QinetiQ

Software Specification Document

Title:	**Geant4-based Microdosimetry Analysis Tool**
Document reference	QINETIQ/KI/SPACE/SRD010066/0.A
Project reference:	Parent project: QINETIQ/KI/SPACE/7/36/8
Prepared by:	Pete Truscott / Fan Lei
Signature:	
Date:	
Authorised by:	P Hedley Stokes
Signature:	
Issued by: Sector/location/telephone/fax:	KI Systems, Space Department Room G081, A8 Building QinetiQ, Farnborough Tel: 01252 393290 Fax: 01252 396330

Abstract

The software requirements and architectural design are described for a computer code to be used to simulate microdosimetry in geometries representing features of a semiconductor device. The tool is intended to be based on the Geant4 radiation transport simulation toolkit, which can or will simulate proton nuclear and electromagnetic, and electron interactions in the energy range applicable to microdosimetry effects induced by the space radiation environment. The Geant4-based Microdosimetry Analysis Tool (GEMAT) will be integrated into the ESA SPENVIS web-based space environment simulation tool-set. The output of the code will be text files, including comma-separated variable data files of the calculated energy deposition spectra.

Record of changes

This is a controlled document. Additional copies are obtained through the issuing authority. In the event of copying locally, including electronically, mark each document 'Uncontrolled Copy'. Full issues and page amendments are identified on this page. Forward proposals for change in writing to the issuing authority.

Issue	Date	Detail of changes
0.a	8 October 2001	First issue to reviewers in draft form

Table of contents

1 Introduction

1.1 Contractual

This document has been issued by QinetiQ Space Department for ESA/ESTEC under contract 14968/00/NL/EC (ESA Technology Research Programme, Space Environment and Effects Major Axis), in part fulfilment of Work Package 3 of the contract "development of software tools".

1.2 Purpose of the Document

This document describes the software requirements and architectural design of the Geant4-based Microdosimetry Analysis Tool (GEMAT) and specifies the operations to be carried out in functional terms. This includes a definition of the code at a broad structural (class) level. The detailed structure (to member-function level) will be defined in the detailed design phase. This document is intended to be an input to this next phase of code development.

1.3 Scope of the Software

GEMAT will utilise the Geant4 radiation transport toolkit as the basis of the simulation to treat the run-management, geometry tracking, physical interaction processes and user interface. Specifically, GEMAT shall use Geant4 4.0, and will interact with and be accessible via the ESA SPENVIS system through the GEMAT web-page interface being developed by BIRA. This document therefore concentrates on those aspects of GEMAT that use Geant4 as the basis of the simulation (GEMAT/executable) and supporting Unix script used for control of GEMAT/executable (GEMAT/shell-script) – see figure 1.1.

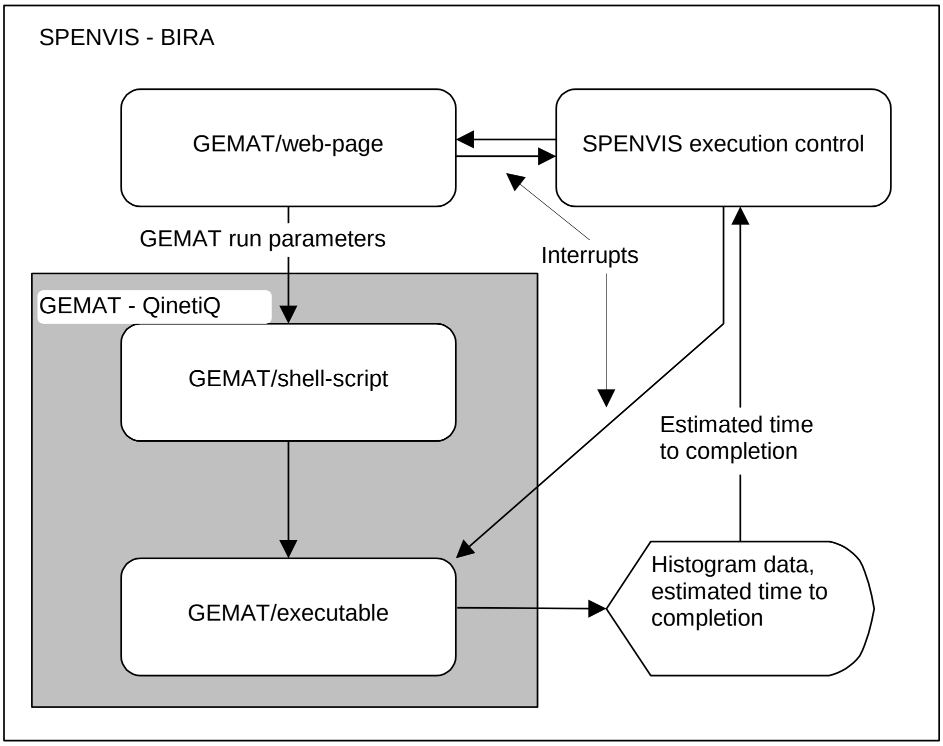

Figure 1.1: Logical model diagram for overall GEMAT suite.

Interface definition for GEMAT/web-page, GEMAT/shell-script and GEMAT/executable is covered in a separate document.

1.4 Definitions, acronyms and abbreviations

ASCII	American Standard Code for Information Interchange
BIRA	Belgisch Instituut voor Ruimte-Aëronomie
CERN	Conseil Europeen pour la Recherche Nuclaire
CRÈME	Cosmic Radiation Effects on Micro-Electronics
CSV	comma-separated value
ESA	European Space Agency
ESTEC	European Space Technology Centre
Geant4	C++ toolkit for Monte Carlo simulation of high-energy, fundamental particle transport, developed by an international collaboration led by CERN.
GEMAT	Geant4-based Microdosimetry Analysis Tool
GUI	graphical user interface
HETC	High-Energy Transport Code
NIEL	non-ionising energy loss
OO	object-oriented
PDF	probability distribution function
PHS	pulse-height spectrum
SPENVIS	SPace ENvironment Information System
SR	software requirement
SSAT	Sector Shielding Analysis Tool
SSD	software specification document
TBC	To Be Confirmed
TBD	To Be Decided
UI	user interface
UR	user requirement
URD	User Requirements Document
UrQMD	Ultra-relativisitic Quantum Molecular Dynamics

VRML Virtual Reality Meta-Language

WWW World-Wide Web

1.5 References

[1] The CERN Geant4 Collaboration provides a significant amount of references information: http://wwwinfo.cern.ch/asd/geant4/geant4.html
From this Web page, access can be obtained to User Documentation:

[2] User Guide for Application Developers: http://wwwinfo.cern.ch/asd/geant4/geant4_public/G4UsersDocuments/UsersGuides/ForApplicationDeveloper/html/index.html

[3] User Guide for Toolkit Developers: http://wwwinfo.cern.ch/asd/geant4/geant4_public/G4UsersDocuments/UsersGuides/ForToolkitDeveloper/html/index.html

[4] The Physics Reference manual: http://wwwinfo.cern.ch/asd/geant4/geant4_public/G4UsersDocuments/UsersGuides/PhysicsReferenceManual/html/index.html

[5] The Software Reference manual provides information on the public methods to the Geant4 classes: http://geant4.kek.jp/cgi-bin/G4GenDoc.csh?flag=1

[6] The parent project reference is the proposed 'ESA Radiation Effects Analysis Tools', QINETIQ/KI/SPACE/7/36/8/2. This parent project is itself defined in Work Order contract 14968/00/NL/EC, file reference QINETIQ/KI/SPACE/7/36/8/1/1 as modified by the minutes of meetings (QINETIQ/KI/SPACE/7/36/8/1/3).

[7] Geant4-based Microdosimetry Analysis Tool: Software Project Management Plan QINETIQ/KI/SPACE/SPMP011037.

[8] Geant4-based Microdosimetry Analysis Tool: User Requirements Document QINETIQ/KI/SPACE/URD011023.

[9] Geant4-based Microdosimetry Analysis Tool: Interface Control Document QINETIQ/KI/SPACE/ICD010174.

[10] Guide to applying the ESA software engineering standards to small software projects, ESA BSSC(96)2 Issue 1, May 1996.

The Booch notation will be used for the OO analysis and design diagrams. Good references for this notation are:

[11] 'Object-Oriented Analysis and Design with Applications', Booch G. – 2nd ed., Addison Wesley Publishing Company, 1994

[12] 'Designing Object Oriented Applications using the Booch Method', Robert C. Martin, Prentice Hall, 1995

Good references used for the C and C++ languages are:

[13] 'C++ Nuts and Bolts for Experienced Programmers', Herbert Schildt, Osborne McGraw-Hill, 1995.

[14] C++ How to Program', Deitel and Deitel – 2nd ed., Prentice Hall, 1998.

1.6 Overview of document

This document describes the top-level structure of the code and the functions the GEMAT software will provide. Two key user requirements are that GEMAT:

- Utilises the Geant4 toolkit, placing a requirement on the final software to interact with the appropriate Geant4 classes to perform geometry definition, particle tracking, and physical interaction simulation.

- Form part of the SPENVIS system, which provides information to users on space environment and effects prediction over the WWW.

The systems requirements in this document are therefore more specific than for normal SRDs, and frequent reference is made to the existing Geant4 classes and their methods. The reader is directed to reference [3] for further information on the toolkit and the methods by which the user is expected to interact with the toolkit.

2 Design and development assumptions and dependancies

It is assumed that the physics processes are modelled through existing classes implemented in Geant4, or currently being developed for the toolkit. This software project will **not** involve the development or implementation of any new physical processes. The key physical process missing, which is relevant to the space radiation environment is the treatment of the nuclear interactions of protons or light heavy ions. It is presently understood that the intranuclear cascade code (a re-engineering of HETC) will be available by the end of 2001 in Geant4 4.0 to address proton-nuclear interactions. However, models such as Ultra-relativistic Quantum Molecular Dynamics (UrQMD) model, which will treat nuclear-nuclear interactions, are being developed by the Geant4 collaboration on a longer time-scale.

