Institute of Physics Publishing Journal of Physics: Conference Series **46** (2006) 561–569
doi:10.1088/1742-6596/46/1/078 SciDAC 2006

VACET: Proposed SciDAC2 Visualization and Analytics Center for Enabling Technologies

W. Bethel
Lawrence Berkeley National Laboratory
E-mail: ewbethel@lbl.gov

C. Johnson, C. Hansen, S. Parker, A. Sanderson, C. Silva, X. Tricoche
Scientific Computing and Imaging Institute, University of Utah
E-mail: [crj,hansen,sparker,allen,csilva,tricoche]@sci.utah.edu

V. Pascucci, H. Childs, J. Cohen, M. Duchaineau, D. Laney, P. Lindstrom
Lawrence Livermore National Laboratory
E-mail: [pascucci1,childs3,cohen22,duchaineau1,laney1,lindstrom2]@llnl.gov

S. Ahern, J. Meredith, G. Ostrouchov
Oak Ridge National Laboratory, [ahern, jsmeredith, ostrouchovg]@ornl.gov
E-mail: [ahern,jsmeredith,ostrouchovg]@ornl.gov

K. Joy, B. Hamann
University of California, Davis
E-mail: [joy,hamann]@cs.ucdavis.edu

Abstract.
This project focuses on leveraging scientific visualization and analytics software technology as an enabling technology for increasing scientific productivity and insight. Advances in computational technology have resulted in an "information big bang," which in turn has created a significant data understanding challenge. This challenge is widely acknowledged to be one of the primary bottlenecks in contemporary science. The vision for our Center is to respond directly to that challenge by adapting, extending, creating when necessary and deploying visualization and data understanding technologies for our science stakeholders. Using an organizational model as a Visualization and Analytics Center for Enabling Technologies (VACET), we are well positioned to be responsive to the needs of a diverse set of scientific stakeholders in a coordinated fashion using a range of visualization, mathematics, statistics, computer and computational science and data management technologies.

1. Scientific Data Understanding Requirements

To engineer our activities to be directly responsive to the needs of the scientific community, we draw upon a number of information sources. In 2002, computational scientists at NERSC provided detailed input describing their visualization requirements for the next three to five years [9]. Their needs were reiterated later in in the SCaLeS workshop [4] and the 2004 "Data-Management Challenge" workshop [13]. Application scientists have expressed a need for more advanced visualization and analysis tools, realizing that the "capabilities of earlier tools are not adequate to effectively present the meaningful information inherent in large, multidimensional data." Scientists have cited the urgent need for data understanding solutions

that span visualization, analysis, and data management. The "urgent" nature for these needs is predicated upon the explosive and seemingly endless growth in data size and complexity, which creates significant impediments to scientific understanding. In 2005, the VACET team conducted an on-line web-based survey of approximately 50 computational scientists who provided detailed information about their current and future data understanding needs in both qualitative and quantitative terms. Late in 2005 and early in 2006, the VACET team conducted in-depth, one-on-one interviews with application stakeholders to derive a concise set of visualization needs. These stakeholders consist of application scientists who are already funded under SciDAC, submitting proposals to be funded under SciDAC2, or who are awardees under DOE's INCITE program.

The needs of our stakeholder community are diverse, and include multivariate data visualization and analysis, exploration through linked views, data/feature mining, remote visualization, scalable visualization, domain-specific interfaces to reduce complexity and increase productivity, coupling visualization, data management and analysis into a seamless solution, and turn-key solutions that do not require a visualization expert. A recurring theme is the urgent need for solutions that enable knowledge discovery in large and complex data across all areas of science. Our Center's activities and deliverables can all be traced directly to specific application needs identified in the user input we have collected over the years, with direct emphasis on the needs of our application stakeholders.

