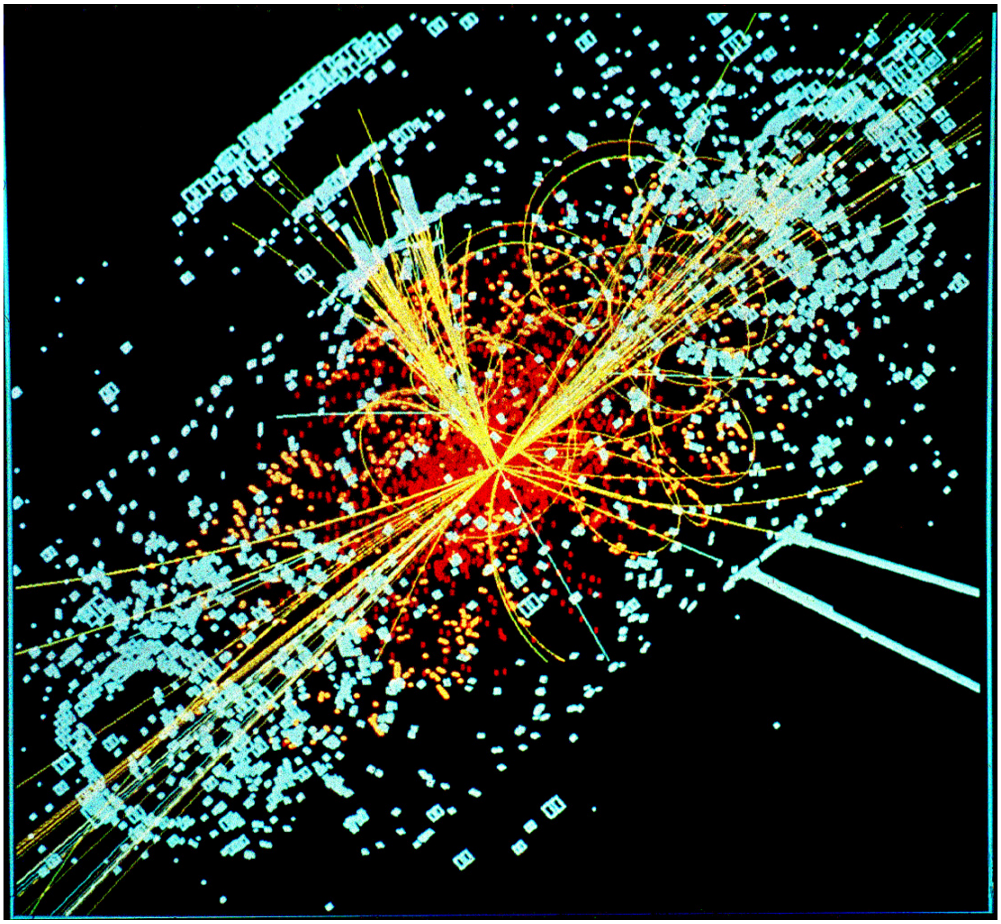
# Transforming Geant4 for the Future

## Report from the Workshop on Transforming Geant4 for the Future

September 2012



## DISCLAIMER

This report was prepared as an account of a workshop sponsored by the U.S. Department of Energy. Neither the United States Government nor any agency thereof, nor any of their employees or officers, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of document authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. Copyrights to portions of this report (including graphics) are reserved by original copyright holders or their assignees, and are used by the Government's license and by permission. Requests to use any images must be made to the provider identified in the image credits.

**On the Cover:**
Simulated Higgs Event at Large Hadron Collider (LHC)
courtesy of Compact Muon Solenoid (CMS) Collaboration

# Transforming Geant4 for the Future

## *Report from the Workshop Held May 8-9, 2012*

**Sponsored by the U.S. Department of Energy, Office of Science (SC),**
**Advanced Scientific and Computing Research (ASCR) and**
**High Energy Physics (HEP).**

*Co-Chair*, **Robert Lucas**
University of Southern California
*Co-Chair*, **Robert Roser**
Fermi National Accelerator Laboratory

*Breakout Panel*, *Multi-core Optimization*,
**Daniel Elvira**, Fermi National Accelerator Laboratory
**Robert Fowler**, University of North Carolina

*Breakout Panel*, *Scientific Data Handling and Analysis*
**Gene Cooperman**, Northeastern University
**Robert Ross**, Argonne National Laboratory

*Representative*, *Advanced Scientific Computing Research*, **Ceren Susut**
*Representative*, *High Energy Physics*, **Lali Chatterjee**

# Table of Contents

# 1 Executive Summary

Geant4 (for GEometry and Tracking) is a software toolkit developed by the high energy physics community for the simulation of the interactions of particles and radiation with complex devices. Its ability to simulate particle physics processes quickly and accurately is critical to the success of high energy/particle physics. Geant4 consumes the majority of the hundreds of thousands of microprocessor cores devoted to experimental particle physics. Data generated by Geant4 occupy most of the hundreds of petabytes of globally distributed disk storage that is dedicated to current experiments funded by Office of Science High Energy Physics (HEP) program.

Today Geant4 achieves acceptable productivity on hundreds of thousands of processor cores by running many event (i.e., collision) simulations concurrently. Nevertheless, increasing the performance of Geant4 is necessary to allow physicists to add increasingly sophisticated models of the events. This is complicated by the rapidly changing nature of commercially available computing systems, as exponential growth in the performance of individual processors has given way to exponential growth in the number of processor cores.

Fortunately, although Geant4 normally executes as a serial code, its internal architecture involves a deep stack of work items that have no mutual dependencies – in principle ideal for parallelism. Recent exploration of the applicability of SciDAC-developed tools to identifying ways to speed up Geant4 as a serial code have been very promising, pointing the way to possible short-to-medium-term execution improvements. Since Geant4 is a toolkit, and thus often used in combination with tens of thousands of lines of "user code", the way to full success must involve examination of the complete code involved in many specific user applications.

While the data generated by Geant4 has so far been effectively dealt with by the HEP community, improvements in performance and the demands of future experiments will lead to challenges on this front as well. Streamlining the analysis interface and creating new tools to manage and distribute large data sets will increase the usability of this code, especially to smaller, resource limited collaborations.

Geant4 is a "living" code under continuous development by about one hundred scientists in Europe, North America and Asia. Maintaining and enhancing the intelligibility of the code is essential, as is the coordination of US transformation efforts with those underway or planned elsewhere. The workshop participants regarded this as an exciting challenge.

Given the importance of Geant4, and the rapid changes taking place in the computing architectures that it runs on, the workshop recommends that the DOE Office of Science HEP and Advanced Scientific Computing Research (ASCR) programs initiate a joint research program to transform Geant4 as follows:

1. Apply SciDAC-developed and other tools in the hands of both computer scientists and Geant4 members to systematically improve the efficiency of Geant4 as serial code;

2. Strengthen the US efforts to refactor/rearchitect Geant4 to allow efficient mapping to a range of emerging, highly parallel platforms and lead the international scene for this while ensuring appropriate integration with the GEANT 4 collaboration;

3. Plan and implement necessary validation and testing processes to ensure reliability of physics and computing outputs as rearchitecturing efforts move forward;

4. Develop efficient I/O strategies for a parallel Geant4, including sparse access to very large data sets;

5. As a research topic, explore the possibility of using a higher level of abstraction – perhaps a domain specific language – to maintain and improve code intelligibility while automatically generating platform-dependent lower-level code;

6. Explore how to handle petabyte-to-exabyte-scale data with much lower human effort than currently achieved by HEP in distributing and using simulation-dominated data.

All of the above need to be done without significantly altering the user interface.

# 2 Introduction

The computing landscape continues to evolve as technologies improve. The modern-day computers used to support High Energy Physics (HEP) experiments now have upwards of 32 processors on a single chip, and that trend will only continue. Furthermore, new generations of computing hardware such as Graphics Processing Units (GPUs) are emerging for markets like the enhancement of end-user experience in gaming and entertainment. Their highly parallel structure makes them often more effective than general-purpose central processing units (CPUs) for compute-bound algorithms, where processing of data is done in highly parallel manner. This technology has great potential for the high-energy physics community.

One of the most important tools of the HEP community is Geant4 [1] (for GEometry ANd Tracking). Geant4 is a platform for "the simulation of the passage of particles through matter," using Monte Carlo methods. In its current incarnation, the program is not well suited to these highly parallelized computing systems.

The "Transforming Geant4 for the Future" workshop was held in Rockville Maryland, May 8-9, 2012, to strategize about how one could modify Geant4 to operate on the emerging, highly parallelized platforms of the future. The workshop was jointly sponsored by HEP and ASCR. The participants of the workshop established a path forward that will enable the Geant4 software to thrive in the emerging computing environment. The success of this program will mean not only faster simulation but also enable the architects of Geant4 to add the next level of sophistication to the physics they are modeling.

The plan developed at the Rockville workshop consists of several thrusts. The plan recognizes that current Geant4 applications almost always take advantage of the trivial parallelism represented by simulating many collisions independently. The first throughput increases will therefore come from analyzing and improving the code to make it run better, as serial code, on current platforms. While this is underway, longer term efforts are needed to refactor the code such that it can be mapped on to a range of upcoming architectures, be they single instruction, multiple data (SIMD), limited function many core, full-function multicore, or as very likely, some combination of all of these. Finally, in addition to being the greatest sink of computing cycles in the HEP community, Geant4 is also the source of most of the hundreds of petabytes of data handled by high energy physicists worldwide. Therefore, future versions of Geant4 will generate extremely large, data-related challenges, many of them typical of data-intensive scientific computing.

The above software engineering research challenges are rendered more difficult by the continuous improvements to Geant4 function and precision that are required by its ongoing and widening use. Therefore, the goal of any transformation must be to not only maintain, but improve the intelligibility of the Geant4 code as seen by its community of scientific developers and ensure validity of its physics and computing outputs.

This introduction continues with an overview of Geant4 itself, a summary of its uses, an examination of the modern computational platforms on which Geant4 must run efficiently, and a summary of the goals for transforming Geant4. The remainder of this report will then discuss the research challenges associated with adapting Geant4 to emerging computing platforms, addressing the deluge of data it generates, and doing all of this in a collaboration that is widely international in the model of most HEP experiments. Furthermore, Geant4 is now gaining wider use that extends beyond HEP scientists as other sciences and industries understand the benefits of this simulation tool.