In addition, the treatment of volume-dependant particle cut-offs is a feature that is a consideration for GEMAT (UR 1.7), and variance reduction for "path-length shrinking" a requirement (UR 3.4) - neither of these are currently treated within Geant4. The inclusion of these facilities in the tool is dependant upon the implementation of such variance reduction features within Geant4 4.0. It is further assumed that Geant4 4.0 is released on 17 December 2001.

GEMAT is intended to be accessible to users of the SPENVIS system, with the code operating on computers at BIRA, Belgium. It is assumed that the developers of the SPENVIS system will be responsible for developing:

- web-page interfaces (GEMAT/web-page);
- utilities to convert text-file data output to graphics files.

It is further assumed that suitably powerful PC/Linux or Sun platforms (see §2.5) are available at BIRA or QinetiQ to operate GEMAT – in the latter case these would be tasked by the SPENVIS system. (Note that contract 14968/00/NL/EC includes provision for the purchase of a computer or computers, potentially for this purpose.)

Execution times for GEMAT may be significantly longer than other programs in the SPENVIS suite, depending upon the detail of the microdosimetry study. It is assumed that the GEMAT simulation will not be killed, except after a maximum execution time (set by the SPENVIS System Administrator) or cancel signal sent by the user (see URs 5.7, 5.8, and 5.9).

3 Logical model description

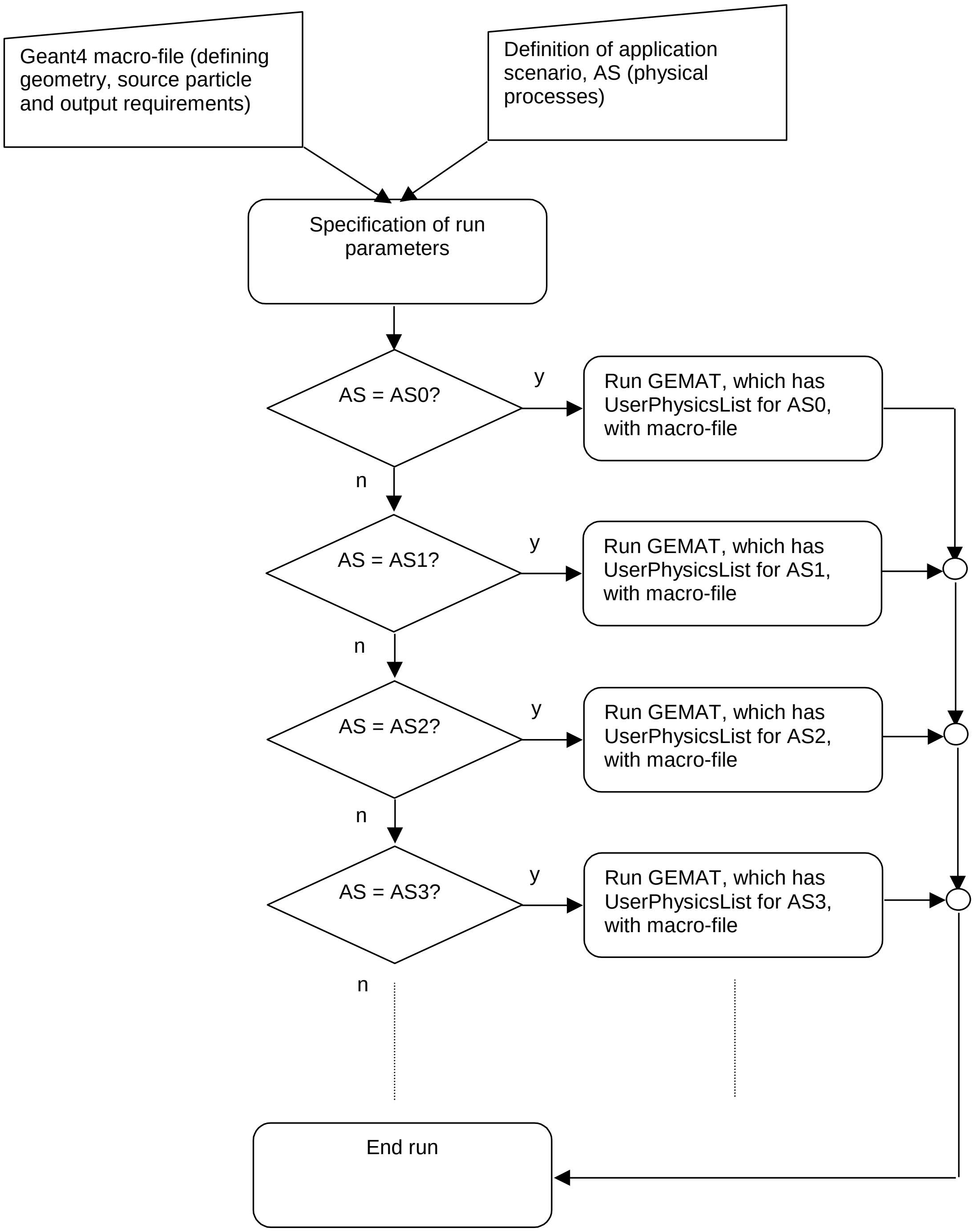

Figure 2.1: Top-level process diagram (GEMAT/shell-script)

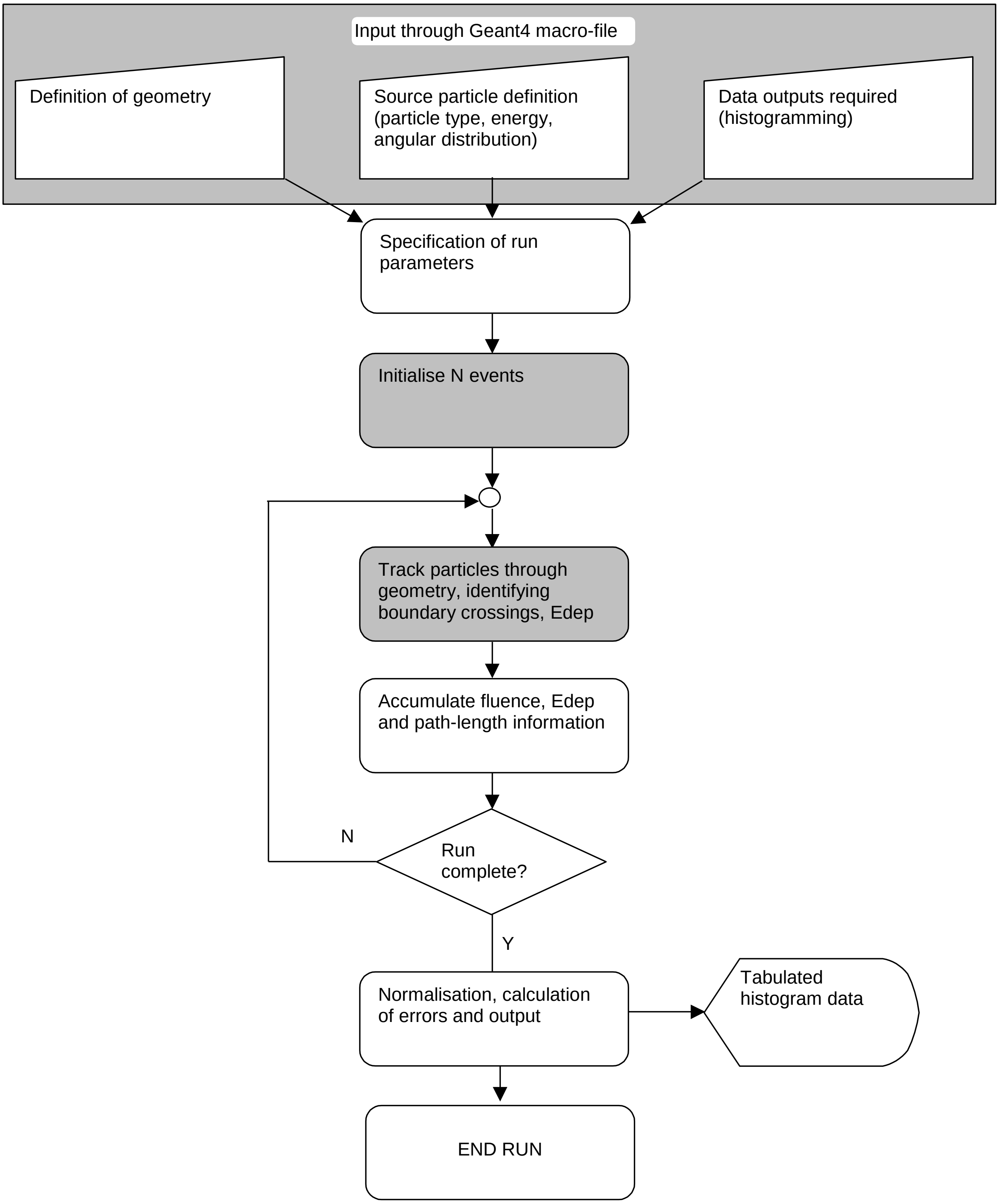

Figure 2.2: Second-level process diagram (GEMAT/executable)

Figure 1.1 shows the top-level process description for the complete GEMAT suite. The diagrams in figures 2.1 and 2.2 show the logical processes required for those aspects of GEMAT to be developed by QinetiQ (GEMAT/shell-script and GEMAT/executable). The user is required to specify the geometry to be analysed, the source particle definition, application scenario and the output data types and format, which are passed to GEMAT/shell-script and through there to GEMAT/executable. Specific information includes:

- geometry construction, material type and energy thresholds / funnel lengths associated with volumes in the geometry;
- particle type, spectrum, area of the geometry over which particles are incident and angular distribution;
- depletion volumes for which microdosimetry data are required;
- range and binning intervals for histograms of particle fluence as a function of depletion volume, particle species and energy;
- range and binning intervals of pulse-height energy deposition spectra as a function of depletion volume;
- range and binning interval for path-length distribution as a function of depletion volume and particle species.

