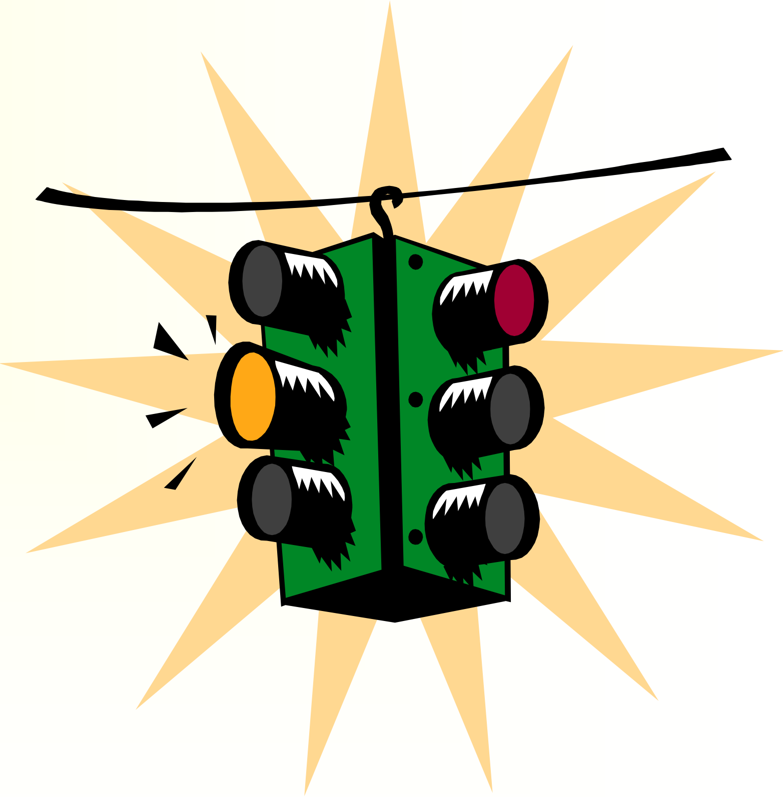
Geant4:User Actions and Analysis

- Geant4 user interface
- User actions
- Analysis tools
- External frameworks

Geant4 User Interface

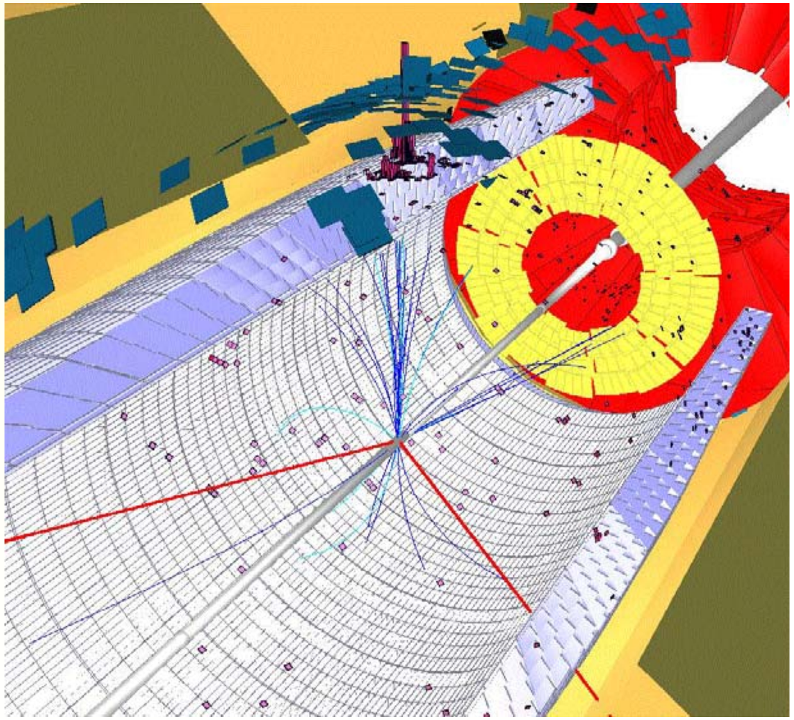
- ✶ Simulation of particle transport and interaction in Geant4 is under control of Geant4 kernel
- ✶ There are a variety of possibilities for user to get intermediate information and to score results of simulation
    - User actions
    - Sensitive detector
    - Physics processes



