Geant4 Radiation Analysis for Space

GRAS

Presentation and demonstration

G. Santin
ESA / Rhea System SA

V. Ivantchenko
CERN

Geant4 Tutorial for Space Industry
ESTEC, March 3, 2005
Outline

- Motivation, and description of the tool
- Demonstration, with a simple example
- Present status, expectations, conclusions
Motivation

* Wide application of Geant4 models at ESA
  - Astronomy (GAIA, JWST…)
  - Planetary (Bepicolombo,…)
  - Earth observation (Aeolus,…)
  - Manned space-flight (ISS, Man2Mars,…)

* Repeated analysis of similar cases
  - Several major geometry modeling per year
  - Similar analyses
→ Avoid re-writing of similar C++ applications

* Internal to ESA
  - Make the support process faster
→ Offer standard advanced simulation outputs

* Support to industry
  - Promote the use of Geant4 for the radiation analysis studies provided in ESA projects
→ Offer standard advanced ready-to-use simulation tool
Effects analysis

Electronics and detector degradation due to Total Ionising Dose (TID) or Displacement Damage.

Single event effects (SEE), transient or destructive.

Detector calibration and interference (including contamination, charging, glitches).

Radiation effects on humans

- human space-flights, exploration programs
- tissue and organ effects, equivalent / ambient / effective dose (ICRP, ICRU recommendations)

Sources

- Jovian electrons
  - Dominant in quiet time

- Trapped radiation
  - Electrons \( \sim < 10 \text{ MeV} \)
  - Protons \( \sim < 10^2 \text{ MeV} \)

- Solar radiation
  - Protons, some ions, electrons, neutrons, gamma rays, X-rays…
  - Softer spectrum
  - Event driven – occasional high fluxes over short periods.

- (Extra) Galactic and anomalous Cosmic Rays
  - Protons and ions
  - \( <E> \sim 1 \text{ GeV}, E_{\text{max}} > 10^{21} \text{ eV} \)
  - Continuous low intensity
The example of MULASSIS

- Geant4 is a “Toolkit”
  - Flexible, powerful, extendable,…
  - But intentionally “not a tool” ready for use

- MULASSIS (Qinetiq, ESA contract)
  - Multiple layer geometry via macro
  - Materials by chemical formula
  - First Geant4 application with web interface (in SPENVIS)
  - User success
  - Raised the standard level of radiation shielding analysis in the space community (both academia and industry)

- Limitations
  - 1D geometry
  - Extensibility
## User Requirements

- Self sufficient tool (Geometry, Physics, Source, Analysis)  
  - Available as standalone executable
    - No need to download and compile Geant4
- Easy to integrate in existing applications
- Analysis types
  - 3D
  - Dose, Fluence, NIEL, activation... for support to engineering and scientific design
  - Dose Equivalent, Equivalent Dose,… for ESA exploration initiative
  - Transients: PHS, LET, SEU models

- Analysis independent from geometry input mode
  - GDML, or existing C++ class, …

- Different analyses set without re-compilation

- Modular / extendable design

- Source and Physics description adequate to space applications
  - Solar events
  - Cosmic rays
Some strategic choices

- Collect experience from previous successful applications

- Geometry
  - GDML as default option
    - Adopted as exchange format by SPENVIS
  - User C++ class

- Physics
  - Modular
  - Selectable via macro
    - G4 official EM lists
    - G4 official EM lists with hadronics
    - G4 official Hadronic lists
  - User can use a private C++ Physics list class

- Primary generation
  - GPS as default
  - User can provide generator class

- Analysis set via macros
  - No need to recompile

- Output
  - Interface to AIDA histogramming
  - ASCII output always available
Geometry format

GDML ...

- GDML as default option
  - Adopted as exchange format by SPENVIS

- User C++ class (if necessary, if it already exists,...)
  - Basic C++ geometry provided for testing

Analysis independent from geometry input mode
- GDML, or existing C++ class, ...

Visualization attributes
- Colour
- Visibility
- Style (Wireframe,...)