## 2.1 Overview of Geant4

The production of enormous simulated event samples is an integral part of the analysis of data collected by High Energy Physics experiments. Simulations are also essential in designing, building, and commissioning the highly complex accelerators, detectors, software tools and computing infrastructure utilized in experimental particle and nuclear physics.

Geant4 is a toolkit for simulating the passage of particles through matter. It describes (1) the tracking of particles through a geometry composed of different materials, (2) their interactions with the electrons and nuclei encountered, and (3) the creation of other particles in these interactions. Geant4 is the fully re-engineered, object-oriented, successor to the Geant3 Fortran code. It is one of a class of particle transport simulation codes, including EGS4/EGSnrc[2], FLUKA[3], MCNP/MCNPX[4], PENELOPE[5] and PHITS[6].

Geant4 is an open source project, founded in 1994, and developed and maintained by an international collaboration of around 100 physicists, computer scientists and software experts. It was a pioneering project that successfully adopted then-modern software engineering techniques to detector simulation in high energy and nuclear physics. The Geant4 toolkit is designed to model all the elements associated with detector simulation: the geometry of the system, the materials involved, the fundamental particles of interest, the tracking of particles through materials and electromagnetic fields, the physics processes governing particle interactions, the response of sensitive detector components, the storage of events and tracks, the visualization of the detector and particle trajectories, and the capture and analysis of simulation data at different levels of detail and refinement. It offers particle interaction codes ranging from fast approximate parameterizations to the precise and resource-intensive models incorporating detailed current knowledge that are used for publishable results.

Geant4 uses stochastic processes to model the interactions that take place when particles interact with matter. This is a time-consuming process and utterly vital to distinguishing between interesting discoveries and unlikely fluctuations of known processes. Most particle physics experiments devote over half of their computing power to Geant4-based simulation. Each year, significant investments are made in computing hardware by the worldwide community to perform the simulations required to analyze existing data and plan new experiments. The total cost of this simulation (including power, cooling, housing, and operation) is many tens of millions of dollars per year. Even with this investment, the increasing statistical weight of the simulations needed by most experiments in the next ten years will require major improvements to the execution efficiency of Geant4.

Monte-Carlo particle transport in complex geometrical structures with complex physics is inherently challenging for parallelization. Geant4 itself is somewhat unique in its object-oriented design and its implementation in C++. The current implementation of Geant4 contains many features that hinder our ability to make use of modern computing platforms with parallel architectures. Of particular significance is the use of global resources and the complexity they introduce into the management of program and processing state. Such an arrangement makes parallelism through threading difficult. Other contributing factors are its heavy reliance on C++ inheritance, abstract interfaces, and organization of data structures.

## 2.2 Emerging Computational Systems for Geant4

The computational systems that Geant4 will run on in the near future will show dramatic improvements in peak performance; this will be achieved by huge increases in the number of cores as well as an evolution away from traditional von Neuman style processors. Although Moore's law [22] continues unabated, the end of Dennard scaling [17] for transistors has necessitated a fundamental shift in computer architecture. In the developed world, the market for personal computers has been saturated, and the focus of the semiconductor industry has turned to mobile, handheld devices. Thus, technology is not being optimized for single processor performance, but rather for power efficiency [13], and processors are becoming increasingly varied as they strive to satisfy competing demands for performance, productivity, reliability, and energy efficiency.

The diversity among these forthcoming machines presents a number of challenges to merely porting scientific software such as Geant4, much less achieving good performance. Extrapolating five years, we anticipate more cores per chip, and these will likely be a heterogeneous mix of processors, with a few optimized to maximize single thread throughput, while most are designed to maximize energy efficiency with wide SIMD data paths. The Advanced Micro Devices (AMD) Fusion family of Accelerated Processing Units (APU), blending Opteron CPUs and Radeon GPUs, is just the beginning. This trend will not only exacerbate today's performance optimization challenges,but also simultaneously promote the issues of energy consumption and resilience to the forefront. Moreover, as new memory technologies begin to appear (e.g., phase change, resistive, spin-transfer torque) begin to appear, computational scientists will need to learn to exploit the resultant asymmetric read/write bandwidths and latencies.

## 2.3 Goals for Transforming Geant4 for the Future

To address both the scientific and the computing challenges facing Geant4, it will be essential to bring together expertise in physics simulation and modeling with specialists in software systems engineering, performance analysis and tuning, and algorithm analysis and development, all under the stewardship of the DOE Office of Science. The overall goal is to move Geant4 into the era of multi-core and many-core computing, effectively utilizing existing and future large-scale heterogeneous high-performance computing resources for both core science simulation dataset generation and also dataset storage, retrieval, and end-user validation and analysis.

Geant4 is a world-wide project, and therefore it is necessary to work with the collaboration and the current developer and user communities to ensure useful results. To help protect the current investment in user-developed Geant4 applications and add-on libraries, it is also necessary to use a multi-phased, multiple path approach to organize this program. One of the purposes of this document is to describe the phases and paths to move towards the desired end. On the first path, the current code base can be moved forward by utilizing the DOE's significant experience in application performance analysis and source code transformations for high-performance computing. This should immediately provide benefits to the community.

Significant exploration and study is needed in order to move the Geant4 toolkit toward the long term goal of being able to utilize large-scale, heterogeneous computing systems, as it was designed for much simpler platforms.  Thus the second path will be to complete these studies and work towards a reengineering of the underlying system software framework that retains connections to crucial existing applications and physics libraries while permitting migration and transition to a new improved construction.  Rapid prototyping will be used to quantify performance gains in specific subsystems as explorations are conducted. An extensive reorganization of the underlying coordination framework will allow for more diverse processing options (e.g., GPUs) and allow for greater flexibility as computing resources evolve.

While exploring concurrency, part of the focus should  be on abstracting the overall structure of the Geant4-based codes in terms of computational flows, algorithms and models.  This analysis can be used to enable the development of code generation schemes and execution strategies targeting future high-performance architectures.  Domain-Specific Languages (DSLs) have been developed specifically to address the problem of simultaneously achieving programmer productivity and machine performance.  The development and validation of a DSL specific to the Geant4-based codes should be explored as an exciting possibility.

# 3 Transforming Geant4

Close collaboration between the HEP and ASCR communities will be vital to achieve the goal of managing the transition of Geant4 to fully exploit the looming generation of highly parallel computing devices. Low-level code optimization will draw most heavily on computer science expertise and tools developed by computer scientists, whereas simulation campaign strategies and physics simulation requirements are the prime responsibility of physicists. Both fields will benefit strongly from collaboration. This section reviews the current software architecture of Geant4. It then presents suggestions for enhancing the software including optimizing performance on individual CPU cores, multi-threaded systems, heterogeneous platforms, mathematical algorithms, and even examination of domain specific language technology. Finally, suggestions are made as to where early adoption of other ASCR research could be profitable for Geant4.

## 3.1 Current Software Architecture

Geant4 is written in C++ and is both a framework and a toolkit. It is a toolkit because it allows the user to pick and choose which of its many software libraries to use within an experiment-specific application. It is a framework because users can expand the functionality of Geant4 through its many interface points, and then configure the Geant4 system to utilize the newly introduced components. Geant4 has many customization points and places to add functionality using abstract interfaces. For example, in particle physics applications it is complemented by a comparable amount of (largely) physicist-written code to provide an execution and I/O framework, the definition of the geometry and materials of the detector to be simulated, a simulation of the initial interactions whose products impinge on the detector, and the way in which energy deposits are turned into data by the sensitive devices in the detector. The physicist-written code must also specify which physics processes should be used, particularly for hadronic interactions where speed versus precision tradeoffs is made.

Geant4 has abstract interfaces for objects representing data necessary for particle tracking, such as events, tracks, and hits (localized energy deposits). A typical user will derive his or her particular experiment-required data structures from these abstract interfaces to allow Geant4 to interact with their libraries during particle tracking.

Geant4 relies heavily on the object-oriented features of C++ for developing class hierarchies. A complex and deep hierarchy of abstract interfaces is used to access data and invoke algorithms during the tracking process. The design makes use of global data structures (e.g., singletons) for managing object lifetimes, for locating services, and accessing status data from anywhere in the application. For many of the user-level APIs, object sharing and ownership are not readily known since the more recent features of C++ allowing ownership to be distinguished from reference pointers are not used. Geant4's coordination framework has been designed with serial processing in mind, carrying one particle at a time through its processing stages. Each part of the design is consistent with the best practices for C++ programming at the time that module was designed. Recent performance-enhancing features such as template meta-programming and expression templates are not present in the current system.

It is necessary for any reasonably complex experiment to provide user code and data structures to collect energy deposits using Geant4-supplied abstract interfaces. A typical application will have user code to start and control the track propagation. There is also an API to allow for user code to handle calculating magnetic fields when needed by the Geant4 propagation engine. Detector geometry is defined by deriving new objects from Geant4 abstract interfacing, and utilizing many library elements, such as shapes, volumes, and materials. Experts may also supply new physics processes to describe what happens when different particles move through various types of material at different energies. Geant4 also has a complex configuration/parameter-management system to control and tune the tracking process.

Many complex experiments provide a geometry handling layer that is common with their production event reconstruction system. This arrangement allows for one set of code to be used and maintained for reconstruction and simulation, thus helping to ensure its correctness.

There is currently a large investment in physics process code and in validation of physics results. This large investment cannot be readily recreated or rewritten without a very large effort. Even if it were, it would require revalidation. Reproducibility of results is critical. While not an absolute requirement, bit-for-bit identity of simulated events that start from the same random number seed is highly prized. This allows validation of many types of code change (or believed lack of change) by simulating only a handful of events. Billions of events may have to be simulated to perform a statistically significant validation if the use of random numbers has changed or is unpredictable.

Many of the underlying utility libraries are HEP-community developed. This includes physics objects, small matrix manipulation, random number generation, and objects for collecting and reporting statistics. The design principles used to develop these libraries are similar to that of Geant4 as a whole, relying heavily on inheritance for abstraction interfaces and aggregation of data.