2. Visualization and Analytics Technologies

Our main goal is to develop and deploy a variety of data analysis and visualization tools for our science stakeholders. They have diverse data understanding needs, use a variety of computing resources, and are geographically distributed. Additionally, we want to leverage solutions developed and deployed for one stakeholder to many other projects. We address these challenges by using a flexible approach to software development and project management that draws from the diverse strengths of our team. Based upon specific input from science stakeholders – which include the fields of climate modeling, fusion, combustion chemistry, astrophysics and environmental management – we group their needs into two main categories: (1) visualization techniques, ranging from classical rendering techniques to the most advanced data streaming and remote data access algorithms for managing extremely large datasets, and (2) analytics techniques, including data exploration, feature extraction, tracking and comparison that aid the scientist in the actual information discovery process. The stakeholders, specific needs and relative priority are shown in Figure 1.

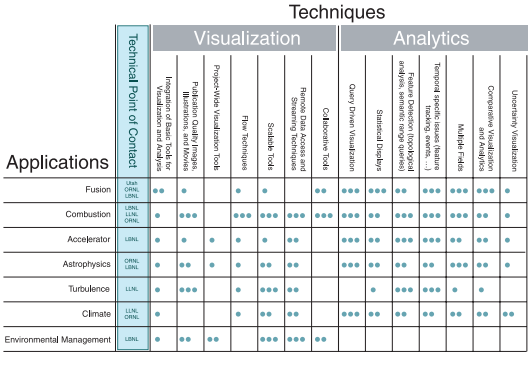

Figure 1. Relationship between application stakeholders and visualization and analytics techniques.

We also recognize the different needs of the users in terms of delivery vehicles of new technology. Some have already committed to a particular visualization tool and need new capabilities like high performance volume rendering or feature extraction. Others require new focused tools that address specific needs such as real time monitoring of large scale simulations or remote data access and collaborative data exploration. Many users can be well served with general purpose visualization tools, such as VisIt or SCIRun, and we'll provide extensions to support new features and special data formats. We will address this diversity of needs with a three-tiered strategy with development of: *libraries* with narrowly scoped capabilities (e.g. a volume

rendering library) and simple APIs that allow easy integration even if at some cost in space and/or time efficiency; *components* that can be easily combined in a data flow network exploiting the efficiency of the parts at their best; and *fully featured application visualization tools* with a large set of features that can satisfy a wide range of users.

This deployment strategy, together with extensive documentation, examples, and tutorials, will also facilitate the process of dissemination to a larger community independent of our ability to provide direct support to each of them.

2.1. Advanced Visualization Techniques

All of our stakeholders point out the important role that "basic" visualization and analysis capabilities play in the day-to-day process of scientific inquiry and discovery. These include charting, graphing, plotting and filtering. In the category of "basic techniques," we also include: visualization "staple algorithms" for scalar fields (isocontouring, hyperslicing, direct volume rendering), vector fields (direct and indirect representation techniques, e.g., glyph-based and streamlines), support tools like transfer function editors, dimension reduction and projection techniques and methods for displaying computational grids. Our stakeholders stress that any new visualization and analytics techniques must seamlessly integrate with their existing working environment. Otherwise, new technologies simply won't be used: our stakeholders clearly indicate they are unwilling to learn and use a separate tool that provides a single new feature. Our Center departs from the typical visualization research project approach in a key regard: through careful attention to software design and engineering, we aim to provide "staple" and new technologies that are well integrated and supported. This approach minimizes disruption to our stakeholders and protects DOE's investment in visualization research and development.

A "project-wide" visualization tool is an application (or framework or technology collection) that is used by all members of a particular project or community. Such a "standard" set of tools helps increase scientific efficiency by reducing the complexity of maintaining many different tools and helping a community establish and maintain "standard ways" of visual data analysis. A "project-wide" visualization tool doesn't necessarily mean a "specific visualization application" as much as it means a consistent and easy-to-use interface to commonly needed and domain-standard capabilities. This type of interface is sometimes referred to as a "dashboard" – where the controls and displays are tailored for a particular scientific endeavor. Several of our current science stakeholders have requested this type of capability; these requests echo the sentiments of previous reports [9].