Controlling of the simulation process and tracking of the particles through the geometry shall be performed by the Geant4 toolkit (some of the functions of which are identified in the shaded process boxes in figure 2.2).

On completion of all particle histories (events), the accumulated data are normalised and error levels calculated. These tabulated data are output to the data file(s). Execution of both GEMAT/executable and GEMAT/shell-script then terminate.

4 Specific Requirements

4.1 Functional requirements

4.1.1 User input interface (SR 1)

SR ID	Description of Requirement
SR 1.1	A user-friendly, text-only input is required, enabling the user to specify the following inputs [SR 1.2-1.26] [UR 5.1, 5.2]:
SR 1.2	The number of layers making up the geometry, and thicknesses. [UR 1.1]
SR 1.3	The shape and dimensions of the depletion volumes. [UR 1.2, 1.4]
SR 1.4	The shape and dimensions of the contact volumes. [UR 1.3, 1.4]
SR 1.5	The materials to be simulated (composition and density). [UR 1.4]
SR 1.6	The details of each volume in the shield: material, funnel length / energy threshold, visualisation attributes, simulation cut-off length, path-length shrinking factor. [UR 1.4, 1.5, 1.6, 1.7]
SR 1.7	The incident particle type, either electron, photon, proton, neutron or ion (defined by atomic and nucleon number and excitation state of the nucleus, and charge state of the atom). [UR 2.1]
SR 1.8	The incident particle spectrum as mono-energetic, power-law, exponential, or piece-wise fit (linear, power-law or exponential) to data. [UR 2.2]
SR 1.9	The name of the source spectrum file (if an output of a previous SPENVIS program is used). [UR 2.3]
SR 1.10	The shape of the incident particle source (square or circular) and extent. [UR 2.4]
SR 1.11	The position of the source with respect to the geometry and the surface of the geometry particles are incident upon. [UR 2.4]
SR 1.12	The incident particle angular distribution as unidirectional, isotropic (*i.e.* PDF varies as $\sin\theta$) or cosine-law (*i.e.* PDF varies as $\sin\theta\cos\theta$). [UR 2.5]
SR 1.13	The normalisation factor in terms of the incident particle fluence and units (particles/cm^2 or particles/m^2). [UR 2.6]
SR 1.14	The number of primary particle histories to simulate. [UR 2.8]
SR 1.15	The Application Scenario. *(Specified as an input to GEMAT/shell-script)* [UR 3.1]
SR 1.16	The initial random number seed. [UR 3.3]
SR 1.17	The depletion volumes for which fluence analysis is required. [UR 4.1, 4.2]
SR 1.18	The energy binning scheme used for fluence output. [UR 4.4]
SR 1.19	The units of particle fluence. [UR 4.5]
SR 1.20	The layers of the geometry for which energy-deposition spectra (non-geantino source) or path-length distribution analysis (geantino source) are required. [UR 4.1, 4.8, 4.17]
SR 1.21	The energy binning scheme used for pulse-height spectra (PHS) or path-length binning scheme for path-length distribution output. [UR 4.11, 4.18]

SR ID	Description of Requirement
SR 1.22	The units of energy deposition or path-length used for PHS/path-length distributions. [UR 4.12, 4.19]
SR 1.23	The volumes for which coincidence data are required. [UR 4.15]
SR 1.24	Whether a graphics file for the geometry and particle tracks is required. [UR 5.5]
SR 1.25	Maximum duration for simulation. [UR 5.8]
SR 1.26	To provide an execution interrupt request. (*Achieved through CSV file input.*) [UR 5.9]

4.1.2 User output interface (SR 2)

SR ID	Description of Requirement
SR 2.1	Output will be written as ASCII text to stdout and to data file. [UR 5.3, 5.4]
SR 2.2	If requested, output shall also be to a graphics file (format TBD, but options are ps, gif, and/or VRML). [UR 5.5]
SR 2.3	GEMAT shall provide an estimate of the program execution time. [UR 5.7]

4.1.3 Geometry definition – construction of geometry in Geant4 (SR 3)

SR ID	Description of Requirement
SR 3.1	GEMAT shall use the Geant4 toolkit to construct the materials, logical volumes and physical volumes based on the geometry information provided by the user. [UR 1.1, 1.2, 1.3, 1.4, 1.6]
SR 3.2	material-dependant particle cut-offs shall be applied according to the user specification. [SR 1.6] [UR 1.7]

4.1.4 Source particle definition – particle sampling (SR 4)

SR ID	Description of Requirement
SR 4.1	GEMAT shall be able to treat electrons, photons, protons, neutrons and ions as source particles, based on user input. [SR 1.7] [UR 2.1].
SR 4.2	GEMAT shall sample the energy spectrum according to the user-specification [SR 1.8]. Spectrum types are mono-energetic, power-law, exponential, or piece-wise linear fit (linear, power-law or exponential) to data. [UR 2.2]
SR 4.3	If requested, GEMAT shall determine the incident particle spectrum from the output of previous SPENVIS codes (including, if requested, data from MULASSIS). [UR 2.3]
SR 4.4	GEMAT will sample the incident particle spectrum over a square or circle, of user-defined dimensions and position, incident from the +z or -z direction, as specified by the user. [SR 1.10, 1.11] [UR 2.4]

SR ID	Description of Requirement
SR 4.5	GEMAT shall sample the particle directional distribution according to the user specification [SR 1.7]. Angular distribution types shall include unidirectional at a user-defined angle of incidence, isotropic (*i.e.* PDF varies as $sin\theta$) or cosine-law (*i.e.* PDF varies as $sin\theta cos\theta$). [UR 2.5]

4.1.5 Selection of application scenario (SR 5)

SR ID	Description of Requirement
SR 5.1	The appropriate version of GEMAT/executable shall be selected based on the user-defined Application Scenario. [UR 3.1]
SR 5.2	If the scenario is not specified, a default set of physical processes shall be assumed subject to the source particle and energy (see Appendix A). [UR 3.2]

4.1.6 Particle tracking and interaction simulation (SR 6)

SR ID	Description of Requirement
SR 6.1	GEMAT shall utilise "call-backs" from the Geant4 particle processing – specifically calls to the user-defined TrackingAction and SteppingAction member functions: PreUser TrackingAction, PostUserTrackingAction and UserSteppingAction tracking algorithms to flag boundary crossings and energy deposition between boundary crossings. [UR 7.1]
SR 6.2	Variance reduction shall be applied to the interaction lengths according to user specification. [SR 1.6] [UR 3.4]

4.1.7 Histogramming of data (SR 7)

SR ID	Description of Requirement
SR 7.1	Fluence data shall be collated as a function of depletion volume, particle species, and energy. [SR 1.17, 1.18] [UR 4.2, 4.3, 4.4]
SR 7.2	Energy binning scheme for fluence shall be linear, logarithmic, or arbitrary (defined by the user). [SR 1.18] [UR 4.4]
SR 7.3	The default energy binning scheme for fluence data shall be as defined in Appendix B. [UR 4.4]
SR 7.4	Errors in fluence shall be based on Poisson statistics associated with the number of counts in each bin. [UR 4.7]
SR 7.5	The fluence data shall be normalised, with appropriate use of units, before outputting to the data file. As a default, the data shall be normalised to an incident particle fluence of 1 cm^{-2}, in units of cm^{-2}. [UR 2.6, 2.7, 4.1, 4.5, 4.6]
SR 7.6	Pulse-height energy deposition spectra (PHS) shall be collated as a function of the shield layer. [SR 1.20] [UR 4.8, 4.9, 4.10]
SR 7.7	Energy binning scheme for PHS shall be linear, logarithmic, or arbitrary (defined by the user). [SR 1.21] [UR 4.11]
SR 7.8	The default energy binning scheme for PHS data shall be as defined in Appendix B. [UR 4.11]
SR 7.9	Errors in the PHS data will be based on Poisson statistics associated with the counts in each bin. [UR 4.14]

SR ID	Description of Requirement
SR 7.10	The units of energy deposition shall be that specified by the user. [SR 1.22] [UR 4.12]
SR 7.11	The PHS data shall be normalised, with appropriate use of units, before outputting to the data file. As a default, data shall be normalised to the incident particle fluence of 1 cm^{-2}. [UR 2.6, 2.7, 4.1, 4.12, 4.13]
SR 7.12	Coincidence events shall be calculated as a function of depletion volume and threshold. [SR 1.23] [UR 4.15]
SR 7.13	Errors in the number of coincidence events shall be based on Poisson statistics associated with the counts in each bin. [UR 4.16]
SR 7.14	Coincidence events shall be normalised before they are output to data file. As a default, data shall be normalised t the incident particle fluence of 1 cm^{-2}. [UR 2.6, 2.7, 4.1]
SR 7.15	Path-length distributions shall be collated as a function of the depletion volume. [SR 1.20] [UR 4.17]
SR 7.16	Path-length binning scheme for path-length distributions shall be linear, logarithmic, or arbitrary (defined by the user). [SR 1.21] [UR 4.18]
SR 7.17	The default energy binning scheme for PHS data shall be as defined in Appendix B. [UR 4.18]
SR 7.18	Errors in the path-length distribution data will be based on Poisson statistics associated with the counts in each bin. [UR 4.21]
SR 7.19	The units of energy deposition shall be that specified by the user. [SR 1.22] [UR 4.19]
SR 7.20	The path-length distribution data shall be normalised, with appropriate use of units, before outputting to the data file. As a default, data shall be normalised to the incident particle fluence of 1 cm^{-2}. [UR 2.6, 2.7, 4.1, 4.19, 4.20]

4.1.8 General operation (SR 8)

SR ID	Description of Requirement
SR 8.1	GEMAT will determine the estimated time to completion, and output this to a file. [UR 5.7]
SR 8.2	The GEMAT simulation shall be interruptible by the user - the results obtained shall be recorded and the execution of the code aborted. [UR 5.9]
SR 8.3	If the execution time of GEMAT exceeds the maximum execution time specified by the user, the results obtained shall be recorded and the execution of the code aborted. [UR 5.8].