The toolkit nature of Geant4 means that optimization of the toolkit by itself can be a very misleading undertaking. More sophisticated user code, for example a magnetic field determination that makes extensive use of caching, can greatly reduce the time spent in parts of the Geant4 code that appear as hotspots in more naive applications.

Geant4 efficiency issues may be separated into three levels: **simulation campaign strategy**, **internal architecture** and **coding**. Resource usage is highly dependent on the simulation campaign strategy that dictates the models to be used, the level of detail to be simulated and the statistical precision required.

At the simulation-campaign-strategy level, Geant4 is normally required to simulate thousands to billions of independent events such as the 20 overlapping collisions every 50 nanoseconds currently produced by the LHC. Campaigns can use tens of thousands of processors for months. The independence of collisions has up to now allowed campaign strategy to rely on trivial parallelism. For example, 12-core CPU servers are exploited by running 12 independent jobs. A simple multithreaded implementation [18] of this approach is available whose main benefit is economizing on memory usage by not bringing 12 copies of static data into memory. The campaign strategy level was largely outside the scope of the workshop, apart from noting that its trivial parallelism already brings many of the throughput benefits accessible by multi-threading.

Figure 1 depicts the internal-architecture level of Geant4 code operation. Geant4 is architected to maintain a stack of particles that are being tracked as its main internal data structure. Until the point where it has lost its energy or its identity, a particle will repeatedly interact with the material it encounters, potentially producing large numbers of daughters, which themselves interact similarly. The particles on the stack can be treated independently, providing a potential source of dynamic thread-level parallelism on multi-core platforms. Restructuring of the processing for each independent particle will be needed to improve computational efficiency.

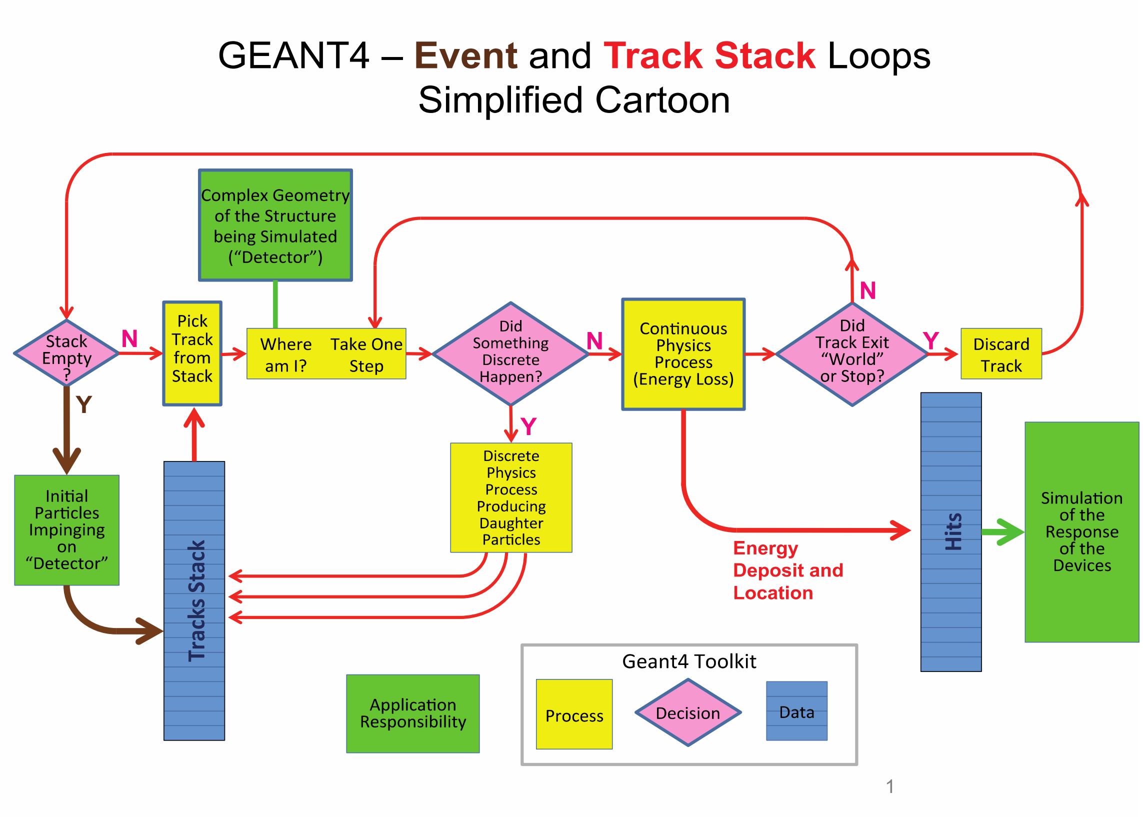

GEANT4 – Event and Track Stack Loops
Simplified Cartoon
Complex Geometry of the Structure being Simulated (“Detector”)
Stack Empty ?
N
Y
Pick Track from Stack
Where am I?
Take One Step
Did Something Discrete Happen?
N
Y
Continuous Physics Process (Energy Loss)
Did Track Exit “World” or Stop?
N
Y
Discard Track
Initial Particles Impinging on “Detector”
Tracks Stack
Discrete Physics Process Producing Daughter Particles
Energy Deposit and Location
Hits
Simulation of the Response of the Devices
Application Responsibility
Geant4 Toolkit
Process
Decision
Data
1

Figure 1: Geant4 overall computational structure.

At the coding level, given its wide use and collaborative nature, Geant4 emphasizes scientific productivity through intelligibility and maintainability over raw computational efficiency. This is justified by the large potential costs of difficult-to-maintain code. Still, the sentiment of the workshop was that it is possible to keep a high standard for productivity and maintainability while adaptively exploiting the performance potential of architectural innovations.

## 3.2 Potential Enhancements

Several preliminary analyses of the Geant4 computational structure and its execution performance have revealed areas where transformation for the application can dramatically increase its utility to the HEP community. The current system contains many features that impede efficient implementation on newer parallel architectures. Of particular significance is the use of global resources and the complexity they introduce into the management of program and processing state. Such an arrangement makes parallelism through threading difficult. It also makes program correctness difficult to assess, especially when concurrent access to shared memory structures is required. Furthermore, the depth and breadth of the object hierarchy that defines data structures and their abstract interfaces make it difficult to group similar calculations together in ways that can keep accelerators or SIMD processing engines busy. That is, the richness and flexibility of the abstract model introduces many layers of indirection that keep pipelining and vectorization from being effective. Finally, the set of user APIs restrict the coordination subsystem of Geant4 to work on small units of data, making it difficult to overcome bus and network communication latencies when data is moved between processors within a single system. Not enough work is done per subsystem or module invocation to reduce communication-related overheads to a low enough level.

A research program to address these and other such issues in Geant4 is presented in this section. Included are plans for investigation into efficient sequential execution of the low-level code, exploiting threading in symmetric shared memory systems, restructuring for emerging heterogeneous platforms, and revisiting underlying mathematical constructs. The workshop also addressed more speculative research in areas such as domain specific languages. Geant4MT, with its modest goals, does not address these issues.

### 3.2.1 Low-level code structure

One target for improving Geant4 is to locate isolated sections of code that are hotspots of computation. These code sections will likely be individual functions or small groups of cooperating functions in a limited number of classes. It will certainly be profitable to leverage many of the existing tools that were developed under SciDAC funding such as HPCToolkit [19]. HPCToolkit was instrumental to the rapid identification, development, and deployment of two such optimizations that led to a 30% reduction in runtime of two simple test applications from the Geant4 example set. In contrast, other examples, including large but simplified simulations of the LHC detectors are currently dominated by the overhead (indirection, method calls/returns). In these examples costs are distributed diffusely across large sections of code and there are no significant hot spots.

Tools like HPCToolkit can be used to discover and implement many localized performance improvements. Once all the localized hotspots have been optimized we need to go beyond local transformations to achieve further significant gains. The most likely path to significant improvements will be to examine and refactor the sequences of method invocations by applying transformations such as specialization (partial evaluation of arguments) as well as data transformations for locality. A promising strategy is the extensive use of templates and template meta-programming. Given the nature of the Geant4 code, such work must be very sensitive to the overall software engineering concepts built into this code, so as not to adversely affect its overall design and hence maintainability. A collaborative computer science and physics approach will be of paramount importance in any transformations or restructuring of the code.

### 3.2.2 Threading

To make more efficient use of modern multi-core shared memory hardware or even distributed memory clusters, particle simulation codes should be organized into multiple cooperating computational threads. The creation of particle cascades in each event can provide a ready source of dynamic thread-level parallelism. While there have been early investigations, it is not yet clear how this should be accomplished, especially given the broad range of computing platforms Geant4 will need to run on. In situations where the entire simulation space can reside in a single shared memory, information about geometry and material properties might be shared amongst threads to minimize memory requirements. On the other hand, some other particle simulation codes, such as Monte Carlo N-Particle (MCNP), distribute the simulated space across processes in a distributed memory model and transfer particles between processes and machines as they cross boundaries. Even on problems that can fit on a shared memory system, such partitioning of responsibility among threads could yield significant speedups by improving cache locality.

An in-depth investigation will be required to identify the appropriate techniques for parallelizing Geant4. Since Geant4 simulations are always run as ensembles of thousands to billions of independent events, parallelism below the event level must contribute to overall throughput improvements in addition to speeding up the simulation of individual events.

Geant4 can handle very complex and realistic geometries. For example, LHC detectors are simulated using hundreds of millions of volumes. To handle such complex geometries, Geant4 makes extensive use of caching to avoid redundant calculations. Another in-depth investigation is required here to determine how such complex geometries could be implemented efficiently using many threads.

### 3.2.3 Reengineering for Platform-Dependent Execution Strategies

This subsection describes the need for an ambitious program of work. In many cases the principles are clear, but the detailed implementation will determine whether the performance gains are massive, acceptable or negligible. The work must necessarily proceed on several fronts using rapid prototyping to understand which promising concepts are likely to bring major benefits. The worldwide community interested in or needing this work will bring many resources. The US program has an opportunity to play a leadership role ensuring that the worldwide effort is coherent, in spite of the need to explore many avenues.