Our Center will focus effort on a set of related advanced visualization technologies – flow field visualization, scalable solutions, remote data access and streaming techniques, and collaborative tools – to be delivered within a set of technology delivery platforms (see Section 3). Virtually all of our stakeholders' simulations model the transport of matter or energy. The resulting vector fields appear in several forms like velocity, momentum and flux. Existing vector field methods have proven to be useful in limited situations (e.g., coarse grids, 2D domains) but are not so practical for large-scale, time-varying or higher-dimensional data. Our work in scalable algorithms and software implementation is driven by our stakeholders' desire to push data generation, collection and management into the petascale regime. Our existing technology delivery infrastructure has proven useful on some of DOE's largest to-date visual data understanding challenges. Complementary technologies include those for remote data access based upon multiresolution and streaming methods. These technologies help balance visual data understanding with I/O and data management challenges posed by petascale data. Several stakeholders have requested the ability to perform collaborative visual data analysis. Our approach will rely on a combination of proven techniques [18] augmented with nascent technologies emerging from DOE's SBIR/STTR program.

2.2. Analytics and Knowledge Discovery

We distinguish the term "analytics" from "visualization" as a technique or methodology that is more targeted for discovery and that relies on an iterative, investigative approach to data exploration. Our approach blends work in several related areas to produce a set of balanced, well-rounded solutions driven by the needs of our application stakeholders.

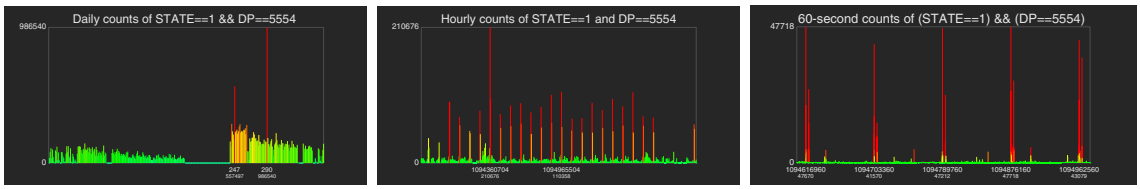

Figure 2. Query-driven visualization techniques are used to interactively explore 42 weeks' worth of network connection data to detect and characterize a distributed scanning attack. Combining a custom visual exploration application for specifying and displaying $n-$dimensional histograms with the FastBit [20] index/query technology from LBNL's Scientific Data Management group, we interactively mine and explore 42-weeks' worth of network connection data to characterize a distributed scanning attack. The histogram in the left image shows per-day counts of suspicious activity. Bars are colored by statistical moments: red bars lie three or more standard deviations from the mean, green bars are near the mean. The date range around day 247 is "interesting" because it shows an elevated and consistent level of suspicious activity. We drill into the data to show levels of suspicious activity in a 4-week window at one-hour temporal resolution (center image). Drilling further into the data, the right image shows a histogram of suspicious activity over a five-day period at one-minute temporal resolution. These images show the suspicious event occurs daily at 21:15 local time. The source of the attack was ultimately traced to a set of approximately twenty different hosts participating in the coordinated, distributed scanning activity. *Images courtesy of LBNL.*

Query-driven visualization refers to the process of limiting visualization and analytics processing to data a user deems "interesting." It forms the basis for many of our Center's work targets, including: feature detection, feature mining and correlated, linked views. Previous work in this area [21] highlights the need for close interactions with the field of Scientific Data Management where technologies for storing, indexing and finding data are blended with visual interfaces and data display techniques.

