Trends in Computing

Jürgen Knobloch and Maria Grazia Pia

Abstract—The challenges of computing in science and technology have grown in several aspects: computing software and infrastructure is mission critical, the requirements in communication, storage, and CPU capacity are often above the currently feasible and affordable level, and the large science projects have a lifetime that is much longer than one cycle in hardware and software technology. Cross-disciplinary solutions in physics and detector simulation, data analysis, as well as in software engineering and Grid technology, are presented, with examples of their use in global science projects in the fields of physics, medicine, and space.

I. CHALLENGES OF PARTICLE PHYSICS COMPUTING

A. Physics Challenges

COMPUTING in particle physics and related research domains faces unprecedented challenges, determined by the complexity of physics under study, the detectors, and the environments where they operate.

The complexity of physics of the Large Hadron Collider (LHC) experiments at CERN [1] derives from the need to extract rare events from an overwhelming QCD background, whose cross section exceeds the interesting signals by nine orders of magnitude. The high machine luminosity necessary to produce such rare events contributes to the complexity of the patterns to be studied, where the interesting physics features must be recognized, reconstructed, and analyzed from detector signals resulting from the overlap of several events occurring at every bunch crossing. The large number of 10^9 events recorded annually per experiment together with the complexity of the physics and of detection techniques is the source of the enormous computational requirements of LHC experiments: 12–14 petabytes of data will be generated each year, and their analysis will require the equivalent of 70 000 of today's fastest PC computers. About 5000 scientists in institutes spread around the globe require access to the data and computing resources. Moreover, the LHC software project lifetime of 20 years requires that the software system be prepared for several technology changes.

The software for astro-particle experiments, which embrace a wide spectrum of scientific programs, from underground neutrino physics and dark matter searches to astrophysics measurements on satellites to cosmic ray physics, is characterized by an ample variety of requirements. In such experiments, the physics interactions involved may span many orders of magnitude in energy range, for instance, from the scale of X-ray fluorescence

emission to ultrahigh energy cosmic rays in the PeV range. Modeling physics interactions over such an extended energy range, as well as complex environments, such as underground experimental areas, or sophisticated spacecrafts enclosing science instruments, represents a computing challenge, especially for simulation software. This challenge is even strengthened by the fact that software often plays a mission-critical role in many of such experiments: The characteristics of detectors and radiation shielding of an experiment hosted on a satellite, usually based on software simulations, cannot be modified after the launch. Therefore, the need to deal with the risks intrinsic to mission-critical applications generates strict requirements of software quality and reliability, in addition to the functional requirements specific to the physics scope of such experiments.

Similar needs of quality and reliability are also fundamental in medical applications, such as oncological radiotherapy and medical imaging. Other requirements derive from the peculiar environment of operation of particle physics software in this domain: friendly user interface and easy configuration of the experimental setup are essential characteristics for usage by hospital staff, without requiring specific software knowledge. The major challenge in this field, especially concerning the usage of particle physics software in radiotherapy practice, consists of achieving adequate speed for clinical application, without sacrificing precision and reliability.

B. Computing Challenges

The complexity of the computing environment itself adds to the challenges associated to the complexity of physics environments and detectors.

If one looks retrospectively at the last 25 years of the past century—i.e., a time scale equivalent to the life cycle of large scale experiments—one notices several major evolutions in computing hardware, software, and operating systems, as illustrated in Fig. 1. Such events go from the production of the first IBM Personal Computer (1981) to the marketing of Intel Pentium processors (1992), from the appearance of new operating systems such as Unix (1975) and Linux (1991) to new programming languages such as C (1978), C++ (1985) and Java (1995); from the development of the World Wide Web (1990) to the launch of grid computing (1998). Therefore, physics experiments must cope not only with their own intrinsic computing needs, but also with a fast changing computing environment.

Moreover, it should be noted that not only the computing environment changes over the life cycle of a typical physics experiment, but also the requirements of the experiment itself: either because of modifications of the experimental setup, or because of new physics perspectives opened by evolutions in the theory or by experimental findings. Such evolutions over the life cycle of an experiment usually go toward a greater diversity:

Manuscript received November 22, 2003; revised May 7, 2004.

