



Closer Integration of Analysis In Geant4 Framework

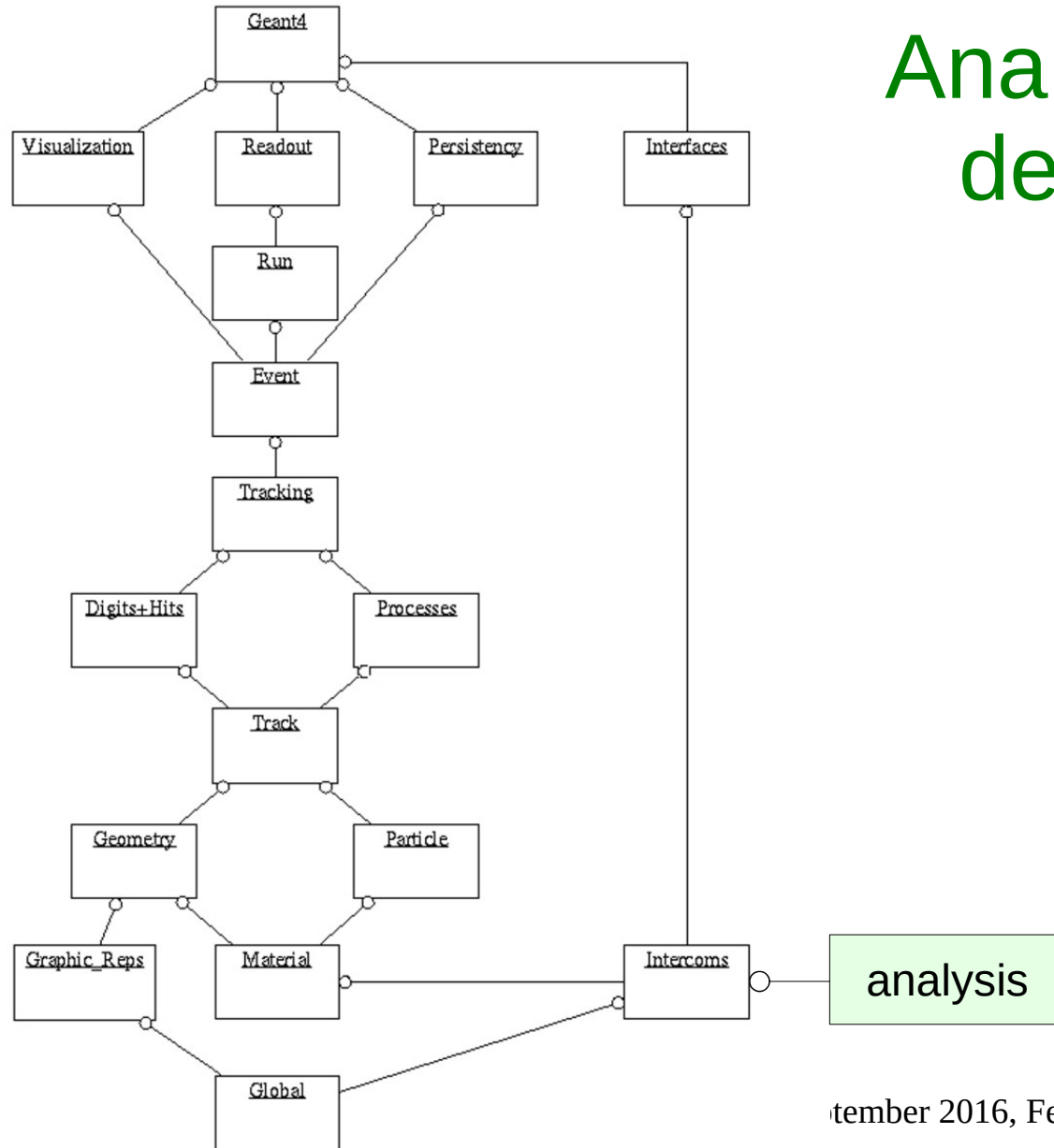
I. Hrivnacova, IPN Orsay (CNRS/IN2P3)

21st Geant4 Collaboration Meeting,
13 September 2016, Ferrara

Outline

- Analysis category dependencies
- Parameters & G4RunManager
- Analysis & Scoring

Analysis category dependencies



Parameters & G4RunManager

Handling Parameters

- The classes for users parameters management were added in 10.2 release for the purpose of simplification of users application code.
- First implementation following the requirements from Luciano Pandola (Advanced Examples WG) and Michel Maire (responsible of B1 and B3 examples)
- Their usage is demonstrated in the basic examples B1 and B3a.
- Further integration in the Geant4 framework is under discussion since their introduction