4.2 Performance requirements (SR 9)

SR ID	Description of Requirement
SR 9.1	TBD [UR 6.1]

4.3 Interface requirements (SR 10)

The user will interface with GEMAT through a simple (dumb) terminal, by entering ASCII commands and reading ASCII text output from the monitor or CSVfile. When operating through the GEMAT/web-page interface, input and output shall be achieved by redirection of *stdin*, *stdout*, and *stderr* to/from files, rather than through a terminal. Visualisation of particle tracks within the geometry shall also be possible using the existing visualisation functions within Geant4. The requirements are covered under [SR 1, 2] [UR 5]

4.4 Operational requirements (SR 11)

GEMAT shall utilise the Geant4 4.0 toolkit. [UR 7.1]

4.5 Verification requirements (SR 12)

The code shall be verified by comparison with set test cases for which results can be determined from analysis of detailed track data. Where appropriate, tests will be performed at module level first using appropriate test harnesses [UR 8.1].

4.6 Acceptance testing requirements (SR 13)

There shall be an acceptance test procedure [UR 8.1].

4.7 Portability requirements

4.7.1 Platform, operating system and compiler (SR 14)

GEMAT will be tested using the following platforms:

1. Sun Ultra 1 Workstation
   HD space - sufficient for installation and running Geant4 4.0
   Solaris 2.7 operating system
   Sun Workshop C++ compiler 5.2

2. IBM compatible PC
   Pentium II 500 MHz processor
   128MB memory
   HD space - sufficient for installation and running of Geant4 4.0
   Linux operating system (Redhat 6.2)
   GNU gcc compiler which has g++ 2.95.2

Geant4 guidelines state that it is a necessary and sufficient condition that the software be demonstrated to operate without error on *two* different platforms (under different operating systems). [UR 9]

4.7.2 Portability to other platforms (SR 15)

The C++ source code will be consistent with programming philosophy adopted under the Geant4 project, in order to increase the portability of the new software to other platforms supported for the Geant4 project [UR 7.2].

4.8 Quality requirements (SR 16)

SR ID	Description of Requirement
SR 16.1	GEMAT shall be developed to ESA PSS-05 standards for small software projects. [UR 8.1]
SR 16.2	The software design will be based on a thorough OO decomposition of the problem. [UR 8.2]

4.9 Maintainability requirements (SR 17)

To ensure that, wherever possible, the code continues to work following upgrade of the Geant4 toolkit, the GEMAT/executable code shall be consistent with existing code within the Geant4 toolkit in terms of the classes and member functions, and design philosophy [UR 7.1].

4.10 Other requirements

There are no further requirements listed under: Input; Resources; Documentation; Security; Reliability; and Safety.

5 System design for GEMAT/executable

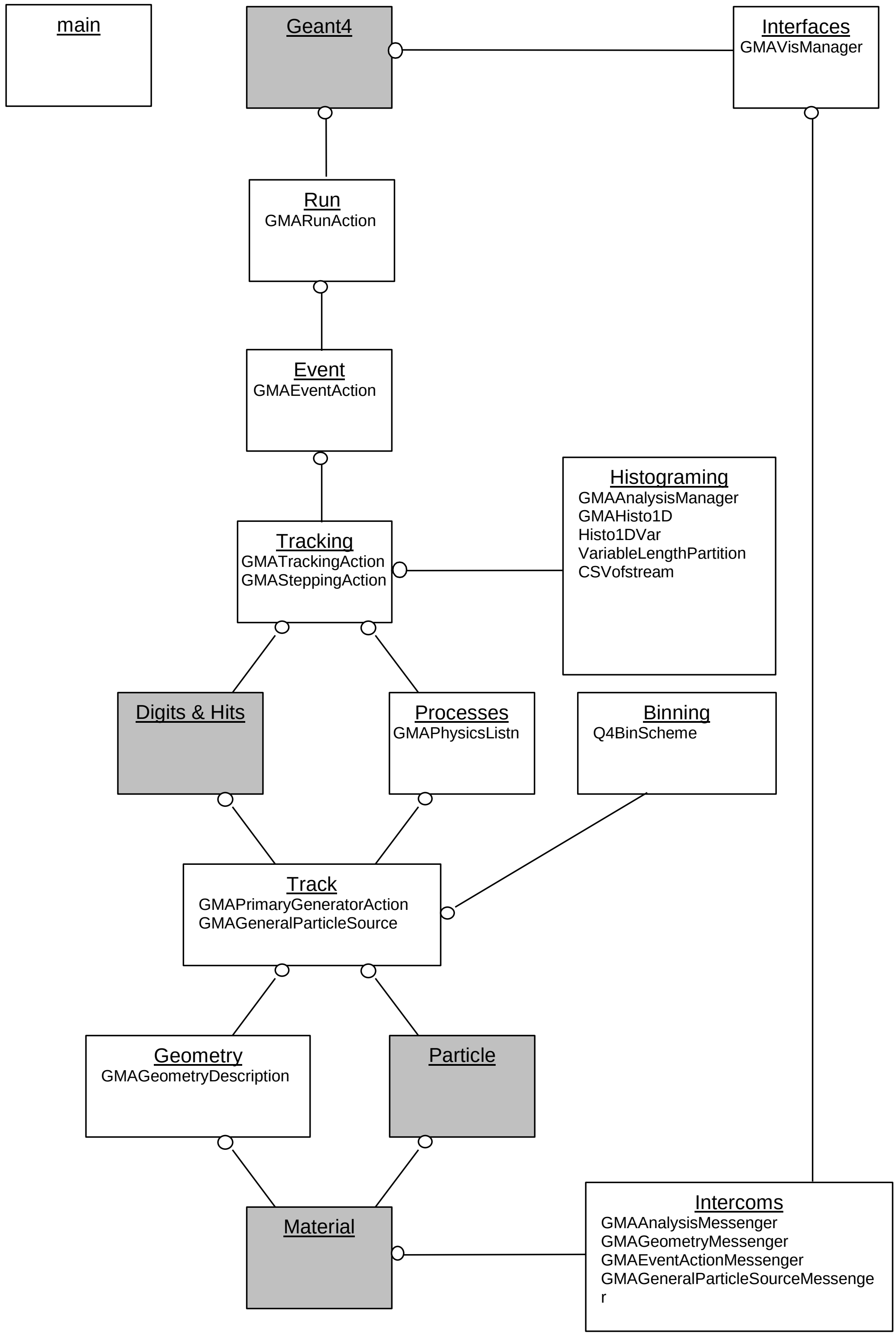

Figure 4.1: Class category diagram of Geant4 with GEMAT/executable

The diagram in figure 4.1 shows the class category diagram for the GEMAT/executable tool combined with the Geant4 toolkit. The unshaded boxes indicate the existing or new class categories required in GEMAT/executable to support the GEMAT simulation functions. Much of the structure of GEMAT/executable is dictated by the method in which the user is expected to interact with the Geant4 toolkit (see [3] Chapter 2). The Application Developer is required to provide a main program that identifies the geometry and declares the user action classes. For GEMAT/executable, the latter include the RunAction, TrackingAction and SteppingAction classes (GMARunAction, GMATrackingAction and GMASteppingAction). The Geant4 G4RunManager also utilises the GMAPrimaryGeneratorAction to generate primary particles through the GMAGeneralParticleSource according to the user-defined parameters (energy spectrum, angular distribution, *etc.*). Definition of the geometry, source and histogramming parameters is achieved through classes derived from the G4UImessenger class: GMAGeometryMessenger for specifying the geometry, GMAGeneralParticleSourceMessenger to allow control of the source particle distribution, and GMAAnalysisMessenger for specifying the histogramming/output parameters. Visualisation of the geometry and particle tracks is possible through the GMAEventAction, GMAEventActionMessenger and GMAVisManager classes.

It was decided to implement histogramming functionality using classes initially developed for the Sector Shielding Analysis Tool (SSAT). Other options considered include the use of the AIDA and hbook interfaces. However these would not provide sufficient flexibility in defining histogram arbitrary binning schemes, and/or are currently subject to significant changes in their interfaces, such that their use would significantly increase risks to this development project.

In the following sections a brief description of the GEMAT module (class) properties and processes are given.

6 Component description

6.1 GEMAT/shell-script

6.1.1 Type

Unix shell-script

6.1.2 Functions

The GEMAT/shell-script selects the appropriate GEMAT/executable (each of which has a different PhysicsList) based on the required application scenario. If no application scenario is selected, the choice of executable is based on the source particle type and energy.

6.1.3 Interfaces

Inputs to the GEMAT/shell-script include the Application Scenario and Geant4 macro-file. The shell-script invokes the GEMAT/executable, passing on the macro-file.

6.1.4 Dependencies

The GEMAT/shell-script is executed after the GEMAT/web-page has been used to collect run parameters.

6.1.5 Data

None

6.1.6 Resources

None

6.1.7 Software requirements met

[SR 1.15, 5.1, 5.2]

6.2 GEMAT/executable main

6.2.1 Type

Main program

6.2.2 Functions

The main program is required to instantiate the G4RunManager and identify the following user-classes to be used with the execution (see ref [3], Chapter 2):

- geometry;
- PhysicsList;
- RunAction;
- EventAction
- PrimaryGeneratorAction;
- TrackingAction;
- SteppingAction;

The main program also instantiates the visualisation manager (GMAVisManager) before transferring control over to the G4UI, so as to allow input of parameters associated with the GEMAT execution.