J. Knobloch is with CERN, CH-1211 Geneva, Switzerland (e-mail: Juergen.Knobloch@cern.ch).

M. G. Pia is with INFN Sezione di Genova, 16146 Genova, Italy (e-mail: Maria.Grazia.Pia@cern.ch).

Digital Object Identifier 10.1109/TNS.2004.835566

Fig. 1. Evolutions in computing hardware, software, and operating systems in the last 25 years of the 20th century, compared to milestones in high energy physics research.

This means that the software developed for a physics experiment should anticipate changes, and should be able to cope with new evolutions.

II. SOFTWARE TECHNOLOGY

Software technology is a key element, enabling science projects to cope with the challenges described in the previous section.

The availability of a new software technology—object-oriented techniques [2], first adopted for particle physics in the 1990s—has significantly contributed to address many of these challenging issues. Object-oriented technology has offered the means to achieve apparently contradictory objectives, fundamental for particle physics experiments and related applications: its characterizing features (inheritance, polymorphism, and dynamic binding) grant the experiments' software openness to change together with stability of core code, robustness for intensive application, as well as flexibility for alternative implementations. The object-oriented technology, and in particular the component-based architectural approach it supports, play a fundamental role in large size experiments, facilitating the maintainability of software over a long time scale and the integration of developments widely distributed geographically.

The relatively recent introduction of this new technology in particle physics experiments and related domains has represented a revolution with respect to well-established techniques in these fields; in fact, the move to the object-oriented technology has implied a change of paradigm, rather than just the adoption of a new programming technique. While the benefits of the technology are significant, the difficulty of its

penetration in an environment educated to, and familiar with, procedural programming are the source of many problems in the experiments. Integration of past expertise into the framework offered by the new technology represents new challenges in the experiments. The effective usage of the technology itself mastered only by a relatively small number of specialists needs to be established. At the same time, while the object-oriented technology has not been fully digested yet in the field of particle physics software, new emerging technologies (generic programming, generative programming, aspect-oriented programming, etc.) have appeared and have started to be used in pioneering projects in the same field.

One of the benefits of the adoption of the object-oriented technology consists in the possibility of defining protocols—in terms of abstract interfaces, thus facilitating the interoperability of different systems. This feature has been exploited by the AIDA [3] project—a common effort among a variety of individual projects for the development of data analysis systems—to define common interfaces. This strategy allows the user to select specific implementations at run time, by loading shared libraries, while the code only uses the common abstract interfaces. The technology allows even to go across the language barrier; in fact, AIDA interfaces allow the interoperability of C++ systems, like Anaphe [4] or Open Scientists [5], and of a Java system like JAS 0. This strategy offers various advantages: The user has the freedom to configure the software application with the analysis tools best fitting his specific needs, at the same time minimizing the risk of introducing dependencies on external software systems that could complicate the long term maintenance of the application. The power of this approach has been widely recognized, to the point that even previously existing software systems like ROOT [7] are adapted to comply to the standard set of interfaces, to allow for an easy selection of

Fig. 2. Object-oriented technology adopted by the Geant4 simulation toolkit allows us to provide a variety of physics models for hadronic interactions. Here is an application of various theory-based and parameterized hadronic models to the simulation of the ATLAS hadronic calorimeter, compared to experimental data and legacy FORTRAN simulation codes.

implementations in the applications, and for the interoperability of different subsystems.

The technology adopted is also instrumental to enable new functional capabilities in the software which, in turn, may open new perspectives to experimental research. Object-oriented technology has been the key to the strategy of the Geant4 [8] Toolkit to offer a variety of models for the simulation of the same physics processes: different approaches, specialized by energy range, particle type, and material, coexist in the toolkit, allowing the user to select the option most appropriate to any specific application, while the Geant4 kernel treats all of them transparently through the same abstract interfaces. Fig. 2 shows an example of application of various Geant4 physics models to the simulation of a hadronic calorimeter, in comparison to legacy Monte Carlo codes and to experimental data. Entire research fields, such as medical physics or space science, have profited of the new capabilities—such as, for instance, extensions of electromagnetic interactions down to low energies (<1 keV—enabled by the technology to coexist in the same toolkit together with more traditional approaches. In a similar way, generic programming techniques adopted in a novel toolkit for statistical data analysis [9] allow the user to configure his data analysis with the algorithm most appropriate to the comparison of any type of distributions, going far beyond the traditional χ^2 or Kolmogorov–Smirnov tests, previously available in the widely used CERN Library [10].