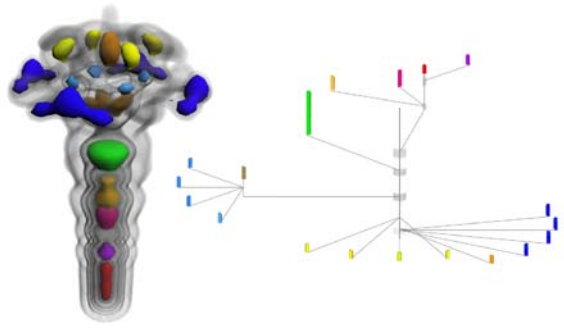

Figure 3. Topological features can guide the data exploration and comparative analysis process. On the left is a depiction of a "contour tree," a type of Reeb graph that serves as an abstract representation of peaks in data. Its branching structure reveals the manner in which maxima are nested. This idea has been extended to identification, tracking and analysis of features in time-varying data like vortices in combustion/turbulence simulations. *Image courtesy of LLNL.*

Other approaches to feature definition, detection, tracking and analysis are based upon a formal topological approach. We support the definition and systematic detection of complex features based upon a formal topological approach and an algorithmic framework that leverages the theory to permit an effective and accurate data analysis (see Figure 3). Our theoretical toolbox combines the classical critical point theory commonly used in fluid dynamics, and combinatorial algebraic topology, which offers guaranteed numerical stability and is robust to non-smooth data. The analysis produces diagrams, measurements, and visualizations that aid understanding intricate structures, provide qualitative domain segmentation, and rank topological features by importance yielding a multi-scale framework within which one can selectively analyze local and global trends in the data. Our direct interaction with the users allows them to formulate their feature characterization hypotheses in terms of this framework and map

the corresponding formal, unambiguous definitions to automatic and reliable extraction algorithms. This approach can replace traditional informal characterizations, which are hard to reproduce and less amenable for a systematic and verifiable analysis within a truly scientific method.

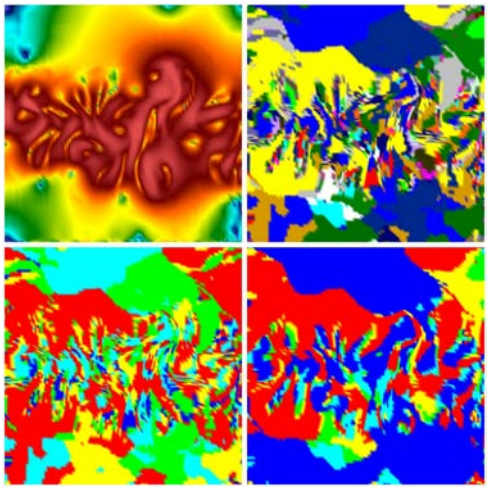

Figure 4. Another use for comparative visualization and analysis is to study the effects of parameter settings on the resulting output. These examples show compare the effects of varying two parameters – coefficients of turbulent viscosity and buoyancy – on the velocity magnitude computed by a Rayleigh-Taylor instability simulation. For each coefficient, five parameter values were selected, then twenty-five runs were executed corresponding to each permutation of parameter value pairs. We perform comparative analysis to visually analyze the effect of these parameters on velocity magnitude. The upper left image shows velocity magnitude from one run. In the upper right, grid points are colored by the simulation index having the maximum velocity at that point. That image shows that no one simulation dominates. In the lower left, grid points are colored by the buoyancy coefficient of the simulation having the maximum velocity. In the lower right, each grid point is colored by the turbulent velocity coefficient of the simulation having the maximum velocity. This final image shows that most of the high speeds come from either very low or very high values of the turbulent viscosity coefficient. *Image courtesy of LLNL.*

Comparative visualization and analytics refers to the process of understanding quantitative and qualitative differences in datasets. Such characterizations may occur at many different levels: image to image, dataset to dataset (entire datasets or subsets), derived quantity to derived quantity, temporal analysis and visualization, (see Figures 6 and 4), and methodology to methodology. (see Figure 5). Image comparisons quantify the differences in images produced by a visualization process. Entire or subsetted datasets may be compared to one another. Derived quantities may be statistical moments of data fields or topological characteristics like the number and distribution of vortex cores. Methodology comparisons involve quantifying the differences in experiment or simulation parameters, or the differences in "recipe" to create an analysis or visualization result. Our work in this area will be driven by specific stakeholder needs as well as based on development of new metrics for data correlation [8].