6.2.3 Interfaces

Controls subordinate processes through the G4RunManager and G4UI objects.

6.2.4 Dependencies

main() is a top-level process, *i.e.* the first process executed in GEMAT/executable.

6.2.5 Data

Data are held in the G4RunManager and G4UIsession objects.

6.2.6 Resources

None

6.2.7 Software requirements met

[SR 11]

6.3 GMAGeometryConstruction

6.3.1 Type

Class derived from G4VUserDetectorConstruction.

6.3.2 Functions

This class identifies the geometry and material construction of the system to be analysed by the tool. The construct() method first defines the elements and materials making up the geometry, then the logical volumes, and finally the construction of the physical volumes. Visualisation attributes are associated with the volumes during geometry construction.

6.3.3 Interfaces

Generates the geometry through instantiation of G4Material, G4Element, G4LogicalVolume and G4VPhysicalVolume classes. The geometry is defined using the construct() method. Parameters used to define the geometry are input via the GMAGeometryMessenger class.

6.3.4 Dependencies

Must be declared to the G4RunManager object by the SetUserInitialization() method. This object is instantiated upon initialisation of the G4RunAction using the initialize() method.

6.3.5 Data

Holds data on materials and geometry in G4Material, G4Element, G4LogicalVolume, and G4VPhysicalVolume objects.

6.3.6 Resources

None

6.3.7 Software requirements met

[SR 3.1]

6.4 GMAGeometryMessenger

6.4.1 Type

Class derived from G4UImessenger.

6.4.2 Functions

The GMAGeometryMessenger shall be instatiated by the GMAGeometryConstruction object and introduce into the UI menu commands to control the construction of the geometry:

- Materials definition;
- Geometry definition (number of layers, thicknesses, shapes and positions of depletion and contact layers);
- Association of materials and layers;
- Visualisation attributes for each layer / depletion volumes / contact volumes.

The GMAGeometryMessenger shall modify the state of the GMAGeometryConstruction object (instantiated by the G4RunAction using the initialize() method) according to UI menu commands issued by the user.

6.4.3 Interfaces

Shall use G4UIcmd objects that respond to keyboard or G4macro commands, to modify the GMAGeometryConstruction object.

6.4.4 Dependencies

Instantiated from the GMAGeometryConstruction object.

6.4.5 Data

Contains G4UIcmd objects and a pointer to the GMAGeometryConstruction object

6.4.6 Resources

UI to the terminal keyboard for input and monitor for output required.

6.4.7 Software requirements met

[SR 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 10]

6.5 GMAPhysicsList

6.5.1 Type

Class derived from G4VUserPhysicsList.

6.5.2 Functions

This class shall identify the particle types and physical processes to be used in the simulation. There shall be a range of GMAPhysicsList classes, each corresponding to a different Application Scenario (see Appendix A), and an appropriate GEMAT/executable will be selected by the GEMAT/shell-script based on the Application Scenario selected by the user.

6.5.3 Interfaces

Particle cut-off lengths, interaction path-length "shrinking" factor, funnel length and energy deposition thresholds can be modified through GMAPhysicsListMessenger, with which this class interfaces. Protected methods of the class define to the base class particles and physical processes used.

6.5.4 Dependencies

Must be declared to the G4RunManager object by the SetUserInitialization() method. This object is instantiated upon initialisation of the G4RunAction using the initialize() method.

6.5.5 Data

None

6.5.6 Resources

None

6.5.7 Software requirements met

[SR 3.2, 4.1, 6.1, 6.2, 11]

6.6 GMAPhysicsListMessenger

6.6.1 Type

Class derived from G4UImessenger.

6.6.2 Functions

The GMAPhysicsListMessenger shall be instatiated by the GMAPhysicsLIst object and introduce into the UI menu commands to control the particle cut-lengths, variance reduction factors, funnel lengths and energy deposition thresholds, and to change the initialial random number seed. The GMAPhysicsListMessenger shall modify the state of the GMAPhysicsList object (instantiated by the G4RunAction using the initialize() method) according to UI menu commands issued by the user.

6.6.3 Interfaces

Shall use G4UIcmd objects that respond to keyboard or G4macro commands, to modify the GMAPhysicsList object.

6.6.4 Dependencies

Instantiated from the GMAPhysicsList object.

6.6.5 Data

Contains G4UIcmd objects and a pointer to the GMAPhysicsList object

6.6.6 Resources

UI to the terminal keyboard for input and monitor for output required.

6.6.7 Software requirements met

[SR 1.1, 1.6, 1.16, 10]

UNCLASSIFIED - UNLIMITED
CONTROLLED DOCUMENT

6.7 GMARunAction

6.7.1 Type

Class derived from G4UserRunAction.

6.7.2 Functions

The GMARunAction shall instantiate the GMAAnalysisManager class on creation. At the BeginOfRunAction [3] it shall initialise the histograms required for the study, and then normalise and output the results at the EndOfRunAction [3].

6.7.3 Interfaces

Public methods of the class respond to the G4RunManager at the beginning and end of a run. The object also contains a pointer to the GMAAnalysisManager object.

6.7.4 Dependencies

Must be declared to the G4RunManager object by the SetUserAction() method.

6.7.5 Data

Contains a pointer to the GMAAnalysisManager object.

6.7.6 Resources

None

6.7.7 Software requirements met

[SR 11]

6.8 GMAAnalysisManager

6.8.1 Type

Class derived from G4VAnalysisManager

6.8.2 Functions

The GMAAnalysisManager object shall be a container for all histogram objects, shall control the histogramming options (the type of analysis being performed on the shield and the binning schemes applied to the histograms), and shall control the storage of data from the histogram simulation. On instantiation, it shall define the default histogramming parameters and instantiate the G4UImessenger (GMAAnalysisMessenger) to permit the user to control the definition of parameters for histogramming and outputting of the shielding data through the Geant4 UI.

This class shall also be required to create and initialise histograms to record data (in the BeginOfRunAction() member-function [3] called by the GMARunAction::BeginOfRunAction) and normalise and output tabulated histogram data (in the EndOfRunAction() member function [3], also called by the G4RunManager). During simulation, GMAAnalysisManager shall interact with the particle tracking process to update the histogram data according to the information from each particle history.

6.8.3 Interfaces

Public methods of the class shall respond to:

- G4UImessenger GMAAnalysisMessenger, following commands from the user.
- GMARunAction, GMATrackingAction and GMASteppingAction classes to initialise, accumulate, normalise and output histogram data.

6.8.4 Dependencies

Instantiated from GMARunAction, and creates GMAAnalysisMessenger the object.

6.8.5 Data

Contains and controls data on the histogramming parameters, contains the CSV output file object, and all 1-D histogram objects.

6.8.6 Resources

None

6.8.7 Software requirements met

[SR 7.1, 7.5, 7.6, 7.10, 7.11, 7.12, 7.13, 7.14, 7.15, 7.19, 7.20]

UNCLASSIFIED - UNLIMITED
CONTROLLED DOCUMENT

6.9 GMAAnalysisMessenger

6.9.1 Type

Class derived from G4UImessenger.

6.9.2 Functions

The GMAAnalysisMessenger shall be instatiated by the GMAAnalysisManager object and introduce into the UI menu commands to control the histogramming of:

- Fluence data as a function of depletion volume, particle species, and energy;
- Energy deposition spectra as a function of depletion volume;
- Coincidence events;
- Path-length distributions in materials;
- The binning schemes for histograms;
- The units for fluence, energy, and path-length used.

The GMAAnalysisMessenger shall allow specification of the incident particle fluence, which is used by GMAAnalysisManager to normalise the resulting histograms.

The GMAAnalysisMessenger shall modify the state of the GMAAnalysisManager object (instantiated by the GMARunAction) according to UI menu commands issued by the user.

6.9.3 Interfaces

Shall use G4UIcmd objects that respond to keyboard or G4macro commands, to modify the GMAAnalysisManager object.

6.9.4 Dependencies

Instantiated from the GMAAnalysisManager object.

6.9.5 Data

Contains G4UIcmd objects and a pointer to the GMAAnalysisManager object

6.9.6 Resources

UI to the terminal keyboard for input and monitor for output required.

6.9.7 Software requirements met

[SR 1.1, 1.13, 1.17, 1.19, 1.20, 1.22, 1.23, 10]

6.10 GMAEnergyBinScheme

6.10.1 Type

Class derived from Q4BinScheme.

6.10.2 Functions

The GMAEnergyBinScheme class shall be used to contain and control the binning scheme for fluence and/or PHS data collected as a function of energy. This class shall be derived from the class S4BinScheme, with modifications to restrict bin edges to ≥0 (>0 for logarithmic binning) since negative values for energy are nonsensical. An object of type GMAEnergyBinScheme shall have a default-binning scheme equal to the default binning intervals identified in Appendix B.

6.10.3 Interfaces

In addition to inherited methods, public methods shall permit definition of a minimum range for the binning scheme that is always ≥0.

6.10.4 Dependencies

Instantiated from the GMAHistograms object.

6.10.5 Data

Contains parameters associated with the binning scheme

6.10.6 Resources

None.

6.10.7 Software requirements met

[SR 7.2, 7.3, 7.7, 7.8]

6.11 GMAEnergyBinSchemeMessenger

6.11.1 Type

Class derived from G4UImessenger.

6.11.2 Functions

The GMAEnergyBinSchemeMessenger shall also be instatiated by GMAAnalysisManagerand introduce commands into the UI menu to control the binning scheme for fluence and/or PHS data as a function of energy, permitting:

- Selection of linear, logarithmic or an arbitrary binning scheme;
- Definition of the binning range;
- Definition of the number of bins (linear and logarithmic binning);
- Definition of bin positions (arbitrary binning scheme).

The information on the binning scheme shall be captured in a GMAEnergyBinScheme object, instantiated by GMAAnalysisManager.