III. SOFTWARE ENGINEERING

While technology is a key factor, it is hard to expect that technology itself may represent a breakthrough with respect to the many challenges in particle physics software. Past observations

[11] have confirmed that, because of the intrinsic nature of software, it is hard to expect a technology revolution in the software domain, with effects comparable to significant innovations in computing hardware, like advancements in electronics and large-scale integration.

Indeed, in the recent years, the attention has shifted from mere software technology to the software process, that is, to the way software is developed; quantitative standards, such as the Capability Maturity Model and ISO/IEC 15504 [13], have been defined to evaluate the software process itself. The discipline of software engineering is as essential as software technology to cope with the challenges of computing for particle physics experiments and related fields; in particular, it is fundamental to address the many nonfunctional requirements associated with the challenges listed in Section I-B.

Geant4 has been the first one among large particle physics software projects to adopt a rigorous software engineering approach [14]. This move has been fundamental to the success of the project, allowing to fully exploit the advantages of the software technology in response to the needs of multidisciplinary applications in the most diverse fields. The adoption of a rigorous software engineering has also contributed to the transparency of implementations of various physics processes in the toolkit, thus contributing to the validation of experimental results.

While the role of software engineering, in general, and of software process, in particular, is well established in the professional software environment as the key to achieve software quality, and has been adopted for several years in physics environments characterized by mission critical software applications, such as space science, this discipline is relatively new to the domain of particle physics software, and is often perceived with some skepticism in particle physics research.

Fig. 3. Regional computing centers in a grid computing model are classified as Tier-0, Tier-1, Tier-2, and Tier3, according to their computing power and mass-storage capabilities.

IV. COMPUTING GRIDS

The emerging grid paradigm [15] represents the most promising solution to provide seamless, global computing resources for particle physics, biomedical research, astronomy, and other sciences and fields where similar requirements exist.

Grid computing is different from conventional distributed computing by providing transparent resource sharing for virtual organizations (VOs). At the heart of “The Grid” are several software components called middleware that hide the complexity of the underlying system of CPU servers, data servers, and performing networks from the end users providing seamless access to these resources. The major components of the system provide replication and location of data sets, allocation of resources, and authentication of users. Other components address the monitoring and management of the computing fabrics.

The LHC Computing Grid (LCG) [17] project sets out to provide the computing infrastructure for the LHC experiments that will start to take data in 2007. While the grid technology has been developed outside of high-energy physics, the presently ongoing debugging, certification, deployment, and operation in a global production quality environment constitutes a major contribution to the expected success of grids in many other fields.

The LCG project has chosen that the compute elements are built from commodity components such as standard PCs running the Linux operating system.

Although the regional computer centers (Fig. 3) are classified as Tier-0, Tier-1, Tier-2, etc., with decreasing requirements on compute power and mass-storage capability, the grid is designed in a decentralized way where the roles of operation centers and support centers can be geographically spread over several time zones.

A first version of the middleware software (LCG-1) deployed in the LCG project has been installed at 23 sites worldwide. The first data challenges of the LHC experiments consist in rather controlled simulation and reconstruction productions with well

defined programs run by a small number of experts. The grid will later be put to the real test when data analysis by many physicists will be tried on distributed data using a large diversity of software in a rather random fashion.

The LHC Computing Grid will provide the springboard for EGEE (Enabling Grids for E-science in Europe), a European consortium to build and provide round-the-clock grid service to scientists throughout Europe. However, the mission of EGEE is also to extend the potential benefits of a grid infrastructure beyond high energy physics. Projects for grid computing are currently in progress in diverse domains, from earth observation science to biology.

Grids being a new technology, several projects have set out to develop middleware leading to the problem of inter-operability between different incarnations of the Grid. It is hoped that the efforts for intergrid coordination lead to more standardization.

V. GLOBAL PERSPECTIVES OF PARTICLE PHYSICS SOFTWARE

A. International Software Collaboration

The complexity of the software to be developed in response to the physics challenges, the wide geographical spread of expertise required, the long-time scale of development, and maintenance over the life cycle of the experiments require significant investments of resources.

