RIVL: A Radiation Imager Virtual Laboratory

Richard A. Thompson, Erdogan Cesmeli, James LeBlanc, David L. McDaniel, Scott Zelakiewicz

Abstract--**A software suite for the modeling of medical imaging detectors has been constructed that uses GEANT4 [1] for particle tracking and DETECT2000 [2],[3] for the optical modeling of scintillation photons. The Radiation Imager Virtual Laboratory (RIVL) is a collection of modular, stand-alone programs that are interfaced to each other via format conversion utilities and is intended to model the energy deposition, scintillation conversion, optical photon transport, signal sensing, electronics and pulse processing. RIVL makes use of a GEANT4 application developed at General Electric Global Research called the Virtual Radiation Imager (VRAI). VRAI is based on GEANT4, a sophisticated and mature collection of C++ libraries that are commonly used in nuclear and particle physics and is seeing an increased use in the medical imaging community. With GEANT4, the physical interactions of particles are well modeled in the energy regime relevant for medical imaging. Modeling of the optical transport of scintillation photons is performed with the program DETECT2000, which enables detailed control over optical properties and is better suited than GEANT4 for modeling the transport of optical scintillation photons. Interfacing GEANT with DETECT2000 harnesses the strengths of each of these programs to create a complete model for various medical imaging applications. The simulation of the response of the photosensor to scintillation photons, as well as the logic applied to the photosensor signals to reconstruct the hit position and energy are accomplished with custom software modules written at the GE Global Research Center. .**

I. INTRODUCTION

In the design and development of a radiation detector, accurate modeling and simulation capabilities can dramatically speed up the detector development cycle, as fewer experimental prototypes are needed. Such capabilities can also enable the invention process where experimental prototypes may not be feasible.

The objective of the effort reported here is to generate a simulation suite suitable for the study of radiation imaging detectors. A typical use case scenario is the design and optimization of a nuclear or PET detector module.

II. SCOPE

As illustrated in Fig. 1, the scope of RIVL is the following processes:

1. Generation of primary events.
2. Tracking of primary event and secondary particles.
3. Conversion of the energy deposited in the detector into conversion quanta.
4. Transport of the conversion quanta to the sensitive element of the detector.
5. Response of the detector to conversion quanta.
6. Pulse processing and event logic applied to detector signals.

Process 1) is the beginning of the chain of events simulated and represents the incident events in an experiment. Process 2) is modeled explicitly with a Monte Carlo particle transport code. After step 2), the information that is available is the energy deposited in the detector material. This represents the energy of particles whose tracking was terminated due to the individual energy of those particles falling below threshold cuts. Process 3), the conversion of this energy to conversion quanta is modeled in a parameterized way via conversion yields, excess noise, and non-proportionality. After step 3), the number and locations of conversion quanta are known. Process 4) is again modeled via a Monte Carlo transport code. After step 4), the number and locations of the conversion quanta on the sensitive surface of the detector are known. Process 5) and 6) are modeled in a parameterized fashion via spatially dependent quantum efficiencies and gains, as well as event logic equations.

III. ARCHITECTURE

A modular toolkit architectural approach was chosen. Each of the processes identified above are associated with a functional module. Each functional module is a stand-alone program that takes its input from UNIX standard input and writes its output to standard output. The input and output of modules constitutes an event stream that can be written to or from disk, or piped directly to or from other modules. In this way, if it is desired to implement a different model for a given process, a stand-alone program can be developed and interfaced with the other programs. This interfacing is done via conversion utility programs that connect the modules together. In the case that a module is used for which there is no control over the interface (input and output) conversion utility programs would be written to interface the module with other modules of the suite. Such is the case for the optical modeling program DETECT2000.

IV. DATA STREAM: VDS

Since most of the functional modules implemented were custom built, with the exception of DETECT2000, a

---

Manuscript received November 13, 2002

R. A. Thompson, E. Cesmeli, J. LeBlanc, and S. Zelakiewicz are with the General Electric Global Research Center, Niskayuna, NY.

D. L. McDaniel is with General Electric Medical Systems, Waukesha, NY.

common event stream format was chosen. This format was originally implemented in the GEANT based application VRAI, which we named the VRAI Data Stream (VDS), and proved to have utility as a general-purpose event stream. VDS is a binary data stream consisting of "primaries" and "hits". The primary is the single incident radiation particle and is carried through the entire processing chain. Hits are the result of a given process and could be, for example, energy deposited by an ionizing particle, the scintillation photons created as a result of that deposited energy, scintillation photons propagated to the sensitive surface of the detector, or the response of a photosensor to those optical photons. At the beginning of every VDS is a byte order word to identify the bye order of the machine on which the data was created. Utilities exist to display to screen the contents of a VDS, or extract data to an ASCII file. An example of a VDS is illustrated in Fig. 3.