To enable the use of multiple concurrent and heterogeneous execution units, Geant4 needs to be re-engineered to have at its core a concurrent particle propagation engine. Integration of the equations of motion and determination of particle interaction has very different structures and computational requirements. Thus a smart dispatcher is required that can dynamically adapt to its runtime environment, scheduling bundles of work for resources best suited for the operation to be performed. The coordination framework must also permit asynchronous interactions amongst the internal components and with the surrounding application software infrastructure necessary for event processing. Such an arrangement allows for more rare and complex particle interactions to not interfere with more common, less resource-intensive ones. It also permits non-uniform and non-Gaussian loads to be handled without problem. Incorporating HPC techniques, such as "work stealing", into internal scheduling components will promote efficient use of all available resources, including systems with multiple accelerator components.

Figure 2 depicts a possible structure for such a concurrent particle propagation engine. A track bundle is a set of tracks that can be conveniently and efficiently scheduled to go through the same set of processes which could potentially include propagation. The role of the track dispatcher is to assemble the tracks in bundles in such a way as to maximize the efficiency of the processes that run on them. The dispatcher also selects which processor or group of processors or cores, for example a GPU enhanced PC cluster, to schedule the execution of the processes for a specific track bundle. In order to leverage the on-chip and on-core caches of general purpose CPUs, the dispatcher might decide to associate a set of geometrical elements with a specific processor/core and to send to this processor/core only the bundle of tracks that are traversing these geometrical elements. It might also gather together tracks that need to go through processes that have been optimized for GPUs and send them to one of the graphics cards. Once the tracks are done going through the simulation, they are added to another stack, which can also be handled in parallel, to run processes that require all the finished tracks for a given simulation event, for example to digitize the energy deposit on the detector elements. Figure 3 gives an example of a parallel processing framework in which the propagation engine of Figure 2 might be embedded.

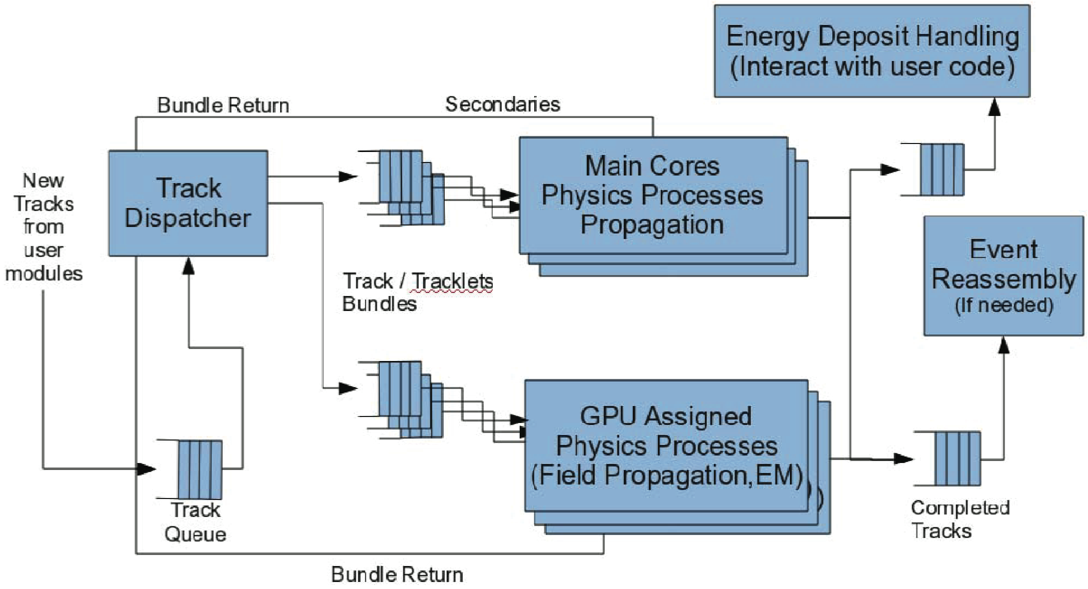

Energy Deposit Handling
(Interact with user code)
Bundle Return
Secondaries
New Tracks from user modules
Track Dispatcher
Main Cores Physics Processes Propagation
Track / Tracklets Bundles
Event Reassembly
(If needed)
GPU Assigned Physics Processes (Field Propagation,EM)
Track Queue
Completed Tracks
Bundle Return

Figure 2: Simplified view of Geant4 re-engineered processing entities.

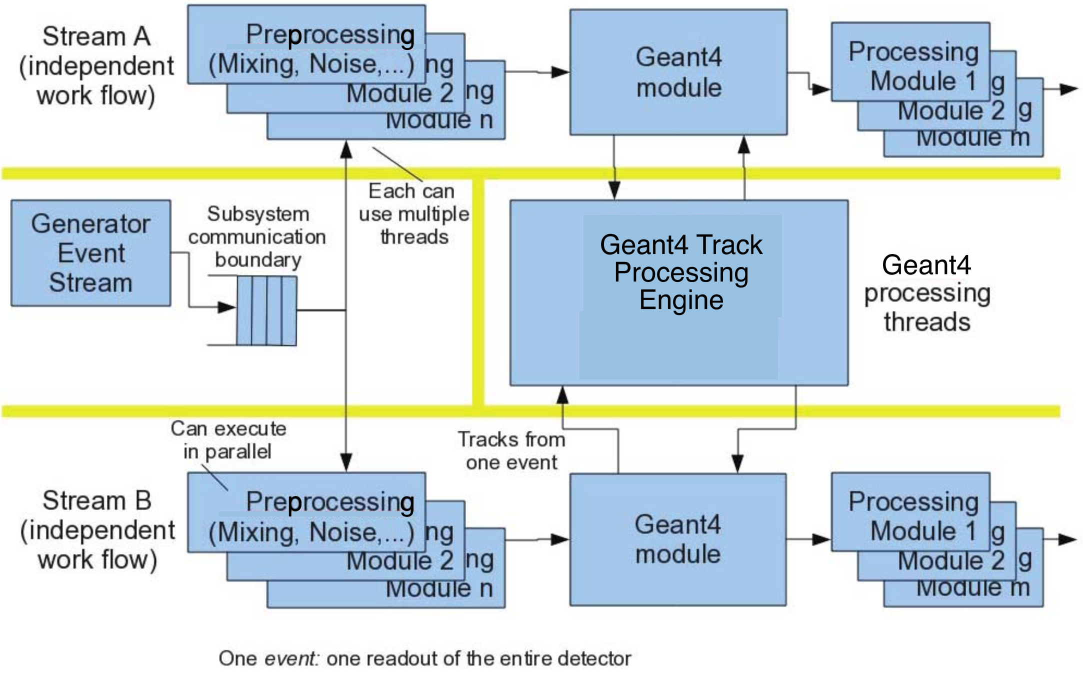


Figure 3: Geant4 embedded in a multithreaded event processing framework.

It will be important to leverage existing efforts that concentrated on analyzing the particle propagation components. This includes the definition and construction of apparatus geometry, material properties, sensitive volume descriptions, and magnetic field organization. All these elements are critical for efficiently solving the equations of motion by stepping a particle through the defined material. In particular, we will study how well these elements can be implemented onto accelerator hardware.

Current detectors are large and complex and must often be modeled in detail. Careful study will be needed to best utilize the memory hierarchy and parallel processing capabilities of the target hardware architectures. Geant4 already contains a very successful “Smart Voxel” system that optimizes navigation for a wide variety of complex geometries. Further enhancements appropriate to the new propagation engine will need to be explored, prototyped, and studied. Methods for precisely calculating the magnetic field using interpolation functions will need to be determined. Quickly answering computational geometry point location queries is always important to reducing tracking time. The latest Geant4 propagation system must be used as a model for prototyping work, as well a basis of comparison.

Real-world Geant4 program traces can be used to drive prototype propagation components. A fundamental change in strategy will be required, removing the event-by-event and particle-by-particle processing ensconced in the current framework. A new strategy will remove these boundaries and allow large particle vectors to be assembled across event boundaries where this would enhance processing efficiency on the target hardware architecture.

Since user code is involved at the higher levels during particle propagation, special care must be taken to allow maximal concurrency. Higher-level event grouping will still involve out-of-order processing. The need for efficient collection of tracking results will require advances in applications in which Geant4 is embedded. We envision that the simulation provided by the Geant4 library will often be embedded in larger frameworks which are themselves heavily parallelized. In this context, Geant4's resource usage needs to be configurable by the larger framework.

A key aspect of this work is adherence to science constraints. Two important ones are validation of the physics and reproducibility. Validation tells us that the modules we are working with give answers that are in statistical agreement with experimental measurements, or at least in agreement with answers from the previously validated code. Reproducibility means that results can be regenerated within a tolerance defined by the scientific community of users – bit-for-bit agreement where possible. Both validation and reproducibility are likely to be affected by parallelism and non-deterministic processing characteristics. Reproducibility will need to be precisely defined by working with the physics collaborations. One area of difficulty will be in random-number stream management (this is a Monte Carlo simulation). A strategy will be defined to prevent bottlenecks where random numbers are used and to record necessary seeds to allow replication of results. Deviations in results, including measurement uncertainties will be studied. Results from ASCR funded research such as ROSE [24] and Active Harmony [16] may be able to assist in these studies by probing targeted function calls and other bodies of code.

### 3.2.4 Improving Geant4 Algorithms

Although not related to the issues of parallelism and multi-threading, speeding up the low- and medium-level algorithms used in Geant4 can be expected to improve performance significantly. Low level algorithms, including the commonly used math functions *exp*, *log*, *sin* and *cos*, are typically among the most frequently called in any Geant4 application. Thus an effort to improve these elementary functions is worthwhile. One often repeated calculation in Geant4 processes is the direction of the particle, which requires both sin and cos of a given angle. Many systems and compilers include a unified *sincos* intrinsic that computes both values for essentially the cost of computing either one. Geant4 should use this where possible. Vectorized versions of trigonometric functions that compute *sin* and *cos* for several angles can also increase pipeline utilization.

Examination of the *exp* and *log* functions might also reveal time savings. In fact it would be useful to examine all of the most-used functions in the vendor-supplied implementations of C++ standard library to see if state-of-the-art methods have been, or can be, applied, and whether vectorization is a possibility.