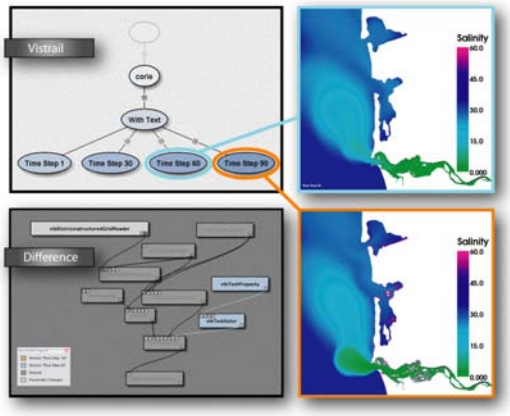

Figure 5. Capturing, comparing and analyzing workflow parameters and provenance is another form of comparative analysis that has applications in validation and verification, reproducibility, and encapsulation of potentially complex workflows. An emerging technology, VisTrails [2], captures and compares such data and is part of the VACET technology portfolio. *Image courtesy of the SCI Institute, University of Utah.*

Our stakeholders need better integrated statistical analysis and graphics, information and scientific visualization tools. An immediate example where such a combination could have an immediate positive benefit is in analyzing, understanding, and representing error and uncertainty in complex simulations [12] and comparisons. The result of such integration are tools that use data summary techniques like large data clustering and high-dimensional analysis to display "small multiples." Data clustering allows large amounts of data to be summarized succinctly, without sacrificing important details. High-dimensional analysis and displays will, among other things, allow scientists to identify lower dimensionality features within their data, greatly increasing data understanding.

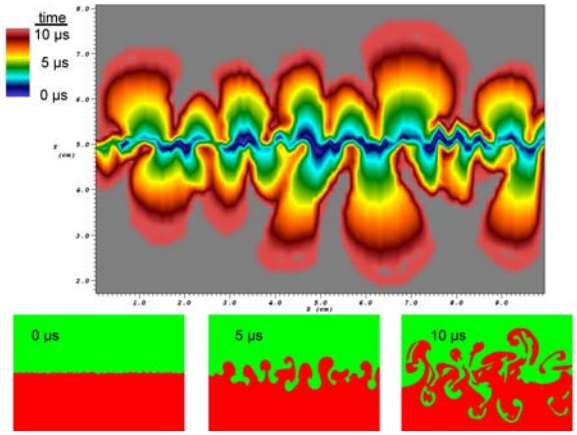

Figure 6. The bottom row of images shows the time evolution of a Rayleigh-Taylor instability simulation: heavy fluid on top (green) mixes with lighter fluid on bottom (red). Rather than focus on the differences between two time slices, the top image shows a summary of time-varying change over the entire dataset at all time steps. The top image shows a derived field $T(P)$, where $T(P)$ is the first time that mixing occurred at point P. Blue areas mixed early in the simulation, while red areas mixed later. The top image shows the mixing rate increased as the simulation progressed since there is more red than blue in the image. *Image courtesy of LLNL.*

3. Delivering Solutions

Team members on this proposal have been the primary developers of two major deployment vehicles, SCIRun, and VisIt, each of which has had a significant impact to date, and each of which will be utilized by our Center.

The SCIRun system has been a focus of research and development at the Scientific Computing and Imaging (SCI) Institute Institute since 1995 [11, 15, 16, 10]. It is a framework for visualization, modeling, and simulation, and has been the test bed for significant fundamental research in visualization techniques and their applications to real-world scientific problems. The strengths of SCIRun derive from its modular data flow architecture, which provides a much wider range of flexibility via modular pipelines and *dynamic compilation*. SCIRun2 [17] expands dramatically on these ideas to bring component-based scientific computing and visualization to an entirely new level. The primary novel feature in SCIRun2 is the concept of a metacomponent, [22] which allows construction of scientific software that involves mixtures of components from