VDS is implemented as a C++ class with well abstracted data and is hence amenable to use in new modules.

V. FUNCTIONAL MODULES

The implementations of the functional modules previously identified are described below. For purposes of demonstration, results at various stages of the simulation are presented for a typical small animal detector block design. This block consists of an 8x8 array of 2x2x15 mm3 crystals that are each wrapped with a reflector material with a 95% reflectivity. For this demonstration, a model for a position-sensitive photodiode was used as the photosensor.

A. Primary Generation

The primary event generation is performed by a program called the Simple Event Generator (SEG). The output of SEG is a VDS, and, as it is the beginning of the processing chain, it has no input. SEG has the capabilities to generate primary events with the following properties:

- Parallel and originating on a plane (flood source)
- From a point source directed at a rectangular detector
- From a point source directed into a cone
- Mono-energetically, or according to a user-supplied spectrum

SEG is typically used by us as a flood or point source for 511 keV gamma rays for studying PET detectors, but has also been used at the GE Global Research Center for studying projection x-ray detectors.

With the use of a format conversion utility, primary events could also be taken from an external source such as the PET simulation program SIMSET.

B. Tracking of Ionizing Radiation

A GEANT4-based application called the Virtual Radiation Imager (VRAI) is used for particle tracking. VRAI was developed at GE Global Research and was originally conceived as a Computed Tomography system simulation. VRAI has the capability to take its primary events from an external source (like from the Simple Event Generator), and can write out data event-by-event to a VDS. A sample of an energy spectra obtained from VRAI is shown in Fig. 4. This is a plot of the total energy deposited per event without detector resolution, so the photoelectric peak shows up as a spike at 511 keV. Also evident is the Compton continuum and edge, a dip in the Compton continuum due to 90 degree scattering, the multiple scatter foot on the Compton edge, and the fluorescence escape peak.

C. Conversion

Scint is a VDS via standard input and its output is also a VDS via standard output. Scint takes each energy deposition event and determines the number of scintillation photons that results, taking into account the following effects:

Statistical (Poisson) noise in the number of photons generated
Light yield of the scintillator material
Intrinsic resolution
Non-proportionality to secondary electrons

The last two effects represent the excess in the resolution of a scintillator to the purely statistical resolution. If the non-proportionality of the scintillation mechanism to the secondary electrons (deposited energy) is known, then that can be implemented as some arbitrary function. If the electron non-proportionality is not known, then this effect can be taken into account with a single "intrinsic resolution".

D. Conversion Quanta Transport

The transport of optical scintillation photons to the detector sensitive surface (e.g. photomultiplier photocathode) is accomplished by DETECT2000. A conversion utility program was written to convert a VDS of scintillation photons (as generated by the program Scint above) to the input format expected by DETECT2000. Modifications were made to DETECT2000 to allow the data of the primary event to be propagated through the output (FATES) file. This modification has been fed back to the developers of DETECT2000 and it is expected that it will be included in the next release. Another conversion utility program was written that converts the output of DETECT2000 (FATES files) back into a VDS.

The integration of a conversion quanta transport simulation module for direct conversion detectors is currently under investigation.

An example of the optical transport in the small animal block design discussed above is demonstrated in Fig. 5. Shown is the intensity of light that reaches the sensitive surface of the photosensor for light that was generated isotropically at the center of a crystal. The crystals borders are every 2mm and are each wrapped in a 95% reflecting, 5% transmitting reflector.

E. Sensor Response

There are currently two implementations of photosensor modules. The input to these is a VDS where the hits are optical photons at the sensitive surface (e.g. photocathode), as could be taken from DETECT2000 via a conversion utility program.

The first is appropriate for the modeling of a multi-anode photomultiplier tube. This model takes into account the spatially dependent quantum efficiency of the photocathode, and the spatial dependence of the gain via user-supplied two dimensional maps. In this way the transport and amplification of photoelectrons is not modeled explicitly, but rather parameterized in terms of quantum efficiency and gain. The configuration of the PMT in terms of the mapping of photoelectrons onto the anode signals is specified in a configuration file. The output of this PMT model is the magnitude of the signal on each anode.

