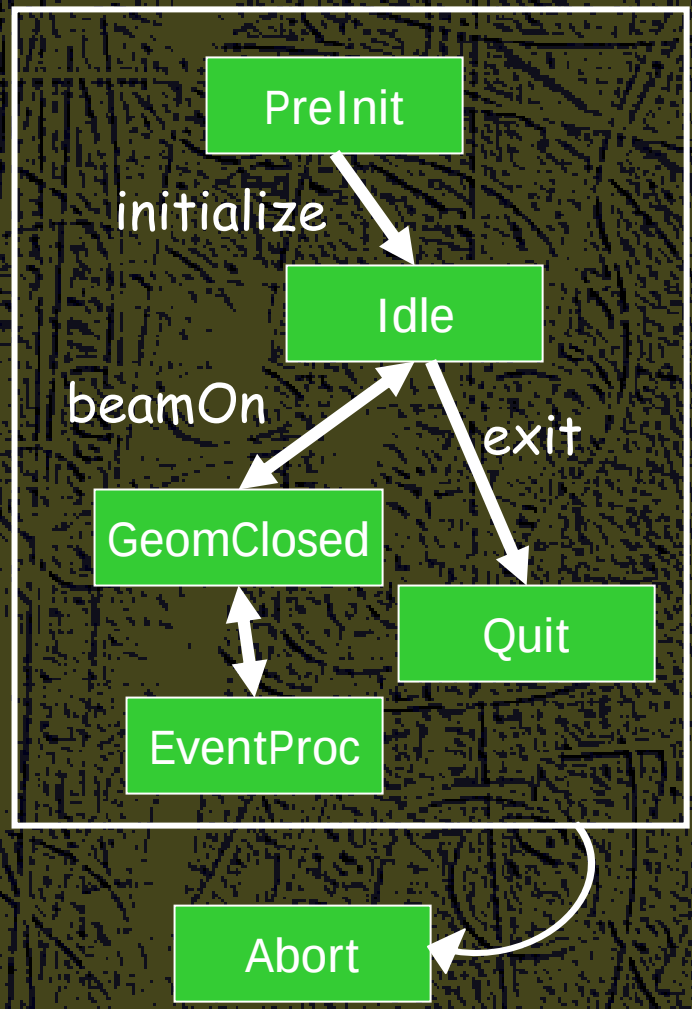
`void SetCandidates(const char* candidateList)`

- 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



Converting 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;
}
```