B1 example

Geant4 10.1.

```
class B1Run : public G4Run {
public:
    ...
    // method from the base class
    virtual void Merge(const G4Run*);
    void AddEdep (G4double edep);
    // ...
private:
    G4double    fEdep;
    G4double    fEdep2;
};
```

Geant4 10.2, 10.3.

```
#include "G4Parameter.hh"
...
class B1RunAction : public G4UserRunAction {
public:
    ...
    // method from the base class
    void AddEdep (G4double edep);
    // ...
private:
    G4Parameter<G4double>    fEdep;
    G4Parameter<G4double>    fEdep2;
};
```

*Run class and
Merge() method
are not needed*

B1 example (cont.)

Geant4 10.2.

```
#include "G4ParameterManager.hh"
```

```
...  
B1RunAction::B1RunAction()  
: G4UserRunAction(),  
  fEdep("Edep", 0.),  
  fEdep2("Edep2", 0.)
```

*The parameters are initialized
with a name and a value*

```
{  
  //Register parameter to the parameter manager  
  G4ParameterManager* parManager = G4ParameterManager::Instance();  
  parManager->RegisterParameter(fEdep);  
  parManager->RegisterParameter(fEdep2);  
}
```

*The parameters not created
via the manager have to be
registered to it*

```
void B1RunAction::EndOfRunAction(const G4Run* run) {  
  ...  
  // Merge parameters  
  G4ParameterManager* parManager = G4ParameterManager::Instance();  
  parameterManager->Merge();  
  ...  
}
```

*The call to Merge() may be not necessary if
we hook the parameter manager to
G4AnalysisManager*

B1 example (cont.)

Geant4 10.3.

```
#include "G4ParameterManager.hh"
```

```
...  
B1RunAction::B1RunAction()
```

```
: G4UserRunAction(),  
  fEdep(0.),  
  fEdep2(0.)
```

```
{
```

```
  //Register parameter to the parameter manager
```

```
  G4ParameterManager* parManager = G4ParameterManager::Instance();
```

```
  parManager->RegisterParameter(fEdep);
```

```
  parManager->RegisterParameter(fEdep2);
```

```
}
```

*The parameters name may be omitted - **DONE***

The parameters not created via the manager have to be registered to it

```
void B1RunAction::EndOfRunAction(const G4Run* run) {
```

```
  ...  
}
```

*The call to Merge() should not be necessary – **TO DO***

Parameters & G4Run(Manager)

- The `parameterManager->Merge()`; is called at the same phase as G4Run
 - Can be added as an additional call to G4RunManager class
- Other possibilities
 - Add G4ParameterManager data (parameter) to G4Run
 - This would however produce “unused code” in user Run classes
 - Make G4ParameterManager a part of G4AnalysisManager
 - Then users would have to include tools headers while not using tools

Analysis & Scoring

Prototype

- Objective: automatic saving of users hits in a file using G4 analysis
 - Straightforward for scorers, as the hit maps type is already defined
 - Not evident for users hits defined via G4VHit base class
- First prototypes in the end of 2015, but did not ended in the release
- The second one, just for the lack of time before the release (I believe)
 - Complete implementation
 - Tested with basic examples which use scorers

Analysis & Scoring

- User can activate saving hits via added command:
 - `/hits/setOutput csv|root|xml|none`
- `G4ScoringAnalysis` class implements storing all hits collections of `G4THitsMap<G4double>`
- An ntuple with three columns is created for each primitive scorer:
 - `int` column - `eventNumber`
 - `int` column - `copyNumber`
 - `double` column - `scored value`
 - `ntupleName = sdName_hitMapName`

Integration in Geant4 Framework

- **G4ScoringAnalysis** has three public methods which have to be called during event processing by Geant4 kernel classes:
 - `void Initialize(G4HCofThisEvent* hce);`
 - by `G4EventManager::DoProcessing()` just after `currentEvent->SetHCofThisEvent(sdManager->PrepareNewEvent())`
 - `void Fill(G4HCofThisEvent* hce, G4int eventNumber);`
 - by `G4RunManager::ProcessOneEvent()` just after `UpdateScoring();`
 - `void Write();`
 - By `G4RunManager::RunTermination()` just after `if(fPersM) fPersM->Store(currentRun);`