Geant4 User Actions

- G4UserRunAction
  - Begin and end of the run
- G4UserEventAction
  - Begin and end of an event
- G4UserTrackingAction
  - Begin and end of tracking of a G4Track
- G4UserSteppingAction
  - At each step
- G4UserStackingAction
  - Classification of new G4Track

CMS simulation of SUSY event



Geant4 User Actions

- Interfaces of user action use const references and pointers to preserve Geant4 kernel from user interventions:

```
void G4UserRunAction::BeginOfRunAction(const G4Run* run)
void G4UserRunAction::EndOfRunAction(const G4Run* run)

void G4UserEventAction::BeginOfEventAction(const G4Event* evt)
void G4UserEventAction::EndOfEventAction(const G4Event* evt)

G4ClassificationOfNewTrack
        G4UserStackingAction::ClassifyNewTrack(const G4Track* track)

G4ClassificationOfNewTrack = fUrgent      - put into urgent stack
                             fWaiting     - put into waiting stack
                             fPostpone    - postpone to the next event
                             fKill        - kill without stacking
```

Geant4 User Actions

- ✶ When G4Track is taken from a stack for tracking the user action is invoked:
  - void G4UserTrackingAction::PreUserTrackingAction (const G4Track* track)
  - Information is available about mother particle, vertex, position, 4-momentum, creation process, etc
- ✶ After the end of tracking the user action is invoked:
  - void G4UserTrackingAction::PostUserTrackingAction (const G4Track* track)
- ✶ Tracking action allows to control event history

Geant4 User Actions

- At each step user have access to complete information about each step of simulation
  - void G4UserSteppingAction::UserSteppingAction (const G4Step* step)
- The pointer to G4Step allows to access other instances:
  - const G4Track* track = step->GetTrack();
  - const G4StepPoint* prePoint = step->GetPreStepPoint();
  - const G4StepPoint* postPoint = step->GetPostStepPoint();
- WARNING: user code may be source of CPU penalty
- All what can be done using sensitive detector approach, can be also done inside UserSteppingAction for complete responsibility of the user
  - Inside G4UserSteppingAction the complete navigation in the geometry tree needs to be done
  - Sensitive Detector is called only if step is performed inside corresponding volume

G4UserSteppingAction example

```
// example/extended/electromagnetic/MuonProcesses

void SteppingAction::UserSteppingAction( const G4Step* aStep ) {
  G4StepPoint* prePoint = aStep->GetPreStepPoint();
  G4StepPoint* postPoint = aStep->GetPostStepPoint();

  //plot energy transfered
  G4double kinEnergyPreStep = prePoint->GetKineticEnergy();
  G4double kinEnergyPostStep = prePoint->GetKineticEnergy();
  G4double lgepsE = 1.0 – kinEnergyPostStep/ kinEnergyPreStep;
   if (etrans > 0.) lgepsE = std::log10(lgepsE);

  //count processes
  G4String procName = postPoint->GetProcessDefinedStep()->GetProcessName();
  G4int id = 0;
  if (procName == "muIoni")     id = 1;
  if (procName == "muIoni")     id = 1;
  if (procName == "muPairProd") id = 2;
  if (procName == "muBrems")    id = 3;
  if (procName == "muNucl")     id = 4;
  histoManager->FillHisto(id,lgepsE);
}
```

Verbosity

- Standard Geant4 verbosity can be activated via UI commands:
  - /run/verbose 1
  - /event/verbose 1
  - /tracking/verbose 1
- User have a possibility to have additional verbosity in user action classes
- It is possible to have user implementation of G4VSteppingVerbose class, for example,
  - $G4INSTALL/example/extended/electromagnetic/TestEm1