While in the past the development of general-purpose software systems was usually confined to small, specialized groups in a laboratory, a recent trend in particle physics consists in the creation of large international collaborations around software and computing projects of general interest. Examples of this trend are the Geant4 Collaboration, consisting of more than one hundred members, and the projects funded by the European Union for the development of computing grids.

The global nature of particle physics software is highlighted not only by the creation of large international collaborations dedicated to software projects, but also by the multidisciplinary spectrum of applications of the major simulation and analysis systems. Similar requirements of functional capabilities—such as physics models or mathematical algorithms—are shared by largely different experimental domains: The ability to offer solutions capable to go across the boundaries of a single experiment, or even of a field of scientific research, represents one of the major challenges of today’s software development. However, in a few cases, only the common needs have been formally recognized and lead to a cooperation of different domains for the development and maintenance of common software: the Geant4 Collaboration, gathering particle physics laboratories and institutes, space agencies, and industry and medical organizations, are some of such examples.

B. Technology Transfer

The flexibility and the rich functionality of modern particle physics software has resulted in multidisciplinary applications of software initially developed for high energy physics experiments. This move has been facilitated by the adoption of the object-oriented technology: The creation of protocols through abstract interfaces has allowed to easily integrate new software systems into diverse application frameworks, at the same time

Fig. 4. Dose distribution of a brachytherapeutic source in a water phantom, resulting from a simulation application of Geant4 and related data analysis with the AIDA-compliant Anaphe system.

limiting the risk of external dependencies. The wide usage of the Geant4 Toolkit, whose development was originally motivated by the simulation requirements of LHC experiments, is a show-case example of the transfer of technologies from the domain of high energy physics to other domains, like space science and medicine [17].

The open, collaborative nature of modern scientific software has configured a new way of operation with respect to the traditional technology transfer [18]: The feedback from usage in diverse domains, the test and validation results provided by experimental groups worldwide, the new ideas and instruments for further extensions of the software represent a significant contribution to the mainstream development and to the overall quality of the software product.

C. Impact of Particle Physics Software on Society

The advanced functionality offered by the software developed for challenging particle physics projects, together with its open source nature, can have a significant impact on society at large. The World Wide Web, originally developed at CERN, is the best-known example of the social impact of computing research for particle physics.

New systems entirely based on technologies originating from the field of high energy physics, currently under development or being discussed, are characterized by a potential social impact of their applications. A prototype of a novel dosimetric system [20] has been developed by integrating into a common framework simulation (Geant4) and analysis (AIDA-compliant packages) systems, to offer a precise tool for radiotherapeutic dosimetry. Fig. 4 shows the dose distribution originating from a radioactive source for brachytherapy, generated by this dosimetric system. The web provides the means for user friendly interface and easy configuration, suitable to the usage in hospital environments. An

intermediate layer, such as the DIANE [21] tool, allows the software to run transparently in sequential mode on a single machine, or in parallel mode either on a local farm or accessing distributed computing resources on a grid. In this way, advanced software for accurate dosimetric studies can be made available at no additional cost to any hospitals, even small ones or those located in less wealthy countries, that cannot afford the high expenses for commercial treatment planning software systems.

The grid technology may also provide opportunities for applications with potential social impact. A proposal [22] has been presented recently for the creation of a grid for the preservation and distributed access of archeological information, thus shielding the patrimony of the cultural roots of mankind from its possible destruction, due either to natural catastrophes (earthquakes, floods, etc.) or to man-provoked ones, like wars or indiscriminate constructions.

VI. CONCLUSION

A trend is observed in the experimental physics computing, where different research domains—high energy physics, astrophysics and space science, and medical physics—share similar needs and adopt common technological solutions. Software systems developed in the recent years for simulation and data analysis are characterized by a wide spectrum of applications, made possible by the versatility and openness enabled by object-oriented methods. The emerging grid paradigm offers global access to computing resources in response to the similar challenging needs of diverse fields. Software and computing solutions originally developed for advanced fundamental physics research are applied in other domains, like medicine, with potential social impact for their sophisticated technical capabilities at low cost.