6.11.3 Interfaces

Shall use G4UIcmd objects that respond to keyboard or G4macro commands, to modify the GMAEnergyBinScheme object.

6.11.4 Dependencies

Instantiated from the GMAAnalysisManager object.

6.11.5 Data

Contains G4UIcmd objects and a pointer to the GMAEnergyBinScheme object.

6.11.6 Resources

UI to the terminal keyboard for input and monitor for output required.

6.11.7 Software requirements met

[SR 1.1, 1.18, 1.21, 10]

6.12 GMAShieldBinScheme

6.12.1 Type

Class derived from Q4BinScheme.

6.12.2 Functions

The GMAShieldBinScheme class shall be used to contain and control the binning scheme for path-length distribution data collected as a function of path-length. This class shall be derived from the class Q4BinScheme, with modifications to restrict bin edges to ≥0 (>0 for logarithmic binning) since negative values for shielding are nonsensical. An object of type GMAShieldBinScheme shall have a default-binning scheme equal to the default binning intervals identified in Appendix B.

6.12.3 Interfaces

In addition to inherited methods, public methods shall permit definition of a minimum range for the binning scheme that is always ≥0.

6.12.4 Dependencies

Instantiated from the GMAAnalysisManager object.

6.12.5 Data

Contains parameters associated with the binning scheme

6.12.6 Resources

None.

6.12.7 Software requirements met

[SR 7.16, 7.17]

6.13 GMAShieldBinSchemeMessenger

6.13.1 Type

Class derived from G4UImessenger.

6.13.2 Functions

The GMAShieldBinSchemeMessenger shall also be instatiated by GMAAnalysisManagerand introduce commands into the UI menu to control the binning scheme for path-length distribution data as a function of shield thickness, permitting:

- Selection of linear, logarithmic or an arbitrary binning scheme;
- Definition of the binning range;
- Definition of the number of bins (linear and logarithmic binning);
- Definition of bin positions (arbitrary binning scheme).

The information on the binning scheme shall be captured in a GMAShieldBinScheme object, instantiated by GMAAnalysisManager.

6.13.3 Interfaces

Shall use G4UIcmd objects that respond to keyboard or G4macro commands, to modify the GMAShieldBinScheme object.

6.13.4 Dependencies

Instantiated from the GMAAnalysisManager object.

6.13.5 Data

Contains G4UIcmd objects and a pointer to the GMAShieldBinScheme object.

6.13.6 Resources

UI to the terminal keyboard for input and monitor for output required.

6.13.7 Software requirements met

[SR 1.1, 1.21, 10]

6.14 Q4BinScheme

6.14.1 Type

Base class to GMAEnergyBinScheme and GMAShieldBinScheme.

6.14.2 Functions

The Q4BinScheme class is used capture a binning scheme for data. Member functions shall permit:

- Selection of linear, logarithmic sine-function or sine-cosine-function, or an arbitrary binning scheme;
- Definition of the binning range;
- Definition of the number of the number of bins (non-arbitrary binning);
- Addition/deletion of bin positions (arbitrary binning-scheme).
- Selection of a default binning-scheme;

6.14.3 Interfaces

Public methods shall allow determination of the binning scheme, definition of the maximum and minimum range and number of bins, addition/deletion of bin-edges, and determination of the bin number associated with a given value.

6.14.4 Dependencies

Instantiated from the GMAAnalysisMessenger as a base class to GMAEnergyBinScheme object, and directly for the binning scheme in θ for particle fluence histograms.

6.14.5 Data

Contains parameters associated with the binning scheme

6.14.6 Resources

None.

6.14.7 Software requirements met

[SR 7.2, 7.3, 7.7, 7.8, 7.16, 7.17]

6.15 GMAHisto1D

6.15.1 Type

Class derived from Histo1DVar.

6.15.2 Functions

This class should be identical to Histo1DVar, except the insertion operator “<<” is defined so that column headings of tabulated data can be modified to the data contained by the histogram, *i.e.* fluence or pulse-height energy deposition as a function of energy, path-length distribution function and the associated Poisson errors.

6.15.3 Interfaces

In addition to base-class methods, GMAHisto1D shall interface with output stream objects ostream and CSVofstream to provide out to the terminal and CSV format output file. The GMAHisto1D object shall be contained within and acted upon by the GMAAnalysisManager object.

6.15.4 Dependencies

Instantiated from the GMAAnalysisManager object.

6.15.5 Data

Contains histogram and binning scheme data.

6.15.6 Resources

Provides output to the terminal and CSV output file.

6.15.7 Software requirements met

[SR 2.1, 7.1, 7.4, 7.6, 7.9, 7.15, 7.18, 10]

6.16 Histo1Dvar (largely based on SSAT SSD)

6.16.1 Type

Base class to GMAHisto1D

6.16.2 Functions

The Histo1DVar class shall allow definition of a 1-D binning-scheme and the accumulation of events according to that scheme. Member functions shall permit output of the bin contents and standard deviation of the events recorded for each bin. The member functions for this class should be only those required to meet the needs of the SSAT. However, where possible the data-types of the arguments and returned values shall be identical to those used in the *Histoograms* category. Information associated with the binning-scheme shall be contained in a VariableLengthPartition object.

6.16.3 Interfaces

Public methods, accessed from the GMAnalysisManager object shall permit definition of the histogram binning scheme, initialisation of stored data, accumulation of events and output of histogram data and associated errors.

6.16.4 Dependencies

Instantiated from the GMAAnalysisManager object as a base class to GMAHisto1D.

6.16.5 Data

Contains histogram and binning scheme data.

6.16.6 Resources

None.

6.16.7 Software requirements met

[SR 7.1, 7.4, 7.6, 7.9, 7.15, 7.18]

6.17 VariableLengthPartition

6.17.1 Type

Class

6.17.2 Functions

The VariableLengthPartition class shall contain information relating to an arbitrary 1-D binning scheme in terms of a sorted vector of values defining the bin edges. Member functions shall identify the bin number associated with any G4double value provided by the user.

6.17.3 Interfaces

Public methods, accessed from the Histo1DVar and Histo2DVar objects shall permit definition of the histogram binning scheme, and determination of the bin number associated with a user-provided value.

6.17.4 Dependencies

Instantiated from Histo1DVar and Histo2DVar objects.

6.17.5 Data

Contains binning scheme data.

6.17.6 Resources

None.

6.17.7 Software requirements met

[SR 7.2, 7.7, 7.16]

6.18 GMAPrimaryGeneratorAction

6.18.1 Type

Class derived from G4VUserPrimaryGeneratorAction.

6.18.2 Functions

The GMAPrimaryGeneratorAction shall be used to instantiate the GMAGeneralParticleSource object, and thereby generate one or more source particle events for transportation by the toolkit. In instatiating the GMAGeneralParticleSource, the user interface for the particle gun (GMAGeneralParticleSourceMessenger) is also instatiated.

6.18.3 Interfaces

Interfaces with the G4RunManager to define the GMAGeneralParticleSource object, and create a primary event containing a series of protons, electrons, neutrons, photons or ions.

6.18.4 Dependencies

Instantiated by the G4RunManager. Instantiates the GMAGeneralParticleSource object, and therefore the GMAGeneralParticleSourceMessenger.

6.18.5 Data

Contains a pointer to the GMAGeneralParticleSource.

6.18.6 Resources

None.

6.18.7 Software requirements met

[SR 11]

6.19 GMAGeneralParticleSource

6.19.1 Type

Class derived from G4GeneralParticleSource.

6.19.2 Functions

The GMAGeneralParticleSource performs the same functions as G4GeneralParticleSource, but also allows the user to define a source particle spectrum from a SPENVIS file (through the GMAGeneralParticleSourceMessenger). The function of this class is therefore to extract the spectrum from the previous SPENVIS run.

6.19.3 Interfaces

None

6.19.4 Dependencies

Instantiated by GMAPrimaryParticleGenerator. Instantiates the G4GeneralParticleSource and GMAGeneralParticleSourceMessenger.

6.19.5 Data

None.

6.19.6 Resources

None.

6.19.7 Software requirements met

[SR 4.2, 4.3, 4.4, 4.5]

6.20 GMAGeneralParticleSourceMessenger

6.20.1 Type

Class derived from G4UImessenger.

6.20.2 Functions

The GMAGeneralParticleSourceMessenger shall be instatiated by the GMAGeneralParticleSource object and introduce into the UI menu for the GPS commands to allow specification of a SPENVIS output file in order to read a the particle spectrum.

The GMAGeneralParticleSourceMessenger shall modify the state of the GMAGeneralParticleSource object (instantiated by the GMAPrimaryParticleGenerator) according to UI menu commands issued by the user.

6.20.3 Interfaces

Shall use G4UIcmd objects that respond to keyboard or G4macro commands, to modify the GMAGeneralParticleSource object.

6.20.4 Dependencies

Instantiated from the GMAGeneralParticleSource object.

6.20.5 Data

Contains G4UIcmd objects and a pointer to the GMAGeneralParticleSource object

6.20.6 Resources

UI to the terminal keyboard for input and monitor for output required.

6.20.7 Software requirements met

[SR 1.1, 1.7, 1.8, 1.9, 1.10, 1.11, 1.12]

6.21 GMATrackingAction

6.21.1 Type

Class derived from G4UserTrackingAction.

6.21.2 Functions

During a call to the PreUserTrackingAction member function [3], the GMATrackingAction object shall cause the GMAAnalysisManager object to initialise variables used to accumulate energy deposition and shielding information during the particle step process. The PostUserTrackingAction member function [3] shall cause the GMAAnalysisManager object to update the histograms according to the results from the single particle track.

6.21.3 Interfaces

Public methods accessed by the G4TrackingManager shall interact with the GMAAnalysisManager to initialise and record shielding data for the particle event.

6.21.4 Dependencies

Instantiated by the G4TrackingManager object.

6.21.5 Data

None.

6.21.6 Resources

None.