Medium-level algorithms such as stepper/integrators and root solvers were written by Geant4 developers, and while effort was expended to make them efficient, they might not be state-of-the-art. Geant4 currently provides eight different stepper/integrators, including various orders of Runge-Kutta. Issues worth studying include testing the stability of each integrator over a track, and determining which stepper is best or fastest for a given class of Geant4 applications.

Root solvers are most frequently used to find the projected intersection of a track with a volume boundary.  Since Geant4 offers many non-trivial volume shapes, these solvers can consume a large amount of CPU time and it would be useful to seek out the best available codes or examine existing code for efficiency.  Related to this is a feature of Geant4 tracking that requires smaller steps as a track approaches a boundary.  It may be that the number of such steps could be reduced if an algorithm were found to approach the boundary in an optimal way.

Another obvious place to look for speed and algorithm improvements is the Geant4 physics code, especially hadronic physics where CPU-intensive nuclear models are used. As improvements are made, it will be crucial to ensure that physics performance  does not suffer. For this reason both CPU and physics bench- marks must be established and compared to over the course of this work.

In all of the above, there would be profit from a close working relationship between physicists, who developed the code but are not experts in algorithm optimization, and mathematicians and software engineers who can use state-of-the-art expertise to evaluate and guide the improvements.  Performance gains cannot come at the expense of physics precision.

### 3.2.5 Domain-Specific Compilation Issues

As stated above, one of the major challenges will be to eliminate overheads associated with abstraction hierarchy while maintaining the familiar and flexible interfaces required by the community.  Going beyond traditional restructuring methods, we suggest exploration of a domain-specific compilation framework, possibly with language extensions, to automatically tune and refactor code on critical paths. T he first part of this would be the application of more recent features and usage patterns of C++ such as templates and template meta-programming a smooth transition between today's Geant4 and a transformed one. The aspects highlighted in the previous sections, namely, performance tuning and exploration of design/performance tradeoffs are notoriously difficult to manage from a perspective of maintainability and thus programmer productivity. If the use of templates is insufficient, the next step is an "embedded domain-specific language".  That is, the input program is still legal C++, but an enhanced compiler would recognize the semantics of idiomatic usage in Geant4 and generate highly-optimized C code for those idioms. The third stage in this strategy would be to define and compile custom language extensions.  At all levels, users would continue to program at a highly-abstract level while compilers would deal with the low-level code generation issues.  The domain-specific compilation and languages approach would facilitate the generation of hardware independent source code for future Geant4 applications that would be custom compiled for a broad range of future target architectures.

Note that a DSL approach as described here is by no means certain to succeed. Furthermore, it will probably require a coordinated effort to define new sets of abstractions that will be useful to the physics community while being designed to compliable in the efficient and portable code.  Thus, at least initially, only the exploration of the potential of a DSL should be a goal to a joint HEP-ASCR research project.

## 3.3 Early Adoption of ASCR Research

For decades, ASCR has been investing in the research and development of new mathematical and computer science algorithms and software.  Much of this research has been in collaboration with HEP. It has also been motivated by challenges that have arisen in research conducted elsewhere in the DOE Office of Science, the National Nuclear Security Administration, and the evolution of the computing systems available to DOE. Automatic performance tuning, uncertainty quantification, enhancing resilience to soft errors, and software-guided reduction of power consumption are all examples of research that is maturing to the point where it should be considered for inclusion in any effort to transform Geant4.

Performance portability, as one migrates code from one computer to another, is a tedious and often error-prone process. One makes innumerable experiments, often changing the unrolling and tiling of loops, in an effort to optimize throughput for a particular microprocessor and the compiler and library versions available. This is exacerbated today by the proliferation of the number of architectures that a code like Geant4 must run on. These include individual microprocessor cores, shared memory multiprocessors, SIMD arithmetic extensions, SIMT graphics co-processors, and heterogeneous mixtures thereof. To alleviate computational scientists of this burden, ASCR has funded research in the SciDAC-2 Performance Engineering Research Institute and the current SciDAC-3 Institute for Sustained Performance, Energy, and Resilience. The focus of work in these Institutes is to automate these performance-tuning experiments. The tools and skills developed as a result of these efforts should be applied to transforming Geant4.

ASCR has also made significant investments in the development of theory, algorithms, and software tools for uncertainty quantification (UQ). Accounting for uncertainties in both empirical measurements and computational modeling is a prerequisite for well-founded validation of models in a wide range of physical systems. ASCR-funded UQ research is focused on the improvement of uncertainty quantification algorithms and software tools, with particular emphasis on scalability to large-scale computational problems/platforms. In experimental HEP, uncertainty (a.k.a. systematic error) quantification is the most effort-intensive part of any analysis. The use of general UQ methods and tools in the Geant4 and broader data analysis context is well worth exploring.

As the semiconductor devices which comprise modern microprocessors continue to shrink in size, and operate ever closer to threshold, the noise margins are narrowing. Soft errors, in which a bit takes the wrong state for as little as one clock period are expected to become increasingly common. ASCR researchers have begun to investigate how applications such as the physics simulations built with Geant4 that are important to HEP can continue to exploit commercial, state-of-the-art processors, yet address those transient errors that would likely go unnoticed in consumer electronics. Applied mathematicians have studied algorithmic based fault tolerance schemes. These range from adding extra state to matrices for error detection and correction, to algorithms such as iterative refinement which can also correct errors that manifest themselves as a modest loss of precision. Computer scientists are injecting faults into applications to understand when they can be tolerated, and developing compiler technology to automatically add redundancy to detect and correct those faults that cannot be accepted. Prototypes of this technology will be available to Geant4 developers in the next few years, and should be explored to minimize HEP's vulnerability to increasingly unreliable computing platforms

For decades, the power consumption of computers was relatively constant. The switch two decades ago to parallel processing with large ensembles of CMOS microprocessors increased the power consumption by an order-of-magnitude. Today, after the end of Dennard scaling, power consumption is increasing further, and with it the cost of provisioning power is approaching the cost of the computers themselves. This is a problem that faces the entire computing industry, ranging from suppliers of mobile devices, to those of the largest servers. Dynamic voltage and frequency scaling allow processor cores to reduce their power consumption as a function of the load they are observing. However, they cannot anticipate how that load will change. ASCR funded research into application and system modeling can, and it has demonstrated an O(10%) reduction in power consumption for some workloads [14]. This technology should be applied to Geant4 and the HEP applications built from it.

# 4 Geant4 Data Issues and Challenges

High precision in a Monte-Carlo simulation inevitably means large ensembles and very big data sets. Whether running on hundreds of thousands of distributed CPU cores for the LHC today, or quite possibly on a leadership-class supercomputer in the future, data storage, access and management present increasing challenges. A majority of the 300 petabytes of disk space used by the LHC program today stores data generated by Geant4. Simulated and measured data must be used together to obtain physics results whose statistical uncertainty is then the sum (in quadrature) of the statistical uncertainties of the simulated and measured events. Although simulation is expensive, an HEP accelerator and its experiments cost orders-of-magnitude more, so it is not uncommon to produce 10 to 100 times the amount of simulated data than one has of actual data for a particular physics analysis.

This is not simply a matter of simulating 10 times the total volume of the acquired data. To be sure of capturing and identifying the interesting events, many less-interesting collisions are recorded. The multiplication factor applied to simulated events is heavily skewed towards simulating the interesting ones. Nevertheless, a detailed simulation of all potential backgrounds to interesting events is essential. A balanced approach ends up requiring an average of two or three times more simulated than measured events and a simulation-dominated data management challenge.

Many issues and challenges related to large-scale data were examined by the workshop. These included the CAD interface for providing geometry to Geant4, parallel I/O, sparse access to data, distributed data management, workflow and provenance, data visualization, and data analysis. These research challenges are examined on more detail below.

## 4.1 CAD Interface to Geant4 Geometry

Most Geant4 applications encode the geometry to be used in the simulation via a C++ API, giving access to many analytically described shapes and features such as cloning shapes or allowing Boolean operations of shapes. The flexibility and capability of this way of encoding geometry is ideal for highly complex, but stable physics experiments. It is less ideal in other fields, such as aerospace and medicine, where many “what-if ” studies may be required and the natural source of the geometry is often a CAD system.

This is not a new requirement, and while there have been some attempts to solve this issue, none have realized anything like the full potential of the CAD approach. The FASTRAD [7] software, commissioned by ESA, has been available since 2001, but fails to meet many users’ needs due to its reported combination of limited function, bugs and an expensive license. All existing approaches fail to automate the handling of materials, volume hierarchy, or to use efficient geometrical models, instead of using a lowest-common-denominator tessellated solids approach.

Currently, optimizing a Geant4 description of an HEP detector for efficient execution is mainly a test of the skill of the programmer encoding the geometry, although some optimizations are performed by Geant4. Research into the extent to which a “dumb” encoding of geometry could be optimized algorithmically would be an exciting additional topic.

## 4.2 Parallel I/O

Parallel I/O is becoming a challenge for simulation, data processing and data analysis in HEP. The data generated by large ensembles of Geant4 jobs are already beginning to stress the data-access, persistency and data-management mechanisms of the file systems of HEP computing resources. Future highly parallel platforms will almost certainly bring I/O bottlenecks that are likely to have a major impact if used by a transformed Geant4 in today's simplistic manner.

Many of the challenges relate to the fact that the optimum data layout and packaging differs for writing, storing, transmission, serial reading and sparse analysis. Today, simulation output is usually written in a one-size-fits-all manner, mainly driven by ease of writing with some concessions to the most probable forms of access.

Geant4 would benefit from research required to optimize I/O in heavily multithreaded systems. In particular, it is important to understand what data-organization tasks can be achieved by the simulation itself and what tasks will be more efficiently performed by separate IO runtime services focused on data collection and organization.

## 4.3 Data Access

The HEP data analysis tasks associated with Geant4 are increasingly challenged by the rate at which they can access data. Although simulated data dominate HEP storage, it is the real rather than the simulated data that is most challenging in analysis. Real data contains vast numbers of different types of primary collision, and access to the data wanted for any particular analysis is normally very sparse. This sparsity can be offset to some extent by long, resource-intensive "skimming" that produces concentrated samples of particular types of events. In contrast, simulated data has a predetermined physics content and can normally be accessed serially.