Analysis Tools

- Analysis tools allow to store histograms, nTuples, and other objects with results of Geant4 simulation
- Geant4 has no any native analysis tool – it is external software
- There are several Geant4 examples using analysis tools:
  - examples/extended/analysis/
  - examples/extended/electromagnetic
  - examples/advanced
- Available tools:
  - PI (CERN)
  - ROOT (CERN)
  - JAS (SLAC)
  - Open Scientist (LAL)
- AIDA interfaces
- Output formats: hbook, root, xml
- Analysis software are under intensive development

Example of Analysis - TestEm7

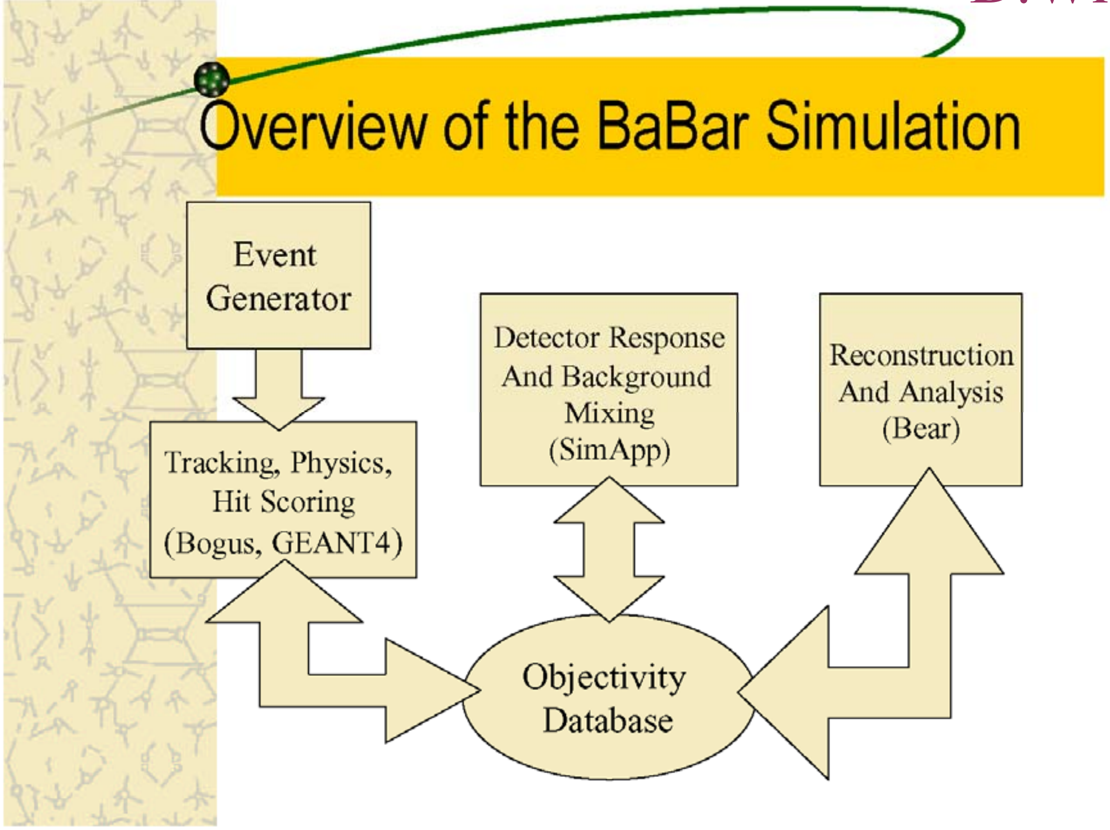
```
class RunAction : public G4UserRunAction   {

public:
  RunAction(DetectorConstruction*, PhysicsList*, PrimaryGeneratorAction*);
   ~RunAction();

   void BeginOfRunAction (const G4Run*);
   void EndOfRunAction (const G4Run*);

   // Specific methods for the example
   void FillTallyEdep (G4int n, G4double e) {tallyEdep[n] += e;};
   G4double GetBinLength() {return binLength;};
   G4double GetOffsetX()   {return offsetX;}
   void     FillHisto(G4int id, G4double x, G4double weight = 1.0);
   void AddProjRange (G4double x) {projRange += x; projRange2 += x*x;};

private:
   void bookHisto();                          // Access to external software
   void cleanHisto();

   // Specific members of the example
   DetectorConstruction*      detector;
   PhysicsList*               physics;
   PrimaryGeneratorAction*  kinematic;
   G4double*            tallyEdep;
   G4double             binLength;
   G4double             offsetX;
   G4double             projRange, projRange2;

   // AIDA interfaces
   AIDA::IAnalysisFactory* af;
   AIDA::ITree*                tree;
   AIDA::IHistogram1D*      histo[1];
};
```