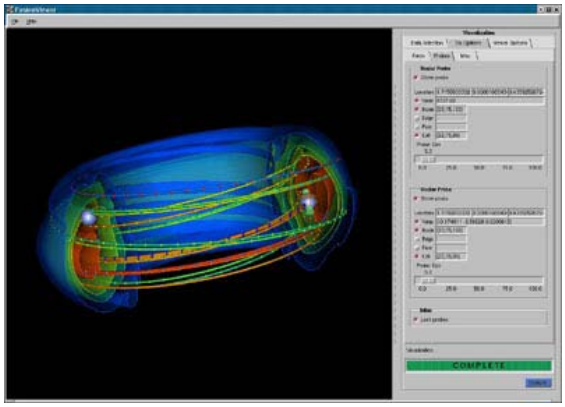

Figure 7. SCIRun, created by the SCI Institute at the University of Utah, is a framework for visualization, modeling, simulation and has served as a testbed for visualization research. This image shows visual output from a "power application" known as *FusionViewer*, which is a domain-specific front-end that uses the SCIRun infrastructure for visualization processing. *Image courtesy of the SCI Institute, University of Utah.*

different sources, including support for the Common Component Architecture (CCA) [1], the Visualization Toolkit [19], CORBA [14], and dataflow components from the original SCIRun. Components from these different sources can be combined in a single computation via the use of automatically- or semi-automatically-created bridges. SCIRun2 also enables parallel components through multi-threading for shared memory programming, and parallel-to-parallel remote method invocation [3, 7] for connecting components in a distributed memory environment.

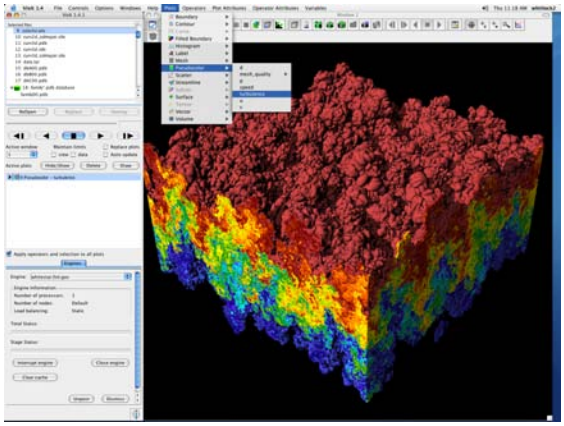

Figure 8. The VisIt application, created by LLNL as part of the ASC effort, is a full-featured open source turnkey solution for petascale visualization and analysis. *Image courtesy LLNL.*

VisIt [5, 6] is a turnkey application for data exploration, code assessment, and quantitative analysis suitable for use on tera- and peta-scale datasets. In addition to standard visualization methods, Visit is actively used for code-to-code comparisons, code-to-experiment comparisons, analysis of parameter studies, and quantitative analysis across a variety of scientific areas. It has won an R&D 100 award, been downloaded over 25,000 times, has over 300 customers at LLNL, and is used at many national laboratories, universities and businesses both domestic and abroad. Originating in 2000 as part of the ASC program, it has grown to over one million lines of code in addition to leveraging many third-party libraries. Its development has focused on key areas where solutions do not already exist: large data infrastructure, unusual data models, custom and extensible quantitative analysis and the infrastructure that binds them together. It has a scalable architecture for running expensive (I/O or compute) operations on a parallel machine to leverage resources "close to" the data and supports a client-server model for effective remote visualization use. It is extensible – developers can write plug-ins for any stage of I/O, visualization or analysis processing. Such plug-ins may run in parallel, thus providing a stable development environment for new techniques in scalable visualization and analysis.

Both VisIt and SCIRun will serve as the delivery vehicle for our Center's technologies. This approach allows us to leverage a large body of existing infrastructure – software technology as well as release engineering and support teams – to quickly deliver turnkey solutions aimed at solving domain-specific data understanding needs.