One fairly specific simulation activity does present major data-access challenges. This is the overlaying of a simulated event on top of one or more measured events. At the LHC there are currently many tens of proton-proton collisions in each bunch crossing, and scientists are expecting upwards of a hundred as luminosity grows. Most of these collisions are very "soft", involving only a small momentum transfer between the colliding protons, but nevertheless producing tens to hundreds of particles for each collision. The new physics sought at the LHC should involve "hard" proton-proton collisions that occur at an average rate of well below one per bunch crossing. Thus when a detector triggers on the products of a hard collision, it will also record the products of 20 or more soft collisions.

The simulation of the exciting hard collisions must correctly reflect the confusing effects of the tens to hundreds of overlaid soft collisions. By far the most correct way to do this is to overlay real measured collisions taken with a "minimum-bias" trigger that records a fraction of the typical bunch crossings. The data access challenge then becomes to select the minimum bias events randomly from a data sample that is far too large to reside in random access memory.

## 4.4 Distributed Data Management

Geant4 simulations are not the unique source of HEP's distributed data management challenge. Experiments also generate tremendous volumes of data. However, simulated data do dominate and are a principal driver of the major successful efforts made within HEP to create a functional distributed data management and processing system.

Current HEP systems support a combination of policy-based worldwide data distribution and automated usage-based data distribution. A complementary capability supporting worldwide data access from executing jobs is now in production test. All systems have a major focus on robustness and recovery from errors and also provide very serviceable monitoring and diagnostics, allowing, for example, a physicist doing an ATLAS Distributed Computing Operations shift to drill down from seeing a slightly elevated failure rate in US-France transfers to the probable root cure of the problem – perhaps an overloaded metadata server at a particular site.

Although this approach has been effective so far, the HEP community recognizes that its distributed data management needs to become much less dependent on physicists on shift in the future. It is sufficiently developer-and operator-intensive that it cannot be recommended, in its present form, as a solution for HEP collaborations of "only" a few hundred physicists, let alone less massively collaborative science. An in-depth examination performed by computer scientists and physicists of HEP's current solutions for simulation-dominated distributed data management is very strongly recommended. The resultant understanding likely point the way to new developments that could benefit other sciences and allow physicists in HEP to spend more time on physics analyses.

## 4.5 Workflow and Provenance

The data generated by Geant4 must be distributed world-wide to the HEP community. This in turn required the creation of provenance tracking and resource-aware workflow management capabilities. These systems handle over 100,000 simultaneous jobs, plus worldwide data movement at rates of 100s of terabytes per day. However the more intimate workflows that take place when a group of physicists performs simulation-intensive study and prepares all the data needed to support a publication are less well captured. Replication of published results should ideally only require access to the automatically captured record of provenance and workflow. Like many or most fields, HEP is far from attaining this goal, but recognizes its importance, especially in the context of experiments that mankind may not repeat in the foreseeable future.

We identify two related paths for computer science and HEP to work together on these issues. The first is to study the "effective, if still far from perfect" large-scale systems created by HEP to understand how they can inform, and be informed by, the current state of the art in computer-science-led workflow and provenance. The second path is to study the largely unstandardized and often unautomated way in which publishable results are obtained using the simulated and real data produced by the worldwide data management and processing systems. Is this creative, ideas-driven process of deriving publishable results amenable to provenance and workflow tools that would empower creativity rather than constrain it? If such tools would bring benefit, would they be generalizable to a wider range of sciences? On both paths, "study" implies some level of development and collaborative deployment of prototype tools.

## 4.6 Data Visualization

Data Visualization challenges in Geant4 flow from the extreme diversity of the uses to which the Geant4 toolkit is applied. Some users rely on visualization tools driven directly from

Geant4 (the OpenGL family of tools, the Inventor tools, ray tracing tools), other users rely on third party visualization tools driven from files produced by Geant4 (HepRApp, DAWN, VRML, gMocren) while still other users make their own visualization tools that live in their own overall simulation frameworks. To this end, Geant4 visualization uses a layered approach with well-defined interfaces, such that a wide variety of tools can be driven by the Geant4 visualization kernel. Geant4 users are often trying to see through dense data to find a "needle in a haystack." They use visualization to debug the minutiae of complex geometries, to find a one in a billion bug in tracking triggered by glancing incidence on a complex volume boundary, to see exactly where energy is deposited, or to produce beautiful three-dimensional renderings for outreach to general audiences.

The existing visualization tools in Geant4 serve most users very well and there is no case for dedicated computer-science research to transform visualization. However the layered approach of Geant4 visualization does provide excellent opportunities to exploit other visualization solutions from elsewhere in the DOE complex.

## 4.7 Data Analysis

The analysis of data from Geant4 is another area that was discussed to complete the data landscape, but does not demand dedicated computer-science research at this time. Simulation results are among the largest and most complex data sets in the world. Their analysis stretches the capacities of existing data analysis tools. Because Geant4 has such diverse user communities, the toolkit does not support just one data analysis system but instead allows the user to specify their system, from tools developed specifically for HEP, such as ROOT and AIDA, to more general solutions such as comma-separated values files. The choice of what data to save is unique to a given simulation, with some simulations wanting to save only the most basic quantities (total energy deposited in a given structure) and others wanting to save details of every tracking step. Providing efficient pathways to pass data out of Geant4 and into analysis tools, in a manner flexible enough to serve such diverse user requirements, is an area for continuous improvement.

# 5 Conclusions

Geant4 is an important software tool for the HEP community as well as many other communities. As experiments evolve, the demands on both the amount of simulated data as well as precision of the simulations will grow. Geant4 must evolve in order to both take advantage of the emerging computing technologies and to be able to meet the future simulation demands in a cost effective way. Furthermore, this evolution of Geant4 will enable HEP experimental scientists to take advantage of current leadership class computing that is available but not utilized by this community.

Transforming Geant4 will require research to overcome a broad range of challenges. Workshop participants articulated these in the preceding sections. They fall into five broad categories: systematically improving the sequential performance of Geant4; refactoring the code for emerging, highly parallel computing systems; improving data access, management, and analysis; exploring novel new programming abstractions, and reducing the human effort required to handle the upcoming, Exabyte-scale, globally distributed, data sets generated by Geant4.

A joint partnership between HEP and ASCR scientists will allow the US to take the leadership role in modernizing this important tool. ASCR and HEP have a long and successful record of collaboration, both at the grass-roots scientist-to-scientist level, and between the programs in DOE. The SciDAC program is an excellent example thereof, with ASCR-HEP collaboration on numerous projects ranging from computational study of the universe for its dark energy and dark matter composition, lattice gauge theories adding precision to the standard model and probing beyond it, to advanced computing simulations for accelerator modeling.

The workshop showed very clearly that there is strong interest on the part of computer scientists in the ASCR community and physicists in HEP community to work together on the many challenges posed by Geant4-based simulation. There appear to be opportunities for near-term performance enhancement on individual processors, as well more profound, long-term changes that would lead to a new Geant4 that effectively exploits a variety of future multicore systems. The participation of both ASCR and HEP communities will come in two ways – those who understand the current algorithms and want to modernize the current approaches and those who will take advantage of the increased speed to then improve the precision modeling The leadership of such a multidisciplinary team should be structured in the SciDAC model and reflect a balance of the participating HEP and ASCR research communities.

Taking Geant4 to the next level in terms of parallelization is paramount for HEP scientists. The next generation of experiments will demand it and the current ones will profit greatly. The expertise in these two communities exists to make this happen and there is willingness by both groups to join forces and take this software to the next level.

# 6 Acknowledgment

The editors of this report would like to acknowledge the guidance, help, and support of many people. Dan Hitchcock and Jim Siegrist invited us to organize and run this workshop on transforming Geant4, a problem of great interest to both of us. Ceren Susut and Lali Chatterjee provided guidance throughout the process of organizing, running, and then reporting on the workshop. Donna Nevels from the Oak Ridge Institute for Science and Education managed the logistics of the workshop. Finally, there are many colleagues who did not attend the workshop, but nevertheless contributed to our understanding of Geant4 and the research needed to successfully transform it for the future. These include Ewa Deelman, Ann Chervenak, Jerome Lauret, and Arie Shoshani.

# 7 References

[1] http://geant4.cern.ch/.

[2] http://irs.inms.nrc.ca/software/egsnrc.

[3] http://www.fluka.org/fluka.php.

[4] http://mcnpx.lanl.gov/.

[5] http://www.oecd-nea.org/tools/abstract/detail/nea-1525.

[6] http://phits.jaea.go.jp/.

[7] http://www.fastrad.net/.

[8] http://projectx.fnal.gov/.

[9] http://www.muonsinc.com/muons3/G4beamline.

[10] http://cosray.unibe.ch/~laurent/planetocosmics/.

[11] ALICE. A Large Ion Collider Experiment. http://aliweb.cern.ch/.

[12] ATLAS. The ATLAS Experiment. http://atlas.web.cern.ch/Atlas/Collaboration/.

[13] S. Borkar and A. Chien. The Future of Microprocessors. Commun. ACM, 54(5):67–77, May 2011.

[14] Modeling Power and Energy Usage of HPC Kernels, Ananta Tiwari, Michael Laurenzano, LauraCarrington, and Allan Snavely, In The Eighth Workshop on High-Performance, Power-Aware Computing (HPPAC'12) , Shanghai China, May 2012

[15] CMS. Compact Muon Solenoid Experiment at CERN’s LHC. http://cms.cern.ch/.

[16] C. Tapus, I.H. Chung, and J. Hollingsworth. Active Harmony: towards automated performance tuning. In Proc. of the 2002 ACM/IEEE Conf. on Supercomputing, Supercomputing ’02, pages 1–11, Los Alamitos, CA, USA, 2002. IEEE Computer Society Press.

[17] R.H. Dennard, F.H. Gaensslen, V.L. Rideout, E. Bassous, and A.R. LeBlanc. Design of ion-implanted mosfets with very small physical dimensions, October 1974.

[18] Xin Dong, Gene Cooperman, John Apostolakis, Andrzej Nowak, Sverre Jarp, Makoto Asai, and Daniel Brandt. Creating and improving multi-threaded geant4. 2012. http://indico.cern.ch/getFile.py/access?contribId=509&sessionId=8&resId=0&materialId=slide&confId=149557.