External Frameworks

- Each large HEP project has a software framework and Geant4 as a toolkit can be driven by the framework
- User actions are used to exchange data between Geant4 kernel and the framework
  - Factory approach: many specialized user actions
- There are also useful tools built on top of Geant4, which can be used in small projects

BaBar (SLAC, Stanford) - Leader Experiment Using Geant4

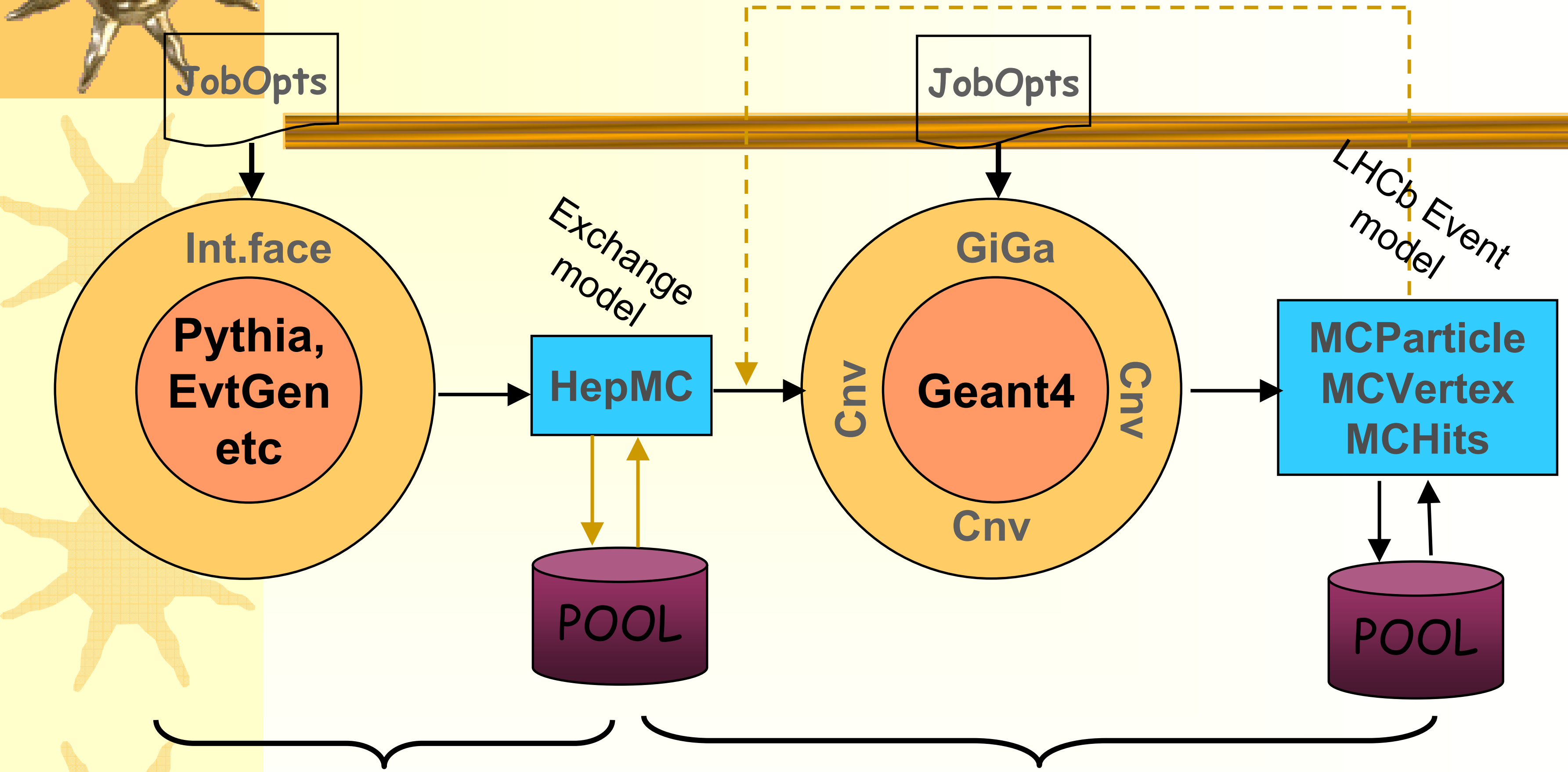
D.Wright, 2003



Overview of the BaBar Simulation

Event Generator
Tracking, Physics, Hit Scoring (Bogus, GEANT4)
Detector Response And Background Mixing (SimApp)
Reconstruction And Analysis (Bear)
Objectivity Database

LHCb: Gauss Application (2004)