The second photosensor module currently implemented is a model of a position sensitive avalanche photodiode (PSAPD). In this model, there are 4 signals out of a single photosensor and the amplitude of each of those signals is a linear function of the distance of the incident photon on the face of the PSAPD. The output of the PSAPD model is the magnitude of the 4 signals as a VDS and represents the response of the sensor to the scintillation photons on the sensitive surface for a given event.

F. Event Logic

The event logic module that is currently implemented is generalized Anger logic. Any linear Anger logic can be specified via an input configuration file that contains weighting factors. The Anger logic equations are given by

$$x = \frac{\sum_i N_{x,i} \cdot S_i}{\sum_i D_{x,i} \cdot S_i} \qquad y = \frac{\sum_i N_{y,i} \cdot S_i}{\sum_i D_{y,i} \cdot S_i}$$

Where $N_{x,i}$ ($N_{z,i}$) and $D_{x,i}$ ($D_{z,i}$) are the x (z) numerator and denominator weighting factors for the signal from signal S_i. Fig. 6 illustrates an event reconstruction position map using the PSAPD photosensor model above and a linear Anger logic module. The pin-cushioning is a natural consequence of using linear Anger logic on a resistive plate type position encoding photosensor and can be avoided with the use of other event logic.

VI. SUMMARY

A suite of programs for the simulation of medical imaging detectors is under development at the General Electric Global Research Center that makes use of a GEANT4 application and the optical model program DETECT2000. This simulation software affords the capability of modeling medical imaging detectors from the point of incident radiation to a digitized signal of position and energy. The modular design of the software allows data to be accessible at each stage and allows for new models to be easily integrated. The utility of RIVL has been demonstrated in the simulation of a small animal detector block coupled to a position sensitive photodiode with Anger logic event reconstruction.

VII. REFERENCES

[1] wwwinfo.cern.ch/asd/geant4

[2] G. F. Knoll, T. F. Knoll and T. M. Henderson, ``Light Collection Scintillation Detector Composites for Neutron Detection", IEEE Trans. Nucl. Sci., NS-35 p. 872 (1988).

[3] G. Tsang, C. Moisan and J.G. Rogers, "A Simulation to Model Position Encoding Multicrystal PET Detectors", IEEE Trans. Nucl. Sci., NS-42 p. 2236 (1995).

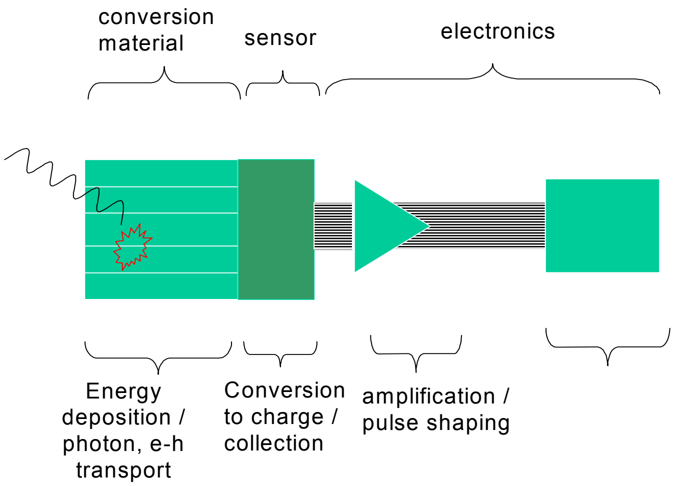


Fig. 1 An illustration of the processes that are within the scope of RIVL

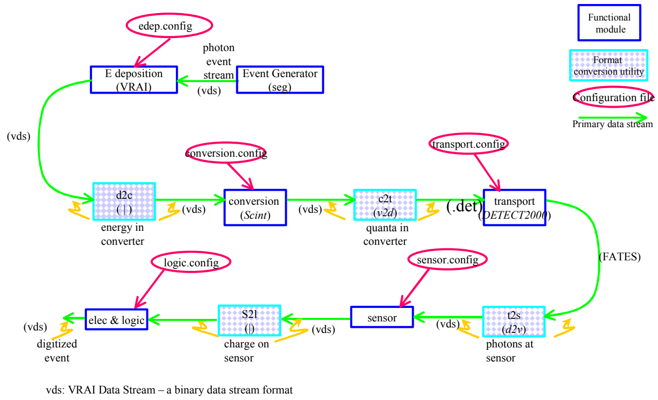
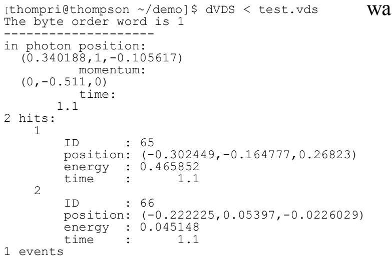


Fig. 2 The Architecture of RIVL. The basic architectural element is a functional module, with an input configuration file coupled to the next functional module via a format conversion utility. The primary data stream is always available to be written to disk or can be piped directly to the next module.