[19] HPCToolKit, 2011. http://hpctoolkit.org/.

[20] LHC. The Large Hadron Collider. http://lhc.web.cern.ch/lhc/. [21] LHCb. The LHCb collaboration. http://lhcb.web.cern.ch/lhcb/.

[22] Gordon Moore. Cramming more components onto integrated circuits. http://download.intel.com/museum/ Moores_Law/ArticlesPress_Releases/Gordon_Moore_1965_Article.pdf.

[23] Particle Physics Project Prioritization Panel. US Particle Physics: Scientific Opportunities. a Strategic Plan for the Next Ten Years. Technical report, US Department of Energy (DoE), May 2008. http://www.science.doe. gov/hep/files/pdfs/P5_Report\%2006022008.pdf.

[24] Daniel J. Quinlan. ROSE: Compiler Support for Object-Oriented Frameworks. Parallel Processing Letters,10(2/3):215–226, 2000. http://www.rosecompiler.org.

# Appendix A: Uses of Geant4

The HEP community of Geant4 users is very large. Most projects recommended by the US Particle Physics Project Prioritization Panel (P5) use Geant4 as their detector simulation engine [23]. This is the case for the ATLAS [12], CMS [15], LHCb [21], and ALICE [11], experiments at the CERN Large Hadron Collider (LHC) [20] accelerator. Each of these experiments consists of international collaborations between thousands of physicists, computing scientists, and engineers who join forces to build and operate highly complex accelerators and detectors worth billions of dollars. Geant4 is also the preferred simulation tool for generic detector R&D, as well as for all world-wide simulation activities which are part of the feasibility studies for high-energy lepton colliders. The demands on the Geant4 tool kit are growing rapidly as the current experiments and future projects focus on quality physics and faster and more robust software, and also as new HEP projects adopt Geant4 as their main simulation engine.

The LHC experiments produced and published a large number of precision physics results only a few months into the 2010 run. This impressive milestone is due, in part, to the successful simulation efforts implemented by the experiments, based on Geant4. Each of the four LHC experiments generated, simulated, and reconstructed large amounts of Monte Carlo data totaling billions of collision events that were used to commission the detector calibration methods, particle reconstruction algorithms, and analysis techniques. These algorithms and methods performed almost out-of-the-box when validated on real collider data, as a consequence of the unprecedented physics and computing performance of the Geant4 based simulation software.

Geant4 is a clear example of HEP software technology with transformational applications to other areas of scientific research. The Geant4 community of developers and users extends beyond HEP and , physics into other areas such as accelerator science, astroparticle physics, space engineering, medical physics, education, and industrial applications.

In accelerator science, Geant4-based applications are used to design accelerators and their shielding structures. For example, the proposed Project-X [8] muon collider is modeled by an application named G4beamline [9].

In astroparticle physics, Geant4s unique ability to simulate the acceleration of charged particles in dynamic electromagnetic fields enables the simulation of solar flares. In addition, the magnetosphere of planets within our solar system are modeled in PlanetoCosmics [10] based on Geant4 to simulate the radiation environment of the planets. Recently a more radical use of Geant4 has been started to simulate black holes, pulsars and supernovas.

In space engineering, Geant4 has been used to evaluate the amount of radiation in a space craft, to calibrate onboard detectors when a craft is in operation, to estimate the radiation dose received by astronauts and electronic devices in the International Space Station and in future manned missions to the Mars, and to simulate the radiation effects to semiconductor devices in space radiation environments. Space applications were pioneered by the European Space Agency (ESA), which contributed heavily to the development of core components such as low-energy electro-magnetic interactions and radioactive decay. In the US, NASA-funded groups also use Geant4, particularly for its ability to model the precise geometry of electronic components and the “upsets” that radiation induces.

Throughout the world of Medical Physics, the use of Monte Carlo simulation is growing. Geant4 is well established as a unique tool for precise prediction of energy deposition in proton and carbon therapy and also brachytherapy, following the long-established success in electron and gamma therapy of software derived from SLAC’s EGS code.  In many particularly exciting areas, such as particle therapy, mixed-mode imaging, and on-board imaging for dose verification, Geant4 has been the simulation  tool of choice. Medical users value Geant4 because of its ability to simulate the dynamic motion of the geometries of patients and the treatment apparatus.

For developing educational software, Geant4 is an ideal tool not only because of its comprehensive physics coverage and flexible geometry description, but also because of its powerful visualization and interactivity.  Several ongoing projects are aimed to teach the nature of fundamental particles and properties of radiation to college-level and even younger students.

Finally, industry has also adopted Geant4 as a tool to design non-destructive systems for inspection and testing.

# Appendix B: Geant4 Collaboration and Users

High energy particle physicists have a long history of building successful collaborations to construct and exploit major detectors. Innovative devices costing hundreds of millions of dollars and thousands of person-years of effort are typically constructed on time and on budget using contributions from independently funded and autonomous collaborators. All of this is invariably achieved within a non-binding framework of Memoranda of Understanding.

A similar approach has been followed with Geant4. The Geant4 collaboration, founded in 1994 is comprised of approximately one hundred scientists. The primary difference between collaborations that construct the detectors for the field versus those that write the software tools is that constructing software is a much more dynamically changeable undertaking than constructing a modern detector and thus in many ways a more difficult challenge to maintain a unified focus. In hardware and software collaborations, no one member could be an expert in all of its components, and thus a broad international coordination is essential.

The formal structure of the Geant4 collaboration has evolved over its 18 years of existence. The current structure places the collaboration management firmly in the hands of the scientists contributing to Geant4, but with processes (the Oversight Board and the Technical Forum) that are designed to ensure that the goals and concerns of resource providers and of the user community are taken into account. The Geant4 collaboration takes responsibility for maintaining Geant4 as a reliable tool for current and near future HEP experiments and all other user domains for at least another decade. As the use of Geant4 expands, new use-cases drive code expansion in all of its components, for example in the kernel (introduction of the concept of a crystal), in physics models (anti-particle hadronic physics, very-low energy EM physics, phonon physics), and in visualization (3-D transparent rendering). Since major applications of Geant4 need very high statistics, systematic testing of Geant4 requires massive CPU resources and network tools. International collaboration greatly facilitates the provision of such testing environments.

The success of Geant4 also relies on extensive user support. A US “Transforming Geant4” initiative should be integrated within the Geant4 toolkit and coordinated with the international Geant4 collaboration so that the worldwide development effort remains coherent. The collaboration uses a variety of formal and less formal constructs to keep in close contact with the evolving user community and to encourage users to share experiences and discuss common needs:

- Technical Forum: t his is a regular open meeting chaired by a scientist from the user community and attended by key members of the collaboration. The Technical Forum is aimed mainly at discussing future development rather than immediate user problems. (http://geant4.cern.ch/geant4/collaboration/technical forum/)

- Experiment Liaisons: Major HEP experiments and some other applications typically have a Geant4 member who also works within the experiment part of the time. Generally this is not a “free gift” by the Geant4 collaboration – the person is assigned by their institute to work part time on Geant4 development and part time on experiment support.

- User’s Workshops and Tutorials:  User-focused  workshops are organized by collaboration members several times a year in various geographic locations with a varying focus. For example a 2010 Space User’s Workshop was hosted by Boeing in Seattle. (http://active.boeing.com/events/GEANT4/index.cfm)

- Autonomous User Organizations:  A good example is G4NAMU, the Geant4 North American Medical Users organization (http://geant4.slac.stanford.edu/g4namu/)

- Hyper news:  This allows informal user-to-user communication and developer-user help, plus the possibility to browse existing threads for answers to questions. (http://hypernews.slac.stanford.edu/HyperNews/geant4/cindex)

# Appendix C: Workshop Participants

| | | |
|---|---|---|
| David | Asner | Pacific Northwest National Laboratory |
| Amber | Boehnlein | SLAC National Accelerator Laboratory |
| Richard | Brower | Boston University |
| Paolo | Calafiura | Lawrence Berkeley National Laboratory |
| Philippe | Canal | Fermi National Accelerator Laboratory |
| Lali | Chatterjee | DOE SC High Energy Physics |
| Gene | Cooperman | Northeastern University |
| Terence | Critchlow | Pacific Northwest National Laboratory |
| Pedro | Diniz | University of Southern California |
| V. Daniel | Elvira | Fermi National Accelerator Laboratory |
| Michael | Ernst | Brookhaven National Laboratory |
| Robert | Fowler | University of North Carolina |
| Salman | Habib | Argonne National Laboratory |
| Andrew | Hanushevsky | SLAC National Accelerator Laboratory |
| Jim | Kowalkowski | Fermi National Accelerator Laboratory |
| David | Lange | Lawrence Livermore National Laboratory |
| Randall | Laviolette | DOE SC Advanced Scientific Computing Research |
| Thomas | LeCompte | Argonne National Laboratory |
| Steven | Lee | DOE SC Advanced Scientific Computing Research |
| Qing | Liu | Oak Ridge National Laboratory |
| Bob | Lucas | University of Southern California |
| David | Malon | Argonne National Laboratory |
| Gabriel | Marin | Oak Ridge National Laboratory |
| John | Mellor-Crummey | Rice University |
| Richard | Mount | SLAC National Accelerator Laboratory |
| Esmond | Ng | Lawrence Berkeley National Laboratory |
| Boyana | Norris | Argonne National Laboratory |
| Lucy | Nowell | DOE SC Advanced Scientific Computing Research |
| Bruce | Palmer | Pacific Northwest National Laboratory |
| Karen | Pao | DOE SC Advanced Scientific Computing Research |
| Marc | Paterno | Fermi National Accelerator Laboratory |
| Joseph | Perl | SLAC National Accelerator Laboratory |
| Allan | Porterfield | University of North Carolina |
| Lawrence | Price | DOE S C High Energy Physics |
| Michael | Procario | DOE SC High Energy Physics |
| Kenneth | Roche | Pacific Northwest National Laboratory |
| Rob | Roser | Fermi National Accelerator Laboratory |
| Paul | Ruth | University of North Carolina |
| Allen | Sanderson | University of Utah |
| Elizabeth | Sexton-Kennedy | Fermi National Accelerator Laboratory |
| Panagiotis | Spentzouris | Fermi National Accelerator Laboratory |
| Ceren | Susut | DOE SC Advanced Scientific Computing Research |
| Timothy | Tautges | Argonne National Lab |
| Craig | Tull | Lawrence Berkeley National Laboratory |
| Brian | Van Straalen | Lawrence Berkeley National Laboratory |
| Torre | Wenaus | Brookhaven National Laboratory |
| Dennis | Wright | SLAC National Accelerator Laboratory |
| John | Wu | Lawrence Berkeley National Laboratory |