Two phases

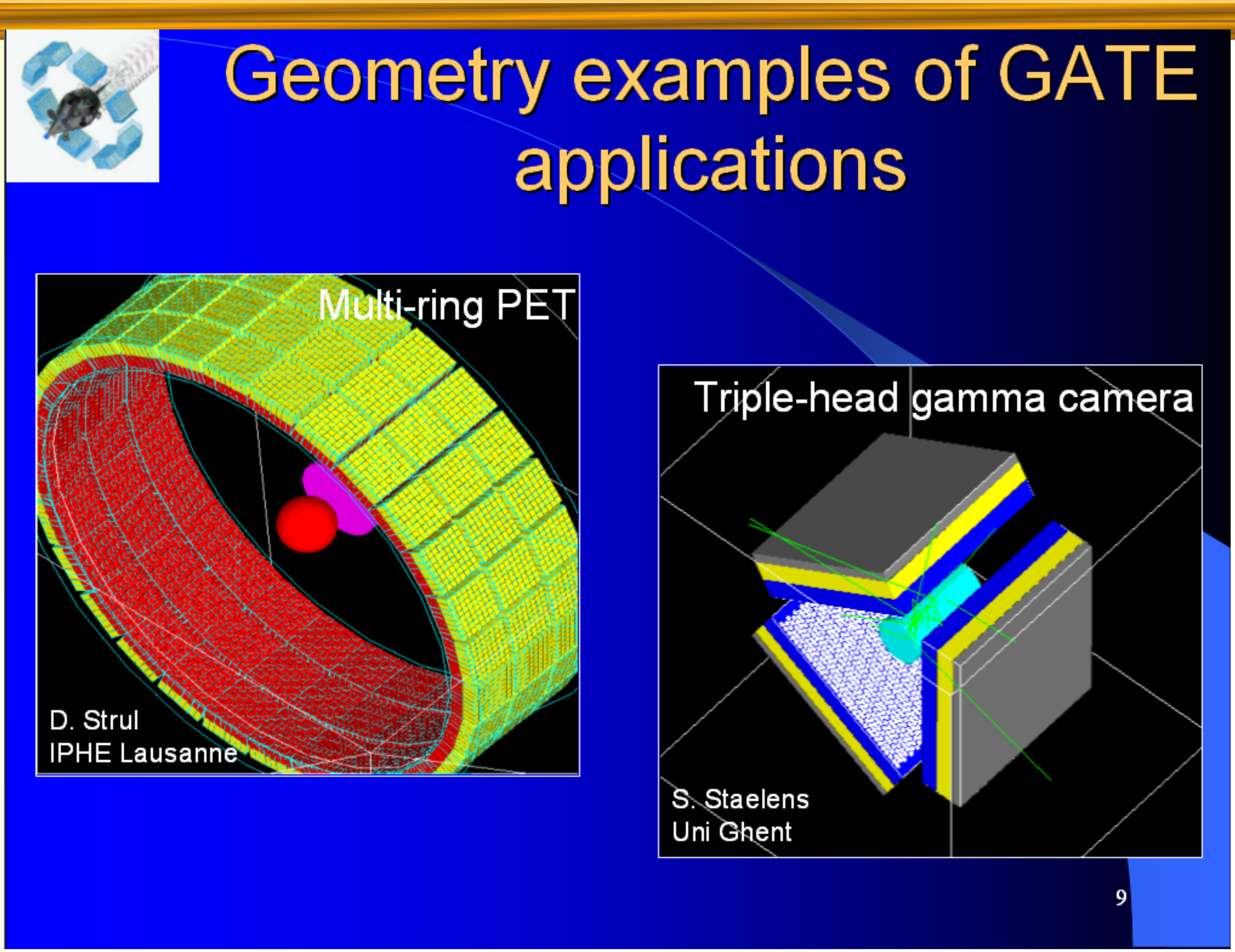
Event generation

primary event generator
specialized decay package
pile-up generation

Detector Simulation

geometry (LHCb → G4)
tracking through materials (G4)
hit creation (G4 → LHCb)
MC truth generation (G4 → LHCb)

GATE Software for Tomography



Geometry examples of GATE applications

Multi-ring PET

D. Strul
IPHE Lausanne

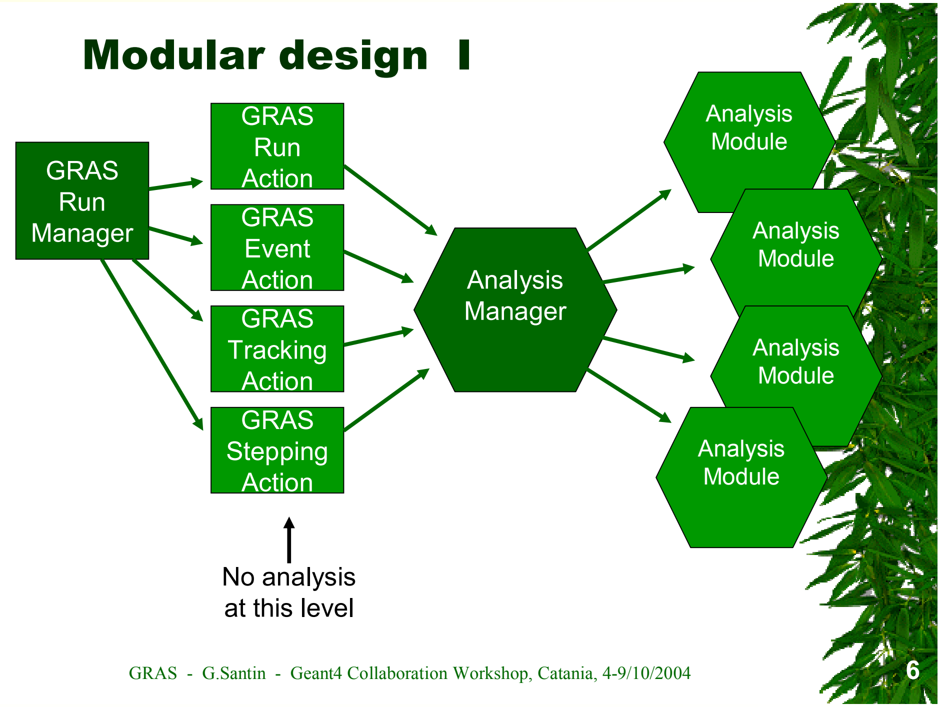
Triple-head gamma camera

S. Staelens
Uni Ghent

9

General Radiation Analysis for Space - GRAS

- New project of ESA
- Service for radiation studies for space applications
- Generic dosimetry and analysis
- Can be applied for studying of different problems
- Available for testers, public release soon



Modular design I

GRAS Run Manager

GRAS Run Action

GRAS Event Action

GRAS Tracking Action

GRAS Stepping Action

Analysis Manager

Analysis Module

Analysis Module

Analysis Module

Analysis Module

No analysis at this level

GRAS - G.Santin - Geant4 Collaboration Workshop, Catania, 4-9/10/2004

6

Conclusion remarks

- The Geant4 toolkit provides a wide choice of user actions allowing detailed monitoring of the simulation
- It is user’s responsibility to choose and design of user actions and/or analysis engine
- In Geant4 examples (novice, extended, advanced) there are proposed solutions for different use-cases
- User actions may require extra CPU and/or memory