Can be assigned once the geometry is loaded

- GDML (Geometry Description Markup Language)
  - Geometry data exchange
  - GDML Schema: a set of XML schema definition (.xsd) files specifying the GDML syntax
  - Looks similar to HTML, with specific tags for materials, shapes, positions, rotations, ...
  - ASCII file: easy to create, read, debug, modify,...
Analysis

Modular design

GRAS Run Manager

GRAS Run Action

GRAS Event Action

GRAS Tracking Action

GRAS Stepping Action

Analysis Manager

Dose Analysis Modules

Fluence Analysis Modules

NIEL Analysis Modules

... Analysis Modules

No analysis at this level
**GRAS elements**

- **VolumeID**
  - To identify a volume in the geometry tree
  - At present implemented as the couple (name, copyNo)

- **VolumeInterface**
  - To identify the boundary between two volumes
  - Couple of VolumeID's

- Each module can have
  - several VolumeID's and
  - several VolumeInterfaces

- Different actions taken by various module types when “in volume” / “at interface”

- Result output units
  - User choice, module type dependent

---

**Example: dose module “DoseB12”**

- Sensitive volumes:
  - b1 and b2

- Interface (to tag particle type):
  - between (sat, world)
  - To detect secondaries created in the satellite structure
Analysis types

At the moment modules for:

- Dose
- Fluence
- NIEL
- Dose equivalent
- Equivalent dose
- Path length
- SEE

Component degradation, background

Human exploration initiatives

Components, SEE

Foreseen types:
- Activation
- LET / more SEE
- Charging (in progress)
- Source monitoring
GRAS Analysis modules:
Component degradation, Background

* Dose
  - Also per incoming particle type, with user choice of interface
  - Gives event Pulse Height Spectrum
    - For analysis of induced signal
  - User input mass
  - Units:
    - MeV, rad, Gy

* Fluence
  - Particle type, energy, direction
  - One/Both ways

* NIEL
  - MULASSIS implementation
  - Modular approach
  - Several curve sets available
    - CERN/ROSE (p, e-, n, pi)
    - SPENVIS/JPL (p)
    - Messenger Si (p, e-)
    - Messenger GaAs (p, e-)
  - Units:
    - 95MeVmb, MeVcm2/g
    - MeVcm2/mg, keVcm2/g

http://www.stsci.edu/hst/nicmos/images/crpersist.frame.gif

JWST NIRSpec
Biological effects modules

- Dose equivalent
  - ICRP60 LET-based coefficients
  - Units: MeV, Sv, mSv, Gy, rad

- Equivalent Dose
  - ICRP60 weights
  - User choice of weight interface
  - Units: MeV, Sv, mSv, Gy, rad

New user requirements include:
- planetary models (e.g. scaling of SPE fluence to other planets, magnetic field description, crustal maps)
- ion physics (electromagnetics / hadronics for HZE)
- biological effects (macroscopic / microscopic models)
GRAS Analysis modules:

**SEE in microelectronics**

- **Path length analysis**
  - Event distribution of particle path length in a given set of volumes
  - If used with “geantinos”, it provides the geometrical contribution to the energy deposition pattern change
    - In a 3D model
    - W.r.t. a 1D planar irradiation model

- **SEE models**
  - Threshold simple model implemented
  - Design open to more complex modeling
GRAS is being used for

- Herschel
  - Test beam detector study
  - Radiation effects to photoconductors and bolometers

- JWST
  - Dose
  - Background

- Electronic components
  - Rad-hardness, local shielding, etc.
GRAS demonstration

- Guide through a simple application
  - Initialisation
  - Geometry
  - Physics
  - Visualisation
  - Analysis
  - Radiation environment
  - Results
GRAS Scripting Control

- GRAS control is based on scripting
  - Initialization
    - Physics
    - Geometry model
  - Analysis
    - Analysis modules
    - Module parameters
  - Output
    - File name / type
    - Units

Scripting can be
  - Interactive
    
    ```
    Idle > /gras/analysis/dose/addModule doseB1
    Idle >
    ```
  - Via macro files
    
    ```
    Idle > /control/execute analysis.g4mac
    Idle >
    ```
  - Executed in batch
    
    ```
    Shell > gras analysis.g4mac > file.log
    Shell >
    ```
Scripting

Initialisation

1. Insert the physics modules
   - `/testem/phys/addPhysics standard`

2. Modify the default cut values
   - `/testem/phys/setCuts 0.1 mm`

3. Define the geometry model
   - `/gras/geometry/type gdml`
   - `/gdml/file geometry/see1.gdml`
   - `/gdml/setup SEE`

4. Initialize Geant4
   - `/run/initialize`
GDML

Some details

Define elements and materials*

* All elements and common materials are provided in a macro

Define shapes

Define volume tree
Scripting

Cuts per region

- Add a region
  - `/gras/physics/region/addRegion region1`

- Modify the cuts in this region
  - `/gras/physics/region/listRegions`
  - `/gras/physics/region/setRegionCut region1 default 0.1 mm`

- Assign one or more volumes to this region
  - `/gras/physics/region/addVolumeToRegion b1 region1`
  - `/gras/physics/region/addVolumeToRegion b3 region1`

- Modify the cuts in the Default region
  - `/gras/physics/region/setRegionCut DefaultRegionForTheWorld default 10. mm`
**Scripting**

**Visualisation**

Define new colours:

* All these colours are predefined and provided in a macro

Assign colours to volumes

Modify visualisation attributes

... 

```
/gras/vis/util/addColour pink 1 0.3 0.7
/gras/vis/util/addColour purple 0.63 0.13 0.94
/gras/vis/util/addColour orchid 0.85 0.44 0.84

/gras/vis/util/addColour light_green 0 0.75 0
/gras/vis/util/addColour aquamarine 0.5 1 0.83
/gras/vis/util/addColour dark_green 0 0.39 0
/gras/vis/util/addColour olive_green 0.33 0.42 0.18
/gras/vis/util/addColour khaki 0.94 0.90 0.55

... 
```

```
/gras/vis/util/setVolumeColour World brown
/gras/vis/util/setVolumeColour satellite gold
/gras/vis/util/setVolumeColour support olive_green
/gras/vis/util/setVolumeColour b1 turquoise
/gras/vis/util/setVolumeColour b2 orchid
/gras/vis/util/setVolumeColour b3 navy_blue
/gras/vis/util/setVolumeColour b4 salmon

/gras/vis/util/setVolumeVisibility World false
```
**Scripting**

**Add an analysis module**

Add a module of a chosen type

```
/gras/analysis/dose/addModule doseB12
```

Assign one or more VolumeIDs

```
/gras/analysis/dose/doseB12/addVolumeID pv_b1_0 -1
/gras/analysis/dose/doseB12/addVolumeID pv_b2_0 -1
```

Assign one or more Interfaces

```
/gras/analysis/dose/doseB12/addVolumeInterface pv_satellite_0 -1 World -1
```

Set specific parameters

```
/gras/analysis/dose/doseB12/setMass 0.0186 g
```

Choose the output unit

```
/gras/analysis/dose/doseB12/setUnit MeV
```

* “Object oriented” scripting
  - G4 UI commands “per module”
  - Automatic module UI tree
  - *a la GATE*

GRAS - G.Santin - Geant4 Tutorial for Space Industry, ESTEC, 3 March 2005
Source

Define the radiation environment

Shoot the particles from a sphere

Mimic the isotropic flux

Set the type of primary particle

Set a spectrum (AE8, GALILEO 15y)

Run!

```plaintext
/gps/pos/type Surface
/gps/pos/shape Sphere
/gps/pos/centre 0.0.0.0 mm
/gps/pos/radius 25.0 mm

/gps/ang/type cos
/gps/ang/mintheta 0. deg
/gps/ang/maxtheta 90. deg

/gps/particle e-

/gps/ene/type Arb
/gps/hist/type arb
/gps/ene/min 4.000E-02 MeV
/gps/ene/max 7.000E+00 MeV
/gps/hist/point 4.000E-02 2.245E+08
/gps/hist/point 1.000E-01 1.490E+08
... 
/gps/hist/point 6.500E+00 0.000E+00
/gps/hist/point 7.000E+00 0.000E+00
/gps/hist/inter Lin

/run/beamOn 1000000
```
Output

Analysis results

... [EventAction::BeginOfEventAction] Event # 1000000 started
Run terminated.

Run Summary
- Number of events processed: 1000000
- User=167.91s Real=171.38s Sys=0.18s

Histograms and Ntuples are saved

...

Module: doseB1 Run: 1 Total average dose (per event)
(rad, for a mass of 9.3e-06 kg): 5.66e-11 +/- 7.33e-12

Module: doseB2 Run: 1 Total average dose (per event)
(MeV): 3.447e-05 +/- 4.423e-06

Module: doseB12 Run: 1 Total average dose (per event)
(rad, for a mass of 1.86e-05 kg): 5.799e-11 +/- 5.287e-12

Module: doseB12 Run: 1 e- average dose (per event)
(rad, for a mass of 1.86e-05 kg): 5.785e-11 +/- 5.286e-12

Module: doseB12 Run: 1 gamma average dose (per event)
(rad, for a mass of 1.86e-05 kg): 1.435e-13 +/- 6.493e-14

...

Some feedback during the execution

Result summary per module

Here:
dose (with error)
for b1, b2 and b1+b2
Output

Analysis results

...  
### Run 1 start  
### Histo books 11 histograms  
Tree store : data/see_see1_1000000.hbook  
### Histo books 16 tuples  
Start Run processing.  
[EventAction::BeginOfEventAction] Event # 100000 started  
...