# Appendix D: Transforming Geant4 for the Future Workshop Agenda

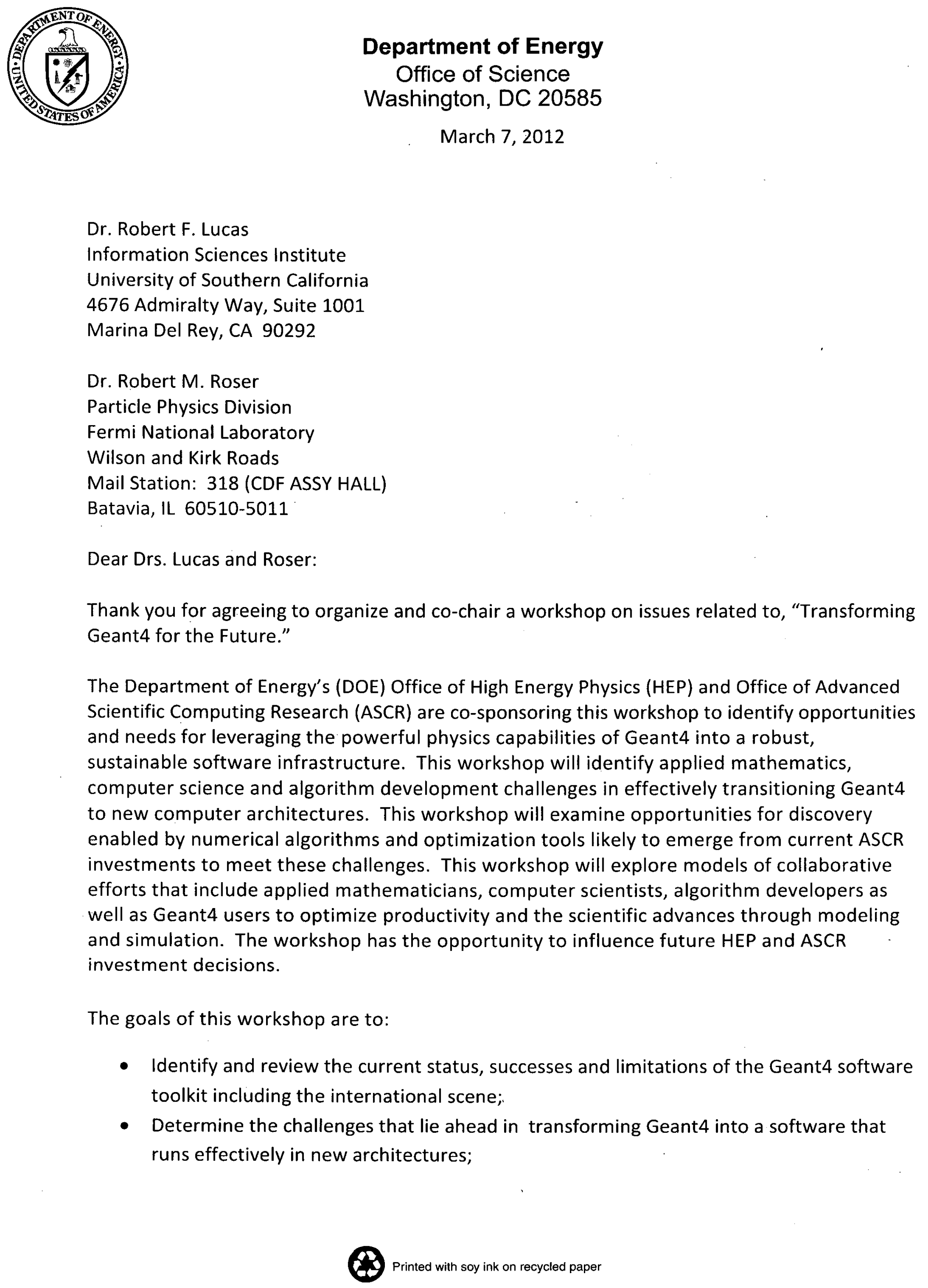
## May 8

| | | |
|---|---|---|
| 8:00-8:30 | Registration Open<br>Continental Breakfast | |
| 8:30-9:00 am | Welcome and Goals<br>ASCR and HEP | Dan Hitchcock, ASCR<br>Jim Siegrist, HEP |
| | Conference Chairs | Bob Lucas, USC<br>Rob Roser, FNAL |
| 9:00-9:30 am | Geant4 overview | Amber Boehnlein, SLAC |
| 9:30-9:45 am | Geant4 Collaboration and History | Asai Makoto, SLAC |
| 9:45-10:15 am | Physics uses of Geant4 | Tom LeCompte, ANL |
| 10:15-10:45 am | Break | |
| 10:45-11:15 am | Trends in multi-core architecture and optimization opportunities | Rob Fowler, UNC |
| 11:15-11:45 am | Exploiting concurrency in Geant4 | Jim Kowalkowski, FNAL |
| 1:45 – 12:15 pm | Scientific data management and analysis challenges | Rob Ross, ANL |
| 12:15-12:30 pm | Charge to Workshop Participants | Bob Lucas, USC<br>Rob Roser, FNAL |
| 12:30-1:45 pm | Lunch on your own | |
| 1:45-3:15 pm | Parallel Sessions<br>Multi-core Optimization<br>Scientific Data Handling and Analysis | |
| 3:15-3:30 pm | Break | |
| 3:30-5:00 pm | Resume sessions | |
| 5:00 – 5:30 pm | Report of Parallel Session progress | |
| 5:30 pm | Adjourn for the day | |

## May 9

| | |
|---|---|
| 8:00-8:30 am | Continental Breakfast |
| 8:30-10:30 am | Resume breakout discussions |
| 10:30-11:00 am | Break |
| 11:00-12:00 pm | Plenary reports from discussions |
| 12:00-12:15 pm | Closing remarks and Paths forward |
| 12:15-1:30 pm | Workshop adjourn<br>Working lunch for organizers and chairs |
| 1:30 pm | Report preparation |

# Appendix E: Charge Letter



**Department of Energy**
Office of Science
Washington, DC 20585

March 7, 2012

Dr. Robert F. Lucas
Information Sciences Institute
University of Southern California
4676 Admiralty Way, Suite 1001
Marina Del Rey, CA 90292

Dr. Robert M. Roser
Particle Physics Division
Fermi National Laboratory
Wilson and Kirk Roads
Mail Station: 318 (CDF ASSY HALL)
Batavia, IL 60510-5011

Dear Drs. Lucas and Roser:

Thank you for agreeing to organize and co-chair a workshop on issues related to, "Transforming Geant4 for the Future."

The Department of Energy's (DOE) Office of High Energy Physics (HEP) and Office of Advanced Scientific Computing Research (ASCR) are co-sponsoring this workshop to identify opportunities and needs for leveraging the powerful physics capabilities of Geant4 into a robust, sustainable software infrastructure. This workshop will identify applied mathematics, computer science and algorithm development challenges in effectively transitioning Geant4 to new computer architectures. This workshop will examine opportunities for discovery enabled by numerical algorithms and optimization tools likely to emerge from current ASCR investments to meet these challenges. This workshop will explore models of collaborative efforts that include applied mathematicians, computer scientists, algorithm developers as well as Geant4 users to optimize productivity and the scientific advances through modeling and simulation. The workshop has the opportunity to influence future HEP and ASCR investment decisions.

The goals of this workshop are to:

- Identify and review the current status, successes and limitations of the Geant4 software toolkit including the international scene;
- Determine the challenges that lie ahead in transforming Geant4 into a software that runs effectively in new architectures;

- Consider the opportunities related to existing algorithms, optimization tools and physics models;
- Ascertain research areas in applied mathematics, computer science, algorithm development and simulation strategies needed to leverage the powerful physics capabilities of Geant4 into a robust, sustainable software infrastructure;
- Create the foundation for information exchanges and collaborations among ASCR and HEP supported researchers, ASCR computing facilities and Geant4 user community;
- Understand the research that may currently be in progress at the international level and identify directions that would not duplicate existing projects;
- Explore possibilities for transformative advances that could ensue through the unique characteristics of the HEP-ASCR collaborations.

The workshop should focus on areas of research and collaboration to position Geant4 to exploit emerging computer architectures, while providing for a strong, diverse and potentially changing user community. We anticipate that the workshop will develop findings in the context of HEP and ASCR missions and the collaborative exchanges between the two communities will seed fruitful directions that enhance the impact of the workshop. We anticipate that you will establish a program committee to organize the workshop and that the workshop will consist of plenary and breakout sessions.

The workshop should be held in the Washington, DC, metropolitan area in early to mid-May 2012 time frame. We request that a written report representing the results of the workshop be prepared by you as workshop chairs, with inputs from panel leads, and other assigned writers. The report should specifically address all workshop goals. We would like a draft version of the Executive Summary, containing an overview of the major findings of the workshop, within 45 days after the workshop. The final report will be used by HEP and ASCR to shape out-year program plans and to inform the Office of Science long-range budget planning process.

Dr. Ceren Susut (Ceren.Susut-Bennett@science.doe.gov), ASCR, and Dr. Lali Chatterjee (Lali.Chatterjee@science.doe.gov), HEP, will be your primary DOE contacts for this workshop and will provide any support needed to organize and conduct a successful workshop.

This workshop is an important step toward developing and executing the strategic vision for porting Geant4 in the future through a partnership between HEP and ASCR. Thank you again for agreeing to contribute to this effort.

Sincerely,

Dr. Daniel A. Hitchcock
Associate Director of Science for the
Office of Advanced Scientific Computing

James Siegrist

Dr. James Siegrist
Associate Director of Science for the
Office of High Energy Physics