ACKNOWLEDGMENT

The authors would like to thank the Geant4 Collaboration and the AIDA Collaboration for many fruitful discussions on the themes outlined in this article.

REFERENCES

- [1] P. Sphicas, "Forward look at LHC physics," *Nucl. Phys. B.*, vol. 117, pp. 298–317, Apr. 2003.
- [2] G. Booch, *Object-Oriented Analysis and Design With Applications*. Menlo Park, CA: Benjamin/Cummings, 1994.
- [3] G. Barrand, P. Binko, M. Donszelmann, A. Johnson, and A. Pfeiffer, "Abstract interfaces for data analysis: Component architecture for data analysis tools," in *Proc. CHEP Conf.*, Beijing, China, Sept. 2001, CERN-IT-2001-013.
- [4] O. Couet, B. Ferrero-Merlino, Z. Molnar, J. T. Moscicki, A. Pfeiffer, and M. Sang, "ANAPHE-OO libraries and tools for data analysis," in *Proc. CHEP Conf.*, Beijing, China, Sept. 2001, CERN-IT-2001-012.
- [5] [Online]. Available: <http://www.lal.in2p3.fr/OpenScientist/>
- [6] M. Donszelmann, A. Johnson, J. Perl, V. Serbo, and M. Turri, "JAS3—A general purpose data analysis framework for high energy physics and beyond," in *Proc. IEEE Nuclear Science Symp.*, Portland, OR, Oct. 2003.
- [7] R. Brun and F. Rademakers, "ROOT—An object oriented data analysis framework," *Nucl. Instrum. Methods A*, vol. 389, pp. 81–86, 1997.
- [8] S. Agostinelli *et al.*, "Geant4: A simulation toolkit," *Nucl. Instrum. Methods A*, vol. 506, pp. 250–303, July 2003.
- [9] S. Donadio, S. Guatelli, B. Mascialino, A. Pfeiffer, M. G. Pia, A. Ribon, and P. Viarengo, "Data analysis in HEP: A statistical toolkit," in *Proc. IEEE Nuclear Science Symp.*, Portland, OR, Oct. 2003.
- [10] [Online]. Available: <http://cern.ch/cernlib/>

- [11] F. P. Brooks, "No silver bullet—Essence and accidents of software engineering," *IEEE Computer*, vol. 20, pp. 10–19, Apr. 1987.
- [12] M. C. Paulk and C. V. Weber, *The Capability Maturity Model: Guidelines for Improving the Software Process*. Reading, MA: Addison-Wesley, 1995.
- [13] "ISO/IEC DTR 15 504 software process assessment," ISO/IEC Joint Technical Committee 1 (JTC1).
- [14] G. Cosmo, "Geant4 software process," in *Proc. CHEP Conf.*, Beijing, China, Sept. 2001.
- [15] I. Foster and C. Kesselman, *The Grid*. San Mateo, CA: Morgan Kaufmann, 2003.
- [16] [Online]. Available: <http://cern.ch/lcg>
- [17] M. G. Pia and J. Knobloch, *Particle Physics Software Aids Space and Medicine*: CERN Courier, 2002.
- [18] S. Chauvie, G. Cosmo, F. Foppiano, E. Lamanna, F. Marchetto, P. Nieminen, M. G. Pia, A. Solano, and M. Tropeano, "From HEP computing to biomedical research and vice-versa: Technology transfer and application results," in *Proc. CHEP Conf.*, Beijing, China, Sept. 2001.
- [19] [Online]. Available: <http://www.w3.org/>
- [20] F. Foppiano, S. Guatelli, J. Moscicki, and M. G. Pia, "From DICOM to GRID: A dosimetric system for brachytherapy born from HEP," in *Proc. IEEE Nuclear Science Symp.*, Portland, OR, Oct. 2003.
- [21] J. T. Moscicki, "Distributed analysis environment for HEP and interdisciplinary applications," *Nucl. Instrum. Methods*, vol. 502, pp. 426–429, 2003.
- [22] G. Pelfer and P. G. Pelfer, "From WEB to GRID: A new perspective for archaeology," in *Proc. IEEE Nuclear Science Symp.*, Portland, OR, Oct. 2003.