AIDA output: ready (configurable) histograms and “tuples”

AIDA (here HBOOK file)
List of file content

1 (1) fluence1 Particle: proton Fluence spectrum in MeV  
2 (1) fluence1 Particle: e- Fluence spectrum in MeV  
...  
6 (1) doseB2 dose in MeV  
7 (1) doseB12 dose in rad  
8 (1) doseEqB2 dose equivalent in mSv  
9 (1) eqDoseB2 equivalent dose in mSv  
10 (1) pathLengthB1 pathLength in mm  
11 (1) pathLengthB1234 pathLength in mm  
1001 (N) fluence1 Particle: proton  
1002 (N) fluence1 Particle: e-  
...  
1009 (N) doseB2 Event total dose  
1010 (N) doseB2 Event dose per particle  
1011 (N) doseB12 Event total dose  
1012 (N) doseB12 Event dose per particle  
1013 (N) pathLengthB1 Event total pathLength  
1014 (N) pathLengthB1 Event pathLength per particle  
1015 (N) pathLengthB1234 Event total pathLength  
1016 (N) pathLengthB1234 Event pathLength per particle
For present Geant4 users

**GRAS and previous work**

* 2 ways of obtaining GRAS output without discarding hours/days/months of work

  A. Inserting C++ Geometry, Physics and/or Primary Generator classes inside GRAS
     • In the main gras.cc
  B. Inserting GRAS into your existing applications

* Which way is the fastest depends on existing work
For present Geant4 users

B. GRAS in your G4 application

- On top of your existing G4 application
  - Without discarding work done so far

- To obtain dose/fluence/NIEL/… in parallel to your analysis
  - Insert GRAS as additional analysis module of existing applications, without interfering with existing Geometry, Physics, UserActions
  - Via the GRAS Run Manager
For present Geant4 users

B. GRAS in your G4 application

GRAS Run Manager

Run Action
Gras Yours ...

Event Action
Gras Yours ...

Tracking Action
Gras Yours ...

Stepping Action
Gras Yours ...

Analysis Manager

Analysis Module

Analysis Module

Analysis Module

* Via the GRAS Run Manager
GRAS in your application HOW-TO

- Use GRASRunManager instead of G4RunManager
  - Derive your RunManager class from the GRAS one in case you have your private one already

- Compilation
  - Compile GRAS (library) separately
  - Link with GRAS library without need of copying GRAS files locally (gras.gmk file provided)

- GRAS design details
  - User Action Stacks implemented for Run, Event, Tracking, Stepping
    - GRAS User Action called first, then the others
  - GRAS Run Manager uses GRAS action stacks
    - Initialized with GRAS geometry, GRAS physics and UserActionStacks
    - G4RunManager methods
      - setUserAction() overloaded
        - They insert the UserActions in the stacks
  - Currently no use of Sensitive Detectors
    - Detection at Stepping
GRAS and future work

* Develop the geometry
  - GDML

* Insert the analysis modules
  - G4 UI macro

* No need to program in C++
  - If analysis types are enough

* Extend GRAS capabilities
  - Derive a new module type from an existing one
  - Develop a new module

* GRAS Analysis Module
  - Framework for the analysis
  - Provides all tracking information ready for use
GRAS

Analysis Module

- Self contained analysis element
  - Initialization, event processing, normalization, printout → all inside

- Only one class to create/derive in case a new type of analysis is needed
  - No need to modify Run+Event+Tracking+Stepping actions

- AIDA histogramming “per module”

- G4 UI commands “per module”

- Easy to extend the present analysis with new module types!

GRAS - G.Santin - Geant4 Tutorial for Space Industry, ESTEC, 3 March 2005
Coming soon…

* New analysis types
  - Activation, LET/SEE, charging, source monitoring
* Minor improvements
  - Automatic dose per unit mass (e.g. rad)
  - Automatic normalization to real flux in space
    * Requires (minor) extensions to GPS
* Interface to future G4 upgrades
  - Dose tallying in parallel geometry
* Geometrical biasing
  - To improve speed for local energy deposition
  - Analysis algorithms are ready for biasing
* Release of source and static executable
  - All Geant4 ESA s/w is open source
* Web Interface inside SPENVIS
  - Internal geometry, GDML exchange format
Status

* CVS repository online

* Code
  - Alpha tag Jan’05, works with
    * Geant4 6.2.p02
    * GDML 1.0
  - Latest stable tag works with
    * Geant4 7.0.p01
    * GDML2 is required and will be available soon from CERN
  - Public release needs a last validation of the last tag
    * Will probably work with G4 6.2.p02

* User Manual
  - TOC: > 1 page!
Conclusions

- Modular analysis package
- Space users oriented, but trying to be generic
- Open to comments to improve design and analysis types
- Already used in the support of a number of space missions

Thanks to
- Radovan Chytracek, Witek Pokorski for support on GDML
- Fan Lei, Pete Truscott for the MULASSIS implementation of NIEL analysis
- Hugh Evans for support on design and CSV output format