6.21.7 Software requirements met

[SR 6.1]

6.22 GMASteppingAction

6.22.1 Type

Class derived from G4UserSteppingAction.

6.22.2 Functions

The UserSteppingAction member-function [3] in GMASteppingAction (called when a geometry boundary is crossed) shall determine of the distance between boundary crossings, energy deposition, and position within the geometry to advise the GMAAnalysisManager object on the particle type and statistical weight (for accumulation of fluence data), energy deposition, and/or material shielding crossed in the step and material type (path-length distribution data).

6.22.3 Interfaces

Public methods accessed by the G4TrackingManager shall interact with the GMAAnalysisManager to accumulate shielding data for the particle event.

6.22.4 Dependencies

Instantiated by the G4TrackingManager object.

6.22.5 Data

None.

6.22.6 Resources

None.

6.22.7 Software requirements met

[SR 6.1]

6.23 CSVofstream (largely based on SSAT SSD)

6.23.1 Type

Class derived from the std::ofstream

6.23.2 Functions

This class identifies a file output stream where the format is intended to be ASCII text as comma-separated variables. The GMAHisto1D class overloads the insertion operator << for a CSVofstream object to output run parameters and histogram results.

6.23.3 Interfaces

Interfaces with GMAHisto1D through overloading of the insertion operator.

6.23.4 Dependencies

Created by the GMAAnalysisManager object.

6.23.5 Data

None

6.23.6 Resources

Output to the file system.

6.23.7 Software requirements met

[SR 2.1, 2.3, 10]

6.24 GMAVisManager

6.24.1 Type

Class derived from the G4VisManager

6.24.2 Functions

This class instantiates the visualisation manager and identifies the graphics device to be used by the user:

- Dawn Fukui Renderer
- VRML (for virtual reality output)

6.24.3 Interfaces

Member function RegisterGraphicsSystems() used to identify which graphics system is/are to be used.

6.24.4 Dependencies

None

6.24.5 Data

None

6.24.6 Resources

Output to the graphics device.

6.24.7 Software requirements met

[SR 2.2, 10]

6.25 GMAEventAction

6.25.1 Type

Class derived from G4UserEventAction

6.25.2 Functions

This class is required to instantiate the GMAEventActionMessenger, in order to draw event (particle) trajectories within the volume. The GMAEventAction shall also monitor the duration of execution, output the estimated time to completion, and respond to an interrupt request.

6.25.3 Interfaces

Instantiates the GMAEventActionMessenger. Public methods accessed by the G4EventManager at the beginning and end of the event (the latter being relevant to the display of completed particle histories). Provides output to CSV file of the estimated time to completion, and monitors the input CSV interrupt-request file for change in state.

6.25.4 Dependencies

Instantiated by the G4EventManager object.

6.25.5 Data

None

6.25.6 Resources

None.

6.25.7 Software requirements met

[SR 1.26, 2.2, 8.1, 8.2, 8.3]

6.26 GMAEventActionMessenger

6.26.1 Type

Class derived from G4UImessenger

6.26.2 Functions

This class shall permit additional UI commands to be included in the menu to control the drawing of event trajectories. GMAEventActionMessenger shall also allow the user to specify the maximum duration of simulation.

6.26.3 Interfaces

Shall use G4UIcmd objects that respond to keyboard or G4macro commands, to modify the GMAEventAction object.

6.26.4 Dependencies

Instantiated by the GMAEventAction object.

6.26.5 Data

Contains G4UIcmd objects and a pointer to the GMAEventAction object.

6.26.6 Resources

UI to the terminal keyboard for input and monitor for output required.

6.26.7 Software requirements met

[SR 1.24, 1.25, 10]

6.27 Remaining system requirements met

6.27.1 Geant4 UI command /run/beamOn

The number of events (incident primary particle simulated) can be changed through the existing G4UI command /run/beamOn, where the single argument provided is the number of particles. [SR 1.14]

6.27.2 Compliance of proposed implementation with other system requirements

System requirement [SR 9.1] (performance requirement) is currently unspecified, due to the dependency on the detail of the user-specified study and the performance of the intranuclear cascade code. System requirements [SR 1.25, 8.1, 8.2, 8.3], which are met, should allow the user and SPENVIS system administrator to limit the effects of over-lengthy simulations.

The remaining system requirements (verification [SR 12], acceptance and testing [SR 13], portability [SR 14, 15], quality [SR 16], and maintainability [SR 17]) can be met within the proposed architectural design provided above.

7 User requirements versus software requirements traceability matrix

Tables 7.1 and 7.2 show a cross-reference of the user requirements and software requirements. The only user requirements which are not accounted for in the software requirements are:

- UR 1.6: “The maximum number of volumes representing depletion zones shall be restricted to 20.”
- UR 2.9: "The default number of primary particle histories shall be 10,000."

There is no intrinsic reason the geometry should comprise less than 21 volumes – Geant4 should be capable of tracking particles in geometries containing thousands of volumes. The user is also at liberty to define histogramming requirements for any (or none) of the depletion volumes [SR 1.17, 1.20, 1.23], so that the memory requirements are defined by the user-defined binning-schemes, and the number of depletion volules *for which fluence/energy deposition spectra data are required*, not the number of volumes in the geometry. Since the limit of 20 depletion volumes is not intrinsic to GEMAT/executable, it is proposed that any limit should be imposed by GEMAT/web-page.

It is proposed to use the existing /run/beamOn G4UI command to initiate the simulation with the required number of particle events. This command by default only simulates one event if no argument is provided. It is recommended that GEMAT/web-page supply the argument of 10,000 to the /run/beamOn command in the macro-file, if the user hasn't specified the number of primary particle histories.

U\S	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	1.10	1.11	1.12	1.13	1.14	1.15	1.16	1.17	1.18	1.19	1.20	1.21	1.22	1.23	1.24	1.25	1.26	2.1	2.2	2.3	3.1	3.2	4.1	4.2	4.3	4.4	4.5	5.1	5.2
1.1		✓																												✓								
1.2			✓																											✓								
1.3				✓																										✓								
1.4			✓	✓	✓	✓																								✓								
1.5						✓																																
1.6						✓																								✓								
1.7						✓																									✓							
1.8																																						
2.1							✓																									✓						
2.2								✓																									✓					
2.3									✓																									✓				
2.4										✓	✓																								✓			
2.5												✓																								✓		
2.6													✓																									
2.7																																						
2.8														✓																								
2.9																																						
3.1															✓																						✓	
3.2																																						✓
3.3																✓																						
3.4																																						
4.1																	✓			✓																		
4.2																	✓																					
4.3																																						
4.4																		✓																				
4.5																			✓																			
4.6																																						
4.7																																						
4.8																				✓																		
4.9																																						
4.10																																						
4.11																					✓																	
4.12																						✓																
4.13																																						
4.14																																						
4.15																							✓															
4.16																																						
4.17																				✓																		
4.18																					✓																	
4.19																						✓																
4.20																																						
4.21																																						
5.1	✓																																					
5.2	✓																																					
5.3																											✓											
5.4																											✓											
5.5																								✓				✓										
5.6																																						
5.7																													✓									
5.8																									✓													
5.9																										✓												
6.1																																						
7.1																																						
7.2																																						
8.1																																						
8.2																																						
9																																						

Table 7.1: Traceability matrix for user and software requirements (Part 1 of 2)

U\S	6.1	6.2	7.1	7.2	7.3	7.4	7.5	7.6	7.7	7.8	7.9	7.10	7.11	7.12	7.13	7.14	7.15	7.16	7.17	7.18	7.19	7.20	8.1	8.2	8.3	9.1	10	11	12	13	14	15	16	17
1.1																																		
1.2																																		
1.3																																		
1.4																																		
1.5																																		
1.6																																		
1.7																																		
1.8																																		
2.1																																		
2.2																																		
2.3																																		
2.4																																		
2.5																																		
2.6							✓						✓			✓						✓												
2.7							✓						✓			✓						✓												
2.8																																		
2.9																																		
3.1																																		
3.2																																		
3.3																																		
3.4		✓																																
4.1							✓						✓			✓						✓												
4.2			✓																															
4.3			✓																															
4.4			✓	✓	✓																													
4.5							✓																											
4.6							✓																											
4.7						✓																												
4.8								✓																										
4.9								✓																										
4.10								✓																										
4.11									✓	✓																								
4.12												✓	✓																					
4.13													✓																					
4.14											✓																							
4.15														✓																				
4.16															✓																			
4.17																	✓																	
4.18																		✓	✓															
4.19																					✓	✓												
4.20																						✓												
4.21																				✓														
5.1																											✓							
5.2																											✓							
5.3																											✓							
5.4																											✓							
5.5																											✓							
5.6																											✓							
5.7																							✓				✓							
5.8																									✓		✓							
5.9																								✓			✓							
6.1																										✓								
7.1	✓																											✓						✓
7.2																																✓		
8.1																													✓	✓			✓	
8.2																																	✓	
9																															✓			

Table 7.2: Traceability matrix for user and software requirements (Part 2 of 2)

8 Software requirements versus components traceability matrix

Tables 8.1 to 8.4 show a cross-reference of the user requirements and software requirements. Its shows that the functional requirements and interface requirements of the software can be met with the proposed components.

For comments on [SR 9.1, 12, 13, 14, 15, 16, 17] see §6.27.2.