4. Project Organization

Our Center is organized into functional groups to achieve several distinct objectives: (1) facilitate the flow of information between the Center's leadership, personnel, and science stakeholders; (2) to provide the organizational structure needed to ensure oversight and coordinated operations of the Center's collection of activities; (3) to ensure we meet our work deliverables; (4) to gracefully accommodate future growth and respond to changing priorities. The Center's functional groups are: the Center PIs, the Executive Committee (EC), the External Advisory Board,

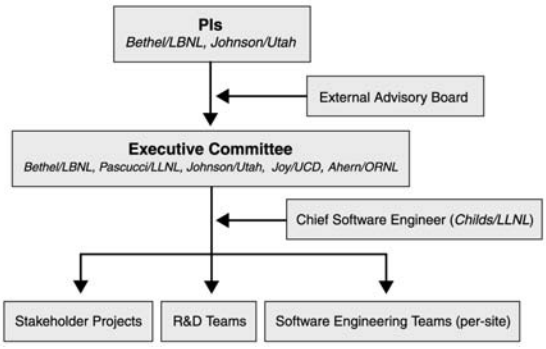

Figure 9. The VACET team's organization is designed to harmoniously accommodate its three primary thrust areas: stakeholder projects, software engineering and focused R&D teams.

Research and Development, Chief Software Engineer, Software Development and Support, and Stakeholder Projects.

Given the high priority on delivering useful software to our scientific stakeholders, each lab site will have a primary software engineer whose duty it is to assure software developed and deployed from that site meets the Center software engineering criteria. The set of site engineers is known collectively as the Software Engineering Group (SEG). In addition to developing, testing, documenting and maintaining the Center software, the Software Engineering Group will integrate results from all research and development groups into the Center software, relying on feedback from the project leads in the Stakeholder Projects Group, the Executive Committee and the Chief Software Engineer (Childs).

The Chief Software Engineer (CSWE) serves many important functions within the center. One is to facilitate the coordinated design, implementation and integration of the Center's technologies into software solutions that meet stakeholder needs. He will provide guidance to the software development teams so that individual software tools and libraries will be readily usable throughout the Center's collection of stakeholder projects. He will coordinate with the EC to prioritize software development targets, and serve as a technical software advisor to the Center as a whole. He will interact with the R&D team to help foster early designs that fit well within the Center's technology implementations. He will direct the development, testing, deployment, and support of the Center software toolsets.

The Stakeholder Projects Group (SPG) is the primary interface to our science stakeholders. In this group, individuals from the Center will interact directly with science stakeholders to obtain and prioritize science needs, coordinate with the Center's EC and Chief Software Engineer to translate those needs into a work plan, to oversee and manage the work so that software is delivered to the science stakeholder.

Acknowledgment

This work was supported by the Office of Science, Office of Advanced Scientific Computing Research, Mathematical, Information and Computational Sciences Division under the U.S. Department of Energy contracts DE-AC03-76SF00098 (UC/LBNL), DE-AC05-00OR2275 (UT-Batelle/ORNL); W-7405-Eng-48 (UC/LLNL); DE-FC02-01ER25457, DE-FG02-04ER25653, B524196, DE-FC02-01ER25493, DE-FC02-04ER25643, as well as by awards from NIH and NSF (Utah).

References

[1] R. Armstrong, D. Gannon, A. Geist, K. Keahey, S. Kohn, L. McInnes, S. Parker, and B. Smolinski. Toward a common component architecture for high-performance scientific computing. In *Proceedings of the 8th IEEE International Symposium on High Performance Distributed Computation*, August 1999.

[2] L. Bavoil, S.P. Callahan, P.J. Crossno, J. Freire, C.E. Scheidegger, C.T. Silva, and H.T. Vo. Vistrails: Enabling interactive multiple-view visualizations. In *Proceedings of IEEE Visualization 2005*, 2005.