The primary event. This data is created by the event generator and is not changed by subsequent functional modules.

The hit represents the result of a given process. This information is used by subsequent modules to determine the response of the detector to the next process.

```
[thompri@thompson ~/demo]$ dVDS < test.vds
The byte order word is 1
--------------------
in photon position:
  (0.340188,1,-0.105617)
          momentum:
  (0,-0.511,0)
          time:
       1.1
2 hits:
    1
        ID      : 65
        position: (-0.302449,-0.164777,0.26823)
        energy  : 0.465852
        time    :      1.1
    2
        ID      : 66
        position: (-0.222225,0.05397,-0.0226029)
        energy  : 0.045148
        time    :      1.1
1 events
```

Fig. 3 The structure of a VDS as viewed as a screen shot of a utility (dVDS) to display to screen the contents of a VDS.

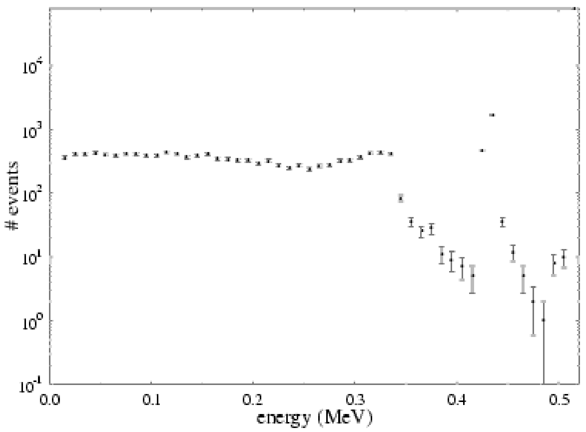


Fig. 4 A representative energy spectrum obtained from the GEANT4 based application VRAI.

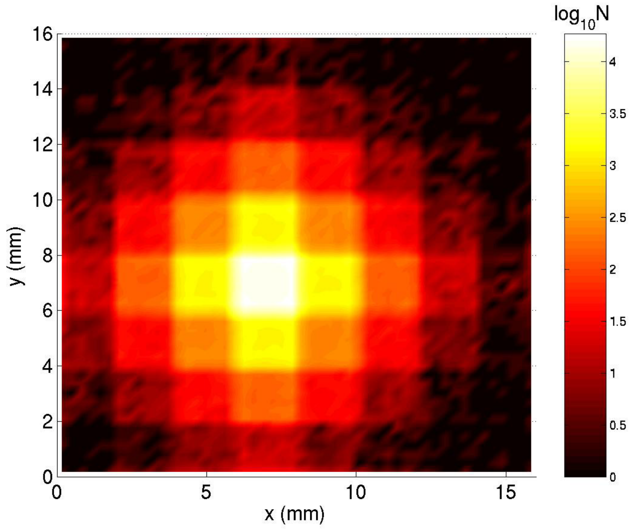


Fig. 5 The intensity of light reaching the detector sensitive surface for simulation of an 8x8 block of 2x2x15 mm3 crystals wrapped in Teflon. Light was generated isotropically at the center of a crystal.

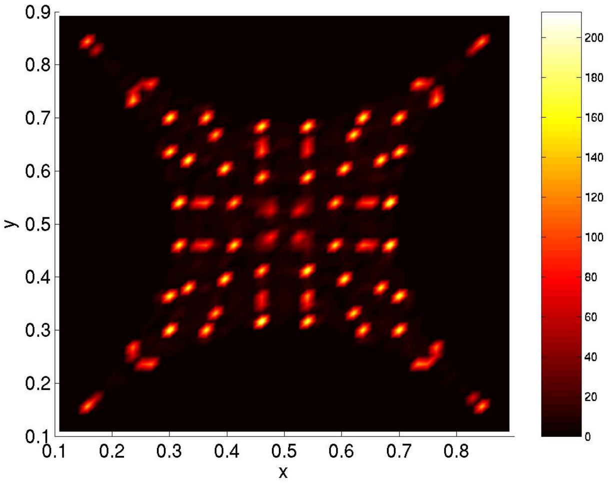


Fig. 6 A flood source position map resulting from a position sensitive photodiode model and linear Anger logic.

0-7803-7636-6/03/$17.00 ©2003 IEEE.