S\C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27 (misc)
1.1				✓		✓			✓		✓				✓					✓							
1.2				✓																							
1.3				✓																							
1.4				✓																							
1.5				✓																							
1.6				✓		✓																					
1.7																				✓							
1.8																				✓							
1.9																				✓							
1.10																				✓							
1.11																				✓							
1.12																				✓							
1.13									✓																		
1.14																											✓
1.15	✓																										
1.16						✓																					
1.17									✓																		
1.18											✓																
1.19									✓																		
1.20									✓																		
1.21											✓																
1.22									✓																		
1.23									✓																		
1.24																										✓	
1.25																										✓	
1.26																									✓		
2.1													✓										✓				
2.2																								✓	✓		
2.3																							✓				
3.1			✓																								
3.2					✓																						
4.1					✓																						
4.2																			✓								
4.3																			✓								
4.4																			✓								
4.5																			✓								
5.1	✓																										
5.2	✓																										

Table 8.1: Traceability matrix for software requirements versus components (Part 1 or 2)

S\C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27 (misc)
6.1					✓																✓	✓					
6.2						✓																					
7.1								✓							✓	✓											
7.2										✓				✓			✓										
7.3										✓				✓													
7.4															✓	✓											
7.5								✓																			
7.6								✓							✓	✓											
7.7										✓				✓			✓										
7.8										✓				✓													
7.9															✓	✓											
7.10								✓																			
7.11								✓																			
7.12								✓																			
7.13								✓																			
7.14								✓																			
7.15								✓							✓	✓											
7.16												✓		✓			✓										
7.17												✓		✓													
7.18															✓	✓											
7.19								✓																			
7.20								✓																			
8.1																									✓		
8.2																									✓		
8.3																									✓		
9.1																											
10				✓		✓			✓		✓		✓		✓								✓	✓		✓	
11		✓			✓		✓											✓									
12																											
13																											
14																											
15																											
16																											
17																											

Table 8.2: Traceability matrix for software requirements versus components (Part 2 or 2)

Appendix A Application scenarios

Table A.1 shows the original table of Application Scenarios from the URD. In tables A.2 and A.3 these Application Scenarios have been extended to include additional simulation conditions (*e.g.* whether <20 MeV neutron transport is treated with the high-energy cascade interactions) and the primary and secondary particle types treated. Table A.4 provides a cross-reference between the descriptive model name and the Geant4 class or classes that are implementations of that model.

Particle	Energy Range of source	Physics Processes	Application Scenario
Geantino	all	G4Transportation	Path length analysis (PATH)
proton	20 – 600 MeV	low-energy EM neutron_hp precompound low-energy hadron or INC[1] or kinetic[2]	Mono, trapped and solar Protons (LE proton)
	20 MeV – 100 GeV	Above + high-energy hadron	Mono and cosmic ray protons (HE Proton)
alpha	all	As above + UrQMD[3] instead of INC or kinetic	Mono, solar and cosmic ray alphas (Alpha)
neutron	thermal -20MeV	neutron_hp	Reactor, fission, D-T, D-D sources (LE neutron)
	up to ~1 GeV	As above + precompound INC[1] or kinetic[2]	Spallation neutrons (HE neutron)
electron	10 keV – 30 MeV	Std EM or low-energy EM	Mono, trapped and planetary electrons (Electron)

[1]Intra-nuclear cascade mode (based on HETC), due for release with Geant4 4.0 (December 2001)
[2]Kinetic model – release date is not known.
[3]Ultra-relativisitic Quantum Molecular Dynamics model – release date is not known.

Table A.1: Cross-reference of applications to Geant4 physics models employed.

AS	Source										Simulation conditions				Models used													
	Geantinos	Trapped protons	Solar protons and light ions	Cosmic ray protons	Mono-energetic protons	Reactor, fission, D-T, D-D neutron	Spallation neutron	Trapped electrons	Mono-energetic electrons	X-ray spectra	Low-energy neutron	γ-$e^{\pm}$-$\mu^{\pm}$	Radioactive decay	LowE electromagnetics	High-energy hadron	Intranuclear cascade	Precompound	Neutron_hp	Evaporation	Fermi break up	Fission	Photo-evaporation	Radioactive decay	Std EM interactions for hadrons/ions	Std EM interactions for leptons/photons	LowE EM interactions for hadrons/ions	LowE EM interactions for leptons/photons
0	✓																										
1		✓	✓		✓		✓									✓	✓		✓	✓	✓			✓			
2		✓	✓		✓		✓				✓					✓	✓	✓	✓	✓	✓			✓			
3		✓	✓		✓		✓				✓	✓				✓	✓	✓	✓	✓	✓	✓		✓	✓		
4		✓	✓		✓		✓				✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
5				✓	✓										✓	✓	✓		✓	✓	✓			✓			
6				✓	✓						✓				✓	✓	✓	✓	✓	✓	✓			✓			
7				✓	✓						✓	✓			✓	✓	✓	✓	✓	✓	✓	✓		✓	✓		
8				✓	✓						✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
9						✓					✓							✓						✓			
10						✓					✓	✓						✓						✓	✓		
11						✓					✓	✓	✓					✓					✓	✓	✓		
12								✓	✓	✓			✓												✓		
13		✓	✓		✓		✓							✓		✓	✓		✓	✓	✓					✓	
14		✓	✓		✓		✓				✓			✓		✓	✓	✓	✓	✓	✓					✓	
15		✓	✓		✓		✓				✓	✓		✓		✓	✓	✓	✓	✓	✓	✓				✓	✓
16		✓	✓		✓		✓				✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓			✓	✓
17				✓	✓									✓	✓	✓	✓		✓	✓	✓					✓	
18				✓	✓						✓			✓	✓	✓	✓	✓	✓	✓	✓					✓	
19				✓	✓						✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓				✓	✓
20				✓	✓						✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓
21						✓					✓			✓				✓								✓	
22						✓					✓	✓		✓				✓								✓	✓
23						✓					✓	✓	✓	✓				✓					✓			✓	✓
24								✓	✓	✓			✓	✓													✓

Table A.2: List of specific application scenarios and Geant4 models.

AS	Source										Simulation conditions				Particle types simulated											
	Geantinos	Trapped protons	Solar protons and light ions	Cosmic ray protons	Mono-energetic protons	Reactor, fission, D-T, D-D neutron	Spallation neutron	Trapped electrons	Mono-energetic electrons	X-ray spectra	Low-energy neutron	γ-$e^{\pm}$-$\mu^{\pm}$	Radioactive decay	LowE electromagnetics	Geantinos	Protons	Anti-protons	Neutrons	Anti-neutrons	K^+, K^-, K^0	π^+, π^-, π^0	Light ions	General ions/nuclei	γ, X-rays	e^+, e^-	μ^+, μ^-
0	✓														✓											
1		✓	✓		✓		✓									✓		✓			✓	✓	†			
2		✓	✓		✓		✓				✓					✓		✓			✓	✓	†			
3		✓	✓		✓		✓				✓	✓				✓		✓			✓	✓	†	✓	✓	✓
4		✓	✓		✓		✓				✓	✓	✓			✓		✓			✓	✓	✓	✓	✓	✓
5				✓	✓											✓	✓	✓	✓	✓	✓	✓	✓			
6				✓	✓						✓					✓	✓	✓	✓	✓	✓	✓	✓			
7				✓	✓						✓	✓				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
8				✓	✓						✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
9						✓					✓					†		✓				†	†			
10						✓					✓	✓				†		✓				†	†	✓	✓	
11						✓					✓	✓	✓			†		✓				†	✓	✓	✓	
12								✓	✓	✓			✓											✓	✓	✓
13		✓	✓		✓		✓							✓		✓		✓			✓	✓	†			
14		✓	✓		✓		✓				✓			✓		✓		✓			✓	✓	†			
15		✓	✓		✓		✓				✓	✓		✓		✓		✓			✓	✓	†	✓	✓	✓
16		✓	✓		✓		✓				✓	✓	✓	✓		✓		✓			✓	✓	✓	✓	✓	✓
17				✓	✓									✓		✓	✓	✓	✓	✓	✓	✓	✓			
18				✓	✓						✓			✓		✓	✓	✓	✓	✓	✓	✓	✓			
19				✓	✓						✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
20				✓	✓						✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
21						✓					✓			✓		†		✓				†	†			
22						✓					✓	✓		✓		†		✓				†	†	✓	✓	
23						✓					✓	✓	✓	✓		†		✓				†	✓	✓	✓	
24								✓	✓	✓			✓	✓									‡	✓	✓	✓

Where hadrons/nuclei are referred to, ✓ indicates nuclear processes treated (*e.g.* radioactive decay for nuclei), † indicates only EM processes treated. ‡ indicates only fluorescence processes simulated for ions.

Table A.3: List of application scenarios and particles treated in simulation.

Descriptive model	Regime	Geant4 class for physical process
High-energy hadron	>5 GeV	G4QGSMFragmentation (TBC)
Intranuclear cascade	10's MeV-5 GeV	(TBD)
Precompound	<100 MeV	G4PreCompoundModel
Neutron_hp	Thermal - 20 MeV	G4NeutronHPCapture G4NeutronHPInelastic G4NeutronHPFission G4NeutronHPElastic
Evaporation		G4Evaporation
Fermi break-up		G4FermiBreakUp
Fission		G4CompetitiveFission
Photo-evaporation		G4PhotoEvaporation
Radioactive decay		G4RadioactiveDecay
Std EM interactions for hadrons/ions		G4hIonisation G4MultipleScattering
Std EM interactions for leptons/photons	>1keV (photon) >10keV (electron)	G4ComptonScatting G4GammaConversion G4PhotoElectricEffect G4MultipleScattering G4eIonisation G4eBremstrahlung G4eplusAnnihilation G4MuIonisation G4MuBremsstrahlung G4MuPairProduction
LowE EM interactions for hadrons/ions	250eV - 100GeV	G4hLowEnergyIonisation G4MultipleScattering
LowE EM interactions for leptons/photons	250eV - 100GeV	G4LowEnergyBremsstrahlung G4LowEnergyCompton G4LowEnergyGammaConversion G4LowEnergyIonisation G4LowEnergyPhotoElectric G4LowEnergyRayleigh G4MultipleScattering G4MuIonisation G4MuBremsstrahlung G4MuPairProduction

Table A.4: Cross-reference of Geant4 physics models and Geant4 classes.

Appendix B Summary of default parameters for GEMAT output

(TBD)