[3] F. Bertrand, R. Bramley, K. Damevski, D. Bernholdt, J. Kohl, J. Larson, and A. Sussman. Data redistribution and remote method invocation in parallel component architectures. 2005. (Accepted, Best Paper Award).

[4] E. Wes Bethel, Randy Frank, Sam Fulcomer, Charles Hansen, Kenneth I. Joy, James Kohl, and Don Middleton. Visual Data Analysis – Report of the Visualization Breakout Session at the 2003 SCaLeS Workshop – Volume II, Arlington, VA. Technical Report LBNL-PUB-886 Vol II, Lawrence Berkeley National Laboratory, June 2003.

[5] Hank Childs, Eric Brugger, Kathleen Bonnell, Jeremy Meredith, Mark Miller, Brad Whitlock, and Nelson Max. A contract based system for large data visualization. In *Proceedings of IEEE Visualization 2005*, 2005.

[6] Hank Childs and Mark Miller. Beyond meat grinders: An analysis framework addressing the scale and complexity of large data sets (to appear). 2006.

[7] K. Damevski and S.G. Parker. Parallel remote method invocation and m-by-n data redistribution. In *Proceedings of the 4th Los Alamos Computer Science Institute Symposium*, page (published on CD), 2003.

[8] Herbert Edelsbrunner, John Harer, Vijay Natarajan, and Valerio Pascucci. Local and global comparison of continuous functions. In *Proceedings of the IEEE conference on Visualization (VIS-04)*, pages 275–280, October 2004.

[9] Bernd Hamann, E. Wes Bethel, H.D. Simon, and J.C. Meza. NERSC Visualization Greenbook-Future Visualization Needs of the DoE Computational Science Community hosted at NERSC. *International Journal of High Performance Computing Applications*, 2003.

[10] C.R. Johnson, S. Parker, and D. Weinstein. Large-scale computational science applications using the SCIRun problem solving environment. 2000.

[11] C.R. Johnson and S.G. Parker. Applications in computational medicine using SCIRun: A computational steering programming environment. In H.W. Meuer, editor, *Supercomputer '95*, pages 2–19. Springer-Verlag, 1995.

[12] C.R. Johnson and A.R. Sanderson. A next step: Visualizing errors and uncertainty. *IEEE Computer Graphics and Applications*, 23(5):6–10, September/October 2003.

[13] Richard Mount. The Office of Science Data-Management Challenge. Report from the DOE Office of Science Data-Management Workshops. Technical Report SLAC-R-782, Stanford Linear Accelerator Center, March-May 2004.

[14] OMG. Corba Component Model, visited 1-11-2000. http://www.omg.org/cgi-bin/doc?orbos/97-06-12.

[15] S.G. Parker and C.R. Johnson. SCIRun: A scientific programming environment for computational steering. In *Supercomputing '95*. IEEE Press, 1995.

[16] S.G. Parker, D.M. Weinstein, and C.R. Johnson. The SCIRun computational steering software system. In E. Arge, A.M. Bruaset, and H.P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 1–40. Birkhauser Press, Boston, 1997.

[17] S.G. Parker, K. Zhang, K. Damevski, and C.R. Johnson. *Integrating Component-Based Scientific Computing Software*, page (accepted). 2005.

[18] Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood, and Andy Hopper. Virtual network computing. *IEEE Internet Computing*, 2(1):33–38, 1998.

[19] W Schroeder, K Martin, and W.E. Lorensen. *The Visualization Toolkit*. Prentice-Hall Inc., 1996.

[20] Lawrence Berkeley National Laboratory Scientific Data Management Group. Fastbit. http://sdm.lbl.gov/fastbit, 2005.

[21] Kurt Stockinger, John Shalf, Kesheng Wu, and E. Wes Bethel. Query-driven visualization of large data sets. In *Proceedings of IEEE Visualization 2005*, pages 167–174, 2005.

[22] K. Zhang, K. Damevski, V. Venkatachalapathy, and S.G. Parker. SCIRun2: A CCA framework for high performance